通过诊断图分析的快速诊断算法

陈蔼祥[1] 陈清亮[2] 潘久辉[2] 姜云飞[3] 杨晋吉[4]

[1](广东商学院数学与计算科学学院 广州 510320)

[2](暨南大学计算机系 广州 510632)

[3](中山大学软件研究所 广州 510275)

[4](华南师范大学计算机学院 广州 510631)

摘 要 基于模型诊断的主要思想是:根据系统的逻辑模型以及系统的输入,通过逻辑的推理理论能推导出系统在正常情况下的预期行为,如果观测到的系统实际行为与系统预期行为有差异,则说明系统存在故障.当系统故障时,可通过逻辑的推理理论来确定引发故障的元件集合.由于经典的基于模型诊断采用的是逻辑推理的手段来产生诊断集合,这导致了传统的基于模型诊断算法的效率非常低下.文中在原有模型诊断基础上,重新定义了诊断,并提出了一种用于诊断的诊断图的数据结构.在此基础上给出了一种基于诊断图分析的快速诊断算法.由于文中的诊断方法是一种过程化的方法,与 Reiter 的模型诊断的基于逻辑的方法有着本质的不同.因此,文中的方法能很好地克服经典模型诊断效率过低的问题,为诊断问题的求解带来新的前景.实验结果证明了这种新的诊断方法的高效性.

关键词 诊断;诊断图;基于模型的诊断;溯因推理;自动规划

中图法分类号 TP18 **DOI 号**: 10.3724/SP.J.1016.2009.01470

Fast Diagnosing Through Diagnostic Graph Analysis

CHEN Ai-Xiang[1] CHEN Qing-Liang[2] PAN Jiu-Hui[2] JIANG Yun-Fei[3] YANG Jin-Ji[4]

[1](*Department of Mathematics, Guangdong University of Business Studies, Guangzhou* 510320)

[2](*Department of Computer Science, Jinan University, Guangzhou* 510632)

[3](*Software Research Institute, Sun Yat-Sen University, Guangzhou* 510275)

[4](*School of Computer Science, South China Normal University, Guangzhou* 510631)

Abstract Given the logical model of system and its input, when an observation of system's behavior conflicts with the way the system meant to behave, we can determine those components of the system which, when assumed to be functioning abnormally, will explain the discrepancy between the observed and correct system behavior. That is the main idea of model-based diagnosis. The classic diagnosis method is very inefficient because its diagnosis procedure is based on logical reasoning. In this paper, the authors redefine the diagnosis, and introduce a new procedure-based rather than logic-based approach to compute diagnosis based on constructing and analyzing a compact structure which we call a diagnostic graph. It is shown that it is a better choice since the search made by this approach is fundamentally different from the search of classic model-based one. So the approach in this paper can provide a new perspective on the diagnosis problem. Finally, the effectiveness of this methodology is demonstrated by the experimental results.

Keywords diagnosis; diagnostic graph; model-based diagnosis; abductive reasoning; automated planning

---

收稿日期:2007-11-12;最终修改稿收到日期:2009-06-09.本课题得到国家"九七三"重点基础研究发展规划项目基金(2005CB321902)、国家自然科学基金(60773201)、国家自然科学基金国际合作项目(60911130005)、暨南大学引进人才启动基金、广东高校优秀青年创新人才培育项目(LYM08017)、广东省科技计划项目(2007B010400068)和广东省产学研结合项目(2007B090400095)支持.**陈蔼祥**,男,1978 年生,博士,讲师,主要研究方向为智能规划与诊断.**陈清亮**(通信作者),男,1980 年生,博士,讲师,主要研究方向为人工智能和自动推理.E-mail: tsingliangchen@gmail.com.**潘久辉**,男,1956 年生,教授,博士生导师,主要研究领域为知识工程.**姜云飞**,男,1945 年生,教授,博士生导师,主要研究领域为自动推理、智能规划和基于模型的诊断等.**杨晋吉**,男,1968 年生,副教授,主要研究方向为人工智能.

1 引　言

基于模型诊断[1]就是根据系统组成元件与元件之间的连接建立起待诊断系统模型,这种模型通常用一阶逻辑语句来描述,根据系统的逻辑模型以及系统的输入,我们能通过逻辑的推理理论推导出系统在正常情况下的预期行为,如果观测到的系统实际行为与系统预期行为有差异,就说明系统存在故障,利用逻辑的推理理论,我们能够确定引发故障的元件集合.基于模型的诊断与传统的诊断方法有着重要的区别,因而被一些人工智能专家誉为诊断理论和诊断技术上的革命.其优点为:(1)不需要积累诊断经验;(2)具有极强的设备独立性;(3)诊断能力强;(4)可以根据系统的组成元件的功能和元件之间的连接对产生的故障给出令人信服的解释,给待诊断系统的修复提供了极大的方便.基于模型的诊断有两个主要的研究方法,一个是基于一致性的诊断[1-4],另一个是溯因诊断[5-7].基于一致性的诊断对诊断空间要求较弱,诊断空间里会包含无用解.溯因诊断对诊断空间的限制强,但如果模型不完备,可能丢失真正的解.Console 和 Torasso[8]把诊断问题看成是带有一致性约束的溯因问题,融合了两种方法.事实上,溯因诊断空间是包含在一致性诊断空间里的.

按照 Reiter 的方法,诊断的过程分两步:(1)产生所有的极小冲突集;(2)求极小冲突集的极小碰集得到极小诊断.然而通过定理证明器或归结方法计算冲突集的复杂性是指数级的.因而基于 Reiter 的模型诊断方法求解诊断的效率是非常低下的,其后的研究者们在对提高诊断的效率方面做了不少的工作:Chittaro 等[9-10]采用结构抽象的方法实现分层的基于模型的诊断,取得了很好的结果.系统分解也是提高效率的方法,李占山等[11]研究了基于模型的诊断问题分解及其算法.另外,在系统模型中加入一些约束可以缩小诊断空间,提高诊断的效率,欧阳丹彤等[12]通过引入定义信息和表示伴随关系的信息,有效地提高了诊断的效率,减少了候选诊断的数量.陈荣等[13]提出了含约束的基于模型的诊断系统,通过增加约束控制诊断空间,并给出了选取理想约束的理论依据.Fattah 和 Dechter[14]与 Stumptner 和 Wotawa[15]给出了诊断树型结构系统的方法,对于输入不超过两个的系统算法在多项式时间内结束.栾尚敏等[16]给出了利用结构信息的故障诊断方法,对于一些特殊结构的系统可在多项式时间内结束.Mozetic[17]与 Childress 和 Valtorta[18]介绍了求第 k 个诊断的方法.但这些工作大多是基于 Reiter 的模型诊断框架的,因此,诊断的产生通常必须借用相应的定理证明器,从而不可避免地带来效率低下的问题.

我们在 Reiter 的模型诊断的基础上,重新定义了诊断,提出了一种用于诊断的诊断图的数据结构.在此基础上,我们给出了一种基于诊断图分析的快速诊断算法.我们已经在 Linux 平台下实现了基于我们方法的快速图诊断工具 GDT(Graph Diagnosing Tool).通过运行该工具,并对一些诊断问题进行诊断,诊断结果都能对系统故障原因给出很好的解释,并且运算速度极快.

作为模型诊断的基础,第 2 节首先对 Reiter 的模型诊断方法进行简要介绍,并分析其不足之处;第 3 节给出一个用诊断图分析进行诊断的例子,对基于诊断图分析进行诊断的方法进行非形式的介绍;第 4 节系统地介绍基于诊断图分析的快速诊断方法;第 5 节给出用诊断图分析进行诊断的例子.最后讨论相关工作并进行总结.

2 基于模型的诊断

作为基础,本小节首先介绍基于模型诊断的基本思想,然后介绍关于基于模型诊断的形式定义以及 Reiter 的诊断产生过程.

2.1 基于模型诊断的主要思想

基于模型诊断的主要思想是:根据系统组成元件与元件之间的连接建立起待诊断系统模型(如结构、功能、行为),这种模型通常用一阶逻辑语句来描述,根据系统的逻辑模型以及系统的输入,我们能通过逻辑的推理理论推导出系统在正常情况下的预期行为,如果观测到的系统实际行为与系统预期行为有差异,就说明系统存在故障,利用逻辑的推理理论,我们能够确定引发故障的元件集合.关于模型诊断的主要思想,可用图 1 表示如下.

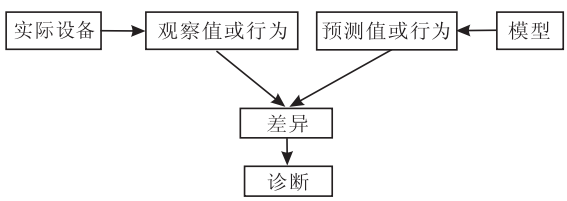

图 1　诊断的主要思想

2.2 基于模型诊断的形式定义

1987 年 Reiter 在《Artificial Intelligence》上发表了“第一原理(first principle)的诊断理论”一文[1],对基于一致性诊断给出了很好的形式化描述.

定义 1. 对$(SD,COMPS,OBS)$的诊断是一(极小)集合$\Delta\subseteq COMPS$,使得$SD\cup OBS\cup\{ab(c)|c\in\Delta\}\cup\{\neg ab(c)|c\in COMPS-\Delta\}$是一致的.其中,$SD$为系统描述,是一阶句子集合;$COMPS$为系统元件,是一有限常量集合;$OBS$为系统观测,是一有限一阶句子集合;$ab$是一元谓词,意味着“abnormal”,当元件$c\in COMPS$异常时,$ab(c)$为真.

Reiter 在他的上述文章中,根据一阶逻辑的推理理论,又简化了诊断的上述定义,给出了一个等价的定义,其定义如下.

定义 2. 对$(SD,COMPS,OBS)$的诊断是一(极小)集合$\Delta\subseteq COMPS$,使得$SD\cup OBS\cup\{\neg ab(c)|c\in COMPS-\Delta\}$是一致的.

因为定义 2 与定义 1 是等价的,但比定义 1 要简单,少了一个$\{ab(c)|c\in\Delta\}$,所以定义 2 比定义 1 更常被研究者们使用.

2.3 Reiter 的诊断产生过程

Reiter 提出的诊断的定义是基于逻辑的一致性的. 根据 Reter 的诊断定义 2,人们很容易得出由系统的描述和实际观测结果求诊断的方法. 这就是:先系统地产生系统元件集合 $COMPONENT$ 的所有子集Δ,然后测试

$$SD\cup OBS\cup\{\neg ab(c)|c\in COMPS-\Delta\}$$

的一致性.但是,我们知道随着系统元件数量的增加,系统的所有子集数量会以指数级的数量增加,而且要判断一个逻辑语句集合是否一致是十分困难的,因此,可以很容易地预见到根据 Reiter 定义直接求一个系统的诊断的效率会是十分低下的.因此,Reiter 采用冲突集的概念来计算诊断,冲突集的概念最先是由 de Kleer 提出来的[19].

定义 3. 系统$(SD,COMPONENT,OBS)$的冲突集是一个集合$\{C_1,C_2,\cdots,C_K\}$,满足下述条件:

$$SD\cup OBS\cup\{\neg ab(C_1),\cdots,\neg ab(C_K)\}$$

是不一致的.

定义 4. 一个冲突集称作是最小冲突集,如果它不存在冲突集作为它的子集.

我们如何利用冲突集来计算诊断呢? 为此,我们先给出碰集的概念.

定义 5. 设 C 是一个集合簇,C 的碰集(hitting set)是一个集合

$H\subseteq\bigcup_{s\in C}S$,使得对所有 $S\in C$,$H\cap S\neq\{\ \}$.

类似于最小冲突集的概念,我们还可以定义最小碰集.

定义 6. 一个碰集称作是最小碰集,如果它不存在碰集作为它的子集.

定理 1. 设Δ是系统元件集合 $COMPONENT$ 的子集,则Δ是系统$(SD,COMPONENT,OBS)$的诊断当且仅当Δ是$(SD,COMPONENT,OBS)$的最小冲突集簇的最小碰集.

综上所述,按照 Reiter 的方法,诊断的过程大体可分两步:(1)产生所有的极小冲突集;(2)求极小冲突集的极小碰集得到极小诊断.但是,通过定理证明器计算冲突集的复杂性是指数级的,因此,Reiter 的关于诊断产生方法的效率是低下的.后来的研究者们虽然对 Reiter 的方法进行了改进,但由于这些改进工作都是在逻辑推理的框架下进行的,因此,改进的效率都不能令人满意.究其原因,主要是由于逻辑演绎的方法效率低下的问题无法解决.基于这一点认识,我们设想能否设计一种基于程序化过程化的诊断方法.在这一思想的指导下,我们对模型和诊断进行了重新定义,并给出了一种通过诊断图分析来快速诊断的办法.

3 通过诊断图分析进行诊断的例子

为了使读者能对本方法有一个直观的印象,下面我们首先以 Reiter 论文中的全加器为例,介绍如何用诊断图分析的方法来产生诊断.

Reiter 在其论文中介绍了一个全加器的例子,这个例子如图 2 所示.

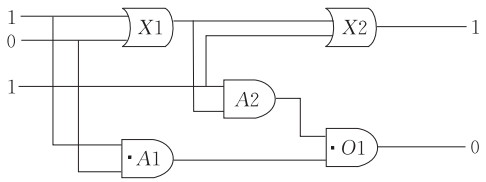

图 2 Reiter 的论文中系统的例子

按照 Reiter 对模型的定义,可对上述系统的模型按下面的方式进行描述:

(1)系统的部件主要有

$$COMPENT=\{A1,A2,X1,X2,O1\}.$$

(2)系统描述可由下列形式给出:

$ANDG(x)\neg AB(x)\supset out(x)=\mathrm{and}(in1(x),in2(x))$,

$XORG(x)\neg AB(x)\supset out(x)=\mathrm{xor}(in1(x),in2(x))$,

$ORG(x)\neg AB(x)\supset out(x)=\mathrm{or}(in1(x),in2(x))$,

$ANDG(A1)$,$ANDG(A2)$,$XORG(X1)$,
$XORG(X2)$,$ORG(O1)$,
$out(X1)=in2(A2)$, $out(X1)=in1(X2)$,
$out(A2)=in1(O1)$, $in1(A2)=in2(X2)$,
$in1(X1)=in1(A1)$, $in2(X1)=in2(A1)$,
$out(A1)=in2(O1)$.

显然,上述关于系统的描述主要由下面 3 部分构成:

(1) 描述异或门、与门、或门功能的语句,例如:
$ANDG(x)\neg AB(x)\supset out(x)=\text{and}(in1(x),in2(x))$.

(2) 描述各元件属性的语句,例如,

$ANDG(A1)$, $XORG(X1)$等.

(3) 描述各元件间互相连接的语句,例如:
$out(X1)=in2(A2)$, $out(X1)=in1(X2)$.

此外,还有包含对输入输出进行限制的语句,例如,

$in1(X1)=0\vee in1(X1)=1$,
$in2(X1)=0\vee in2(X1)=1$ 等.

以及系统的观察值 OBS,例如,

$in1(X1)=1$, $in2(X1)=0$, $in1(A2)=1$,
$in1(A1)=1$, $in2(A1)=0$,
$out(X2)=1$, $out(O1)=0$.

显然,Reiter 的上述关于模型的描述,决定了其对诊断的定义以及求解诊断的方法与逻辑推理脱不了关系,这也是基于 Reiter 的模型诊断效率低下的重要原因.

为了能提高诊断产生的效率,就必须设法改变 Reiter 的这种用逻辑推理产生诊断的办法,而要做到这一点,必须对模型的描述进行适当的修改.下面同样以全加器为例,对我们的工作先进行一个非形式化的介绍.

首先,对系统功能描述部分进行修改,我们的做法是,给出每个部件分别在正常状态下(用谓词 NR 表示)和出故障的情况下(用 AB 表示)的各种输入输出情况.例如,对于图 1 的例子中的异或门 $X1$,我们可以给出 $X1$ 在正常情况下和异常情况下的各种输入输出情况如表 1 所示.

表 1　异或门 *X1* 在正常模式和故障模式下的各种输入输出情况

NR-1($X1$) Input: $in1(X1)=0$, $in2(X1)=0$ Output: $out(X1)=0$ Link: $in1(X2)=0$, $in2(A2)=0$	AB-1($X1$) Input: $in1(X1)=0$, $in2(X1)=0$ Output: $out(X1)=1$ Link: $in1(X2)=1$, $in2(A2)=1$
NR-2($X1$) Input: $in1(X1)=0$, $in2(X1)=1$ Output: $out(X1)=1$ Link: $in1(X2)=1$, $in2(A2)=1$	AB-2($X1$) Input: $XORG(X1)$, $in1(X1)=0$, $in2(X1)=1$ Output: $out(X1)=0$ Link: $in1(X2)=0$, $in2(A2)=0$
NR-3($X1$) Input: $in1(X1)=1$, $in2(X1)=0$ Output: $out(X1)=1$ Link: $in1(X2)=1$, $in2(A2)=1$	AB-3($X1$) Input: $XORG(X1)$, $in1(X1)=1$, $in2(X1)=0$ Output: $out(X1)=0$ Link: $in1(X2)=0$, $in2(A2)=0$
NR-4($X1$) Input: $in1(X1)=1$, $in2(X1)=1$ Output: $out(X1)=0$ Link: $in1(X2)=0$, $in2(A2)=0$	AB-4($X1$) Input: $in1(X1)=1$, $in2(X1)=1$ Output: $out(X1)=1$ Link: $in1(X2)=1$, $in2(A2)=1$

其它各个部件在正常情况下和异常情况下的输入输出情况可按类似的方式给出.由此,我们可以得到所有部件的正常情况下和异常情况下的输入输出情况集合,我们将这样的集合称为元件的工作模式集合,记为 T.有了系统的元件工作模式集合 T 后,根据系统的输入,可得到下面如图 3 所示的关系转换图,我们称之为诊断图.

上述诊断图中,最后一层命题层必须包含表示系统的所有输出的命题.因此,对于给定的待诊断系统,我们可以从系统的初始输入出发,根据元件的工作模式集合,可以逐层构造类似图 3 所示的诊断图,直至得到系统的所有输出为止.然后在诊断图上根据系统的输出,反向搜索能导致系统输出的系统元件的转换模式集合,直至达到系统的输入层.反向搜索过程中找到的系统转换模式集合就是我们所需要的诊断.这就是基于诊断图的整个诊断过程.

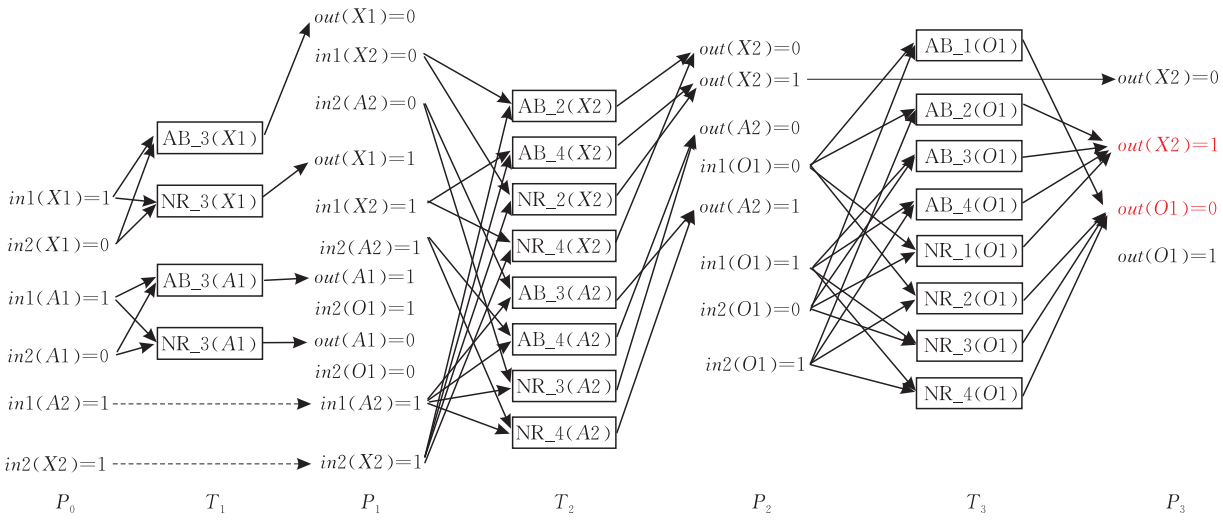

图 3 用于诊断的诊断图(图中带方框的表示元件的转换模式,没带方框的表示元件的输入输出状态的命题,并且图中命题层和转换模式层交替出现,图的最后一层必须包含系统的最终输出,图中箭头表示元件的输入或输出.为简洁起见,图中忽略了一些连接边,命题之间、元件工作模式之间的不一致关系也没有在图中标明,虚线表示的伪元件工作模式也只画出了一部分)

4 基于诊断图分析的快速诊断

我们先给出新的模型以及诊断的形式定义,在此基础上介绍基于诊断图的诊断算法.

4.1 模型及诊断相关定义

定义 7. 假定 L 为一阶逻辑语言,则所谓状态就是表示待诊断系统的各个元件的输入输出集合.该集合是用一阶语言 L 表示的逻辑原子集合.

请注意,上述关于状态的定义是基于闭世界假设的,即对于不在状态中出现的元件的输入或输出,我们就认为该状态下关于该元件的输入输出是不知道的.

例如,图 2 中系统的初始状态可表示成

$$s_0=\{in1(X_1)=1,in2(X_1)=0,in1(A_2)=1,in2(X_2)=1,in1(A_1)=1,in2(A_1)=0\}$$

的形式.显然,图 2 中元件 $X2$ 的输入 $in1(X2)$ 就不在状态 s_0 中出现,因为此时并不知道 $in1(X2)$ 的取值情况.同样,整个系统的输出 $out(X2)$ 和 $out(O1)$ 也不在系统的初始状态 s_0 中出现,因为虽然 $out(X2)$ 和 $out(O1)$ 是系统的输出值,是可通过观察得到的观察值,但在经过系统的加工处理前,我们不认为 $X2$ 和 $O1$ 的输出是已经知道的,因此 $out(X2)$ 和 $out(O1)$ 也不在系统的初始状态中出现.

定义 8. 所谓元件的工作模式是四元组

$$t=\langle name(t),input(t),out(t),link(t)\rangle,$$

其中,$name(t)$ 表示元件工作模式名称;$input(t)$ 表示元件的输入集合;$output(t)$ 表示元件的输出集合;$link(t)$ 表示元件之间连接引起的其它元件的输入集合.

例如,图 2 中元件 $X1$ 的一个工作模式可表示成如下形式:

AB-1($X1$)

Input: $in1(X1)=0,in2(X1)=0$

Output: $out(X1)=1$

Link: $in1(X2)=1,in2(A2)=1$

上述工作模式表明,元件 $X1$ 处于故障模式下,此时如果元件 $X1$ 的输入 $in1(X1)=0,in2(X1)=0$ 的话,其输出 $out(X1)=1$,并且导致元件 $X2$ 和 $A2$ 的相应输入为 $in1(X2)=1,in2(A2)=1$.

对于待诊断系统,给定输入,经过系统相应元件处理后,系统的状态将会发生改变.由于对于同一个元件,其工作模式可能有多种,比如一个元件可能正常工作,也可能出了故障,因此,在得到系统的诊断之前,我们对系统中的元件的工作模式是不可能知道的.那么,对于给定输入的系统,我们该如何选择相应元件的工作模式?当元件的某个工作模式所对应的元件输入集合存在于某一状态 s 时,称该工作模式可用于状态 s.这样,在某个状态 s 下,使用一个模式 t 将得到一个结果状态,该结果状态可按下式计算得到

$$\gamma(s,t)=\begin{cases}s\cup output(t)\cup link(t), & input(t)\subseteq s\\ undefine, & \text{否则}\end{cases}.$$

在给出了系统的状态和元件的工作模式的定义

后,我们可以得到有关系统的模型的定义.

定义 9. 所谓模型就是一个二元组 $M=\langle S,T\rangle$,其中,S 表示待诊断系统的所有可能状态集合;T 表示待诊断系统的所有元件的所有可能工作模式集合.

上述模型的定义与 Reiter 关于模型的定义有着明显的差别,在 Reiter 的模型的定义中,是以命题语言或一阶语言的形式给出系统的元件描述、系统的功能描述以及系统各元件之间的连接描述.但在我们的定义中,系统的元件描述、系统的功能描述以及系统各元件之间的连接描述都被紧凑地表示成元件的工作模式的形式.有了模型的定义后,下面我们给出关于诊断问题的定义.

定义 10. 所谓诊断问题就是三元组 $P_D=\langle s_0,M,OBS\rangle$,其中,$s_0:s_0\subseteq S$ 表示待诊断系统的初始状态;M 表示待诊断系统的模型;OBS 表示系统的观察值.

有了模型及诊断问题的定义之后,我们可以得到新的诊断的定义.

定义 11. 对于给定的诊断问题 P_D,所谓诊断就是指元件的可能工作模式集合,记为 $D=\{t_1,t_2,\cdots,t_n\}$,在这些工作模式集合下,系统能在给定的输入下,得到相应的观察值.

显然,智能诊断的任务是给定系统的模型以及系统输入,当系统的观察值与系统的正常输出值不一致时,要求能自动地确定那些出了故障的元件.前面我们给出了新的关于系统的模型及诊断的相关定义.我们的关于系统模型以及诊断的定义跟 Reiter 的模型和诊断的定义是有着本质的不同,Reiter 关于诊断的定义是基于逻辑一致性的,因此,其求解诊断的方法也是基于逻辑推理的,这导致求解诊断的效率极为低下.为此,我们给出了新的关于模型和诊断的定义,并在此基础上,提出了一种基于诊断图的诊断算法.

4.2　诊断图及相关定义

定义 12. 所谓诊断图是指包含两类节点、三类边的有向无环分层图.诊断图中表示元件的输入输出状态的命题层和表示元件工作模式的元件工作模式层交替出现,命题层包含一些命题节点(标识为一些命题),元件工作模式层包含元件工作模式的节点(标识为元件工作模式).在第 t 层的元件工作模式节点代表时间步 t 所有可能的元件工作模式.事实节点表示的命题对应于时间步 t 被例化的一个或多个元件工作模式的输入,或者是时间步 $t-1$ 被例化的一个或多个元件工作模式的输出及其相应的连接元件的输入.诊断图第 0 层是命题层,包含诊断问题初始状态的所有命题,即系统元件的初始输入.诊断图的最后一层应包含诊断问题的所有观察值.诊断图中元件工作模式节点和命题节点之间有三类边相连.假定 t 为第 i 层元件工作模式节点,则有下面的边与 t 相连:输入边(input edges):连接第 i 层的表示 t 的输入的命题节点;输出边(output edges):连接第 $i+1$ 层的表示 t 的输出的命题节点;连接边(link edges):连接第 $i+1$ 层的表示其它元件由于与元件 t 相连而得到的相应的输入的命题节点.

根据诊断图的上述定义,可得到如图 4 所示的诊断图.图中黑圆点代表命题节点,空白方框代表元件工作模式节点.

诊断图命题层中可能存在不一致命题,例如,图 3 中,命题 $in1(X2)=0$ 和 $in1(X2)=1$ 同时出现在命题层 P_1 上,显然这两个命题是不一致的命题.同样,元件工作模式层中也可能存在相互不一致的元件工作模式,例如,图 3 的 T_1 层元件工作模式层中,同时存在 AB_3($X1$)和 NR_3($X1$)两个工作模式,而这两个工作模式下,将会导致命题 $in1(X2)=0$ 和 $in1(X2)=1$ 同时出现在下一命题层上,而上述两个命题是不一致的,因此,AB_3($X1$)和 NR_3($X1$)两个工作模式也是不一致的.下面给出伪工作模式、不一致的命题和不一致的元件工作模式的定义.

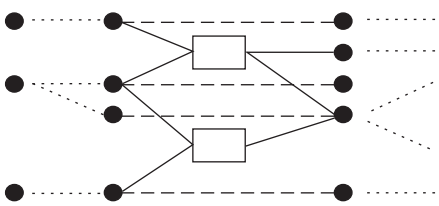

图 4　用于诊断的诊断图(示意图)

定义 13. 对于某一命题 p,所谓相对于命题 p 的伪工作模式就是指元件的输入和输出均为 p,且连接为空的工作模式,记为 NO-MODE(p).

显然,伪工作模式的作用是将前一层命题层的命题往后一层命题层传播,如图 4 中虚线即表示相应的命题的伪工作模式.例如图 3 中,P_0 层命题层存在一命题 $in1(A2)=1$,该命题在伪工作模式 NO-MODE($in1(A2)=1$)的作用下,从 P_0 命题层传播到 P_1 命题层.

定义 14. 两元件工作模式(t_1,t_2)满足以下两种情况中的任一种情况时,则两元件工作模式是不一致的:输出/连接不一致(inconsistent):一种元件

工作模式的输出或连接与另一种元件工作模式的输出或连接不一致;输入不一致:两个元件工作模式同时具有 $i-1$ 层不一致的输入.

例如,图 3 中 T_1层元件工作模式层有两个工作模式 AB_3($X1$)和 NR_3($X1$),这两个工作模式分别具有 $out(X1)=0$ 和 $out(X1)=1$ 的输出,由于 $out(X1)=0$ 和 $out(X1)=1$ 两个命题是不一致的,因为同一个元件不可能同时有 0 和 1 两个不同的输出,因此两个工作模式 AB_3($X1$)和 NR_3($X1$)的输出不一致,从而这两个工作模式也是不一致的.同样,T_2 层元件工作模式层有两个工作模式 AB_2($X2$)和 NR_4($X2$)是不一致的,因为 AB_2($X2$)的一个输入 $in1(X2)=0$ 和 NR_4($X2$)的一个输入 $in1(X2)=1$ 是不一致的.

定义 15. 当出现下面两种情况之一时,两个表示元件的输入输出的命题不一致:一个命题是另一个命题的否定;不一致支持(inconsistent support)是指获得这对命题的所有元件工作模式都是两两不一致的.

例如图 3 中,P_1 命题层中的命题 $in1(X2)=0$ 和 $in1(X2)=1$ 是不一致的,因为同一个元件的输入不可能既是 0 又是 1.同样,$in1(X2)=0$ 和 $out(X1)=1$ 是不一致的,因为获得这两个命题的元件工作模式 AB_3($X1$)与 NR_3($X1$)是不一致的.

约定诊断图中第 i 层元件工作模式层用 T_i 表示,第 i 层不一致元件工作模式集合用 μT_i 表示.诊断图第 i 层命题层用 P_i 表示,不一致命题集合用 μP_i 表示.

根据上述诊断图及其相关定义,我们可以得到以下两个命题.

命题 1. 如果在诊断图命题层 P_{i-1} 有两命题 p 和 q,且 $(p,q)\notin\mu P_{i-1}$,则 $(p,q)\notin\mu P_i$;如果在诊断图元件工作模式层 T_{i-1} 有两元件工作模式 t_1,t_2,且 $(t_1,t_2)\notin\mu T_{i-1}$,则 $(t_1,t_2)\notin\mu T_i$.

证明:根据诊断图的定义,对于 P_i 中每一命题 p,至少存在一个伪元件工作模式 t_p 支持该命题.对于 P_{i-1} 中的两个命题 p 和 q,由于 $(p,q)\notin\mu P_{i-1}$,即命题 p 和 q 是一致的.因此 $(t_p,t_q)\notin\mu T_i$,即将两命题 p 和 q 从 P_{i-1} 向 P_i 传播的伪元件工作模式也是相互一致的.由此可得在 P_{i-1} 层一致的命题在 P_i 层仍然保持一致.类似地,如果 $(t_1,t_2)\notin\mu T_{i-1}$,则表明两元件工作模式 t_1 和 t_2 是相互一致的元件工作模式,这意味着两元件工作模式 t_1 和 t_2 的输入在命题 P_{i-1} 是一致的,因此,可推得元件工作模式 t_1 和 t_2 在 T_i 层保持一致,即 $(t_1,t_2)\notin\mu T_i$.

上述命题表明,诊断图中的不一致关系不具备传递性.而诊断图上的命题层的命题数和元件工作模式层的元件工作模式数则是单调递增的,即对于诊断图上任意第 i 层,$P_{i-1}\subset P_i$,$T_{i-1}\subset T_i$.

定义 16. 对于命题集合 OBS,如果在诊断图上存在一命题层,使 $OBS\subset P_i$,并且 OBS 中不存在有不一致命题,即 $\neg\exists o_1,o_2\in OBS,(o_1,o_2)\in\mu P_i$,则称命题集合 OBS 从初始状态 s_0 可达.

上述定义的含义是说,命题集合 OBS 在 P_i 层能一致地出现,表明 T_{i-1} 层元件工作模式层存在相互一致的元件工作模式支持命题集合 OBS,而这些相互一致的元件工作模式必然存在相互一致的输入存在于命题层 P_{i-1},依次类推,直到初始状态层 P_0.因此,命题集 OBS 从 s_0 可达.

4.3 诊断的计算

诊断的计算过程分为两个阶段进行:构造诊断图阶段和诊断计算阶段.构造诊断图阶段从系统输入(表示系统的初始状态)出发,根据系统的元件之间的连接关系,前向扩展诊断图,直到能一致地得到系统的观察值集合.诊断计算阶段从系统的观察值出发,反向搜索诊断图,以求出能导致系统观察值的系统元件工作模式集合.

4.3.1 诊断图构造算法

令 $P_D=\langle s_0,M,OBS\rangle$ 为一诊断问题,其中 s_0 和 OBS 为命题集合,代表诊断问题的初始输入和观察值,M 为系统的模型,其中包含系统所有可能的状态 S 以及系统元件所有可能的工作模式集合 T,包括元件的正常工作模式集合和异常工作模式集合.由此,我们可以得到从 i 层诊断图构造 $i+1$ 层诊断图的构造算法.

算法 1. 诊断图扩展算法(从第 i 层诊断图扩展得到第 $i+1$ 层诊断图).

Expand($\langle P_0,T_1,\mu T_1,\cdots,T_i,\mu T_i,P_i,\mu P_i\rangle$)

Input: $G=\langle P_0,T_1,\mu T_1,\cdots,T_i,\mu T_i,P_i,\mu P_i\rangle$. A diagnostic graph for that problem expanded up to level i

Output: $G=\langle P_0,T_1,\mu T_1,\cdots,T_{i+1},\mu T_{i+1},P_{i+1},\mu P_{i+1}\rangle$, A diagnostic graph for that problem expanded up to level $i+1$

1. $T_{i+1}\leftarrow\{t\in T\mid input(t)\subseteq P_i$ and $input^2(t)\cap\mu P_i=\varnothing\}$
2. $P_{i+1}\leftarrow\{p\mid\exists t\in T_{i+1}:p\in output(t)\vee p\in link(t)\}$
3. $\mu T_{i+1}\leftarrow\{(t_1,t_2)\in T_{i+1}^2,t_1\neq t_2\mid\exists p_1\in output(t_1)\cup link(t_1),\ \exists p_2\in output(t_2)\cup link(t_2),(p_1,p_2)\in\mu P_{i+1}$ or $\exists p_1\in input(t_1),$

$\exists p_2 \in input(t_2), (p_1, p_2) \in \mu P_i\}$

4. $\mu P_{i+1} \leftarrow \{(p,q) \in P_{i+1}^2, p \neq q \mid \exists (t_1, t_2) \in \mu T_{i+1}: p \in output(t_1) \cup link(t_1), q \in output(t_2) \cup link(t_2)\}$

5. for each $t \in T_{i+1}$ do

link t with input arcs to $input(t)$ in P_i

output arcs to $output(t)$ and link arcs to $link(t)$ in P_i

6. return $(\langle P_0, A_1, \mu A_1, \cdots, A_{i+1}, \mu A_{i+1}, P_{i+1}, \mu P_{i+1}\rangle)$

一般来讲,我们可以不停地按照算法 1 扩展诊断图,但是对于任何系统模型,其对应的诊断图都存在一个所谓的不动点层,如果按照算法 1 无限扩展诊断图,所得到的诊断图最终都将收敛到诊断图的不动点层.下面介绍诊断图不动点的定义和诊断图收敛定理.

定义 17.　所谓的诊断图不动点层是指从诊断图某一层开始,后续层的诊断图均是相等的,假如诊断图的不动点层为 k,则 $\forall i, i>k$,诊断图的第 i 层与诊断图的第 k 层均是相等的,也就是说,$P_i = P_k$, $\mu P_i = \mu P_k$, $T_i = T_k$, $\mu T_i = \mu T_k$.

定理 2.　对于任一诊断图,均存在一不动点层 k,该不动点层 k 是能使 $|P_{k-1}| = |P_k|$ 以及 $|\mu P_{k-1}| = |\mu P_k|$ 成立的最小的 k 值.

证明.　首先,我们要证明,对于任一诊断图,均存在不动点层.注意到,对于任何诊断问题,都有(1)在诊断问题中存在有限个命题,(2) $\forall i, P_{i-1} \subseteq P_i$,(3)如果 $(p,q) \notin \mu P_{i-1}$,则 $(p,q) \notin \mu P_i$.因此,命题层 P_i 中的命题数要么多于命题层 P_{i-1} 层命题数,要么 P_i 中的命题数与 P_{i-1} 命题数相等并且 P_i 中不一致命题对的数目等于或少于 P_{i-1} 中的不一致命题对的数目.由于诊断问题中命题数是有限的,因此这种单调递减的属性必然有界,也就是说,诊断图必然会在某一层达到 $(P_{i-1} = P_i)$ 和 $(\mu P_{i-1} = \mu P_i)$,此时,也就会有 $(T_i = T_{i+1})$ 且 $\mu T_i = \mu T_{i+1}$.

其次,假定 $|P_{k-1}| = |P_k|$ 且 $|\mu P_{k-1}| = |\mu P_k|$,我们还要证明第 k 层后所有的诊断图层均与第 k 层诊断图相等.注意到,由于 $|P_{k-1}| = |P_k|$,并且 $\forall i, P_{i-1} \subseteq P_i$,由此可以得到 $(P_{k-1} = P_k)$.由 $(P_{k-1} = P_k)$ 和 $|\mu P_{k-1}| = |\mu P_k|$,又可推得 $\mu P_{k-1} = \mu P_k$,因为由命题 1 可知,在 $k-1$ 层命题层一致的命题在 k 层命题层仍然是一致的.另外,由于 T_{k+1} 完全取决于 P_k 和 μP_k,因此,由 $(P_{k-1} = P_k)$ 和 $\mu P_{k-1} = \mu P_k$ 可推得 $T_{k+1} = T_k$,进而可得 $P_{k+1} = P_k$.显然,由于两元件工作模式集合 T_k 和 T_{k+1} 相等,并且它们有相同的输入不一致约束(因为 $\mu P_{k-1} = \mu P_k$),因此 $\mu T_{k+1} = \mu T_k$,进而推得 $\mu P_{k+1} = \mu P_k$.综合上述分析,可以得到诊断图第 $k+1$ 层与诊断图第 k 层是相等的.重复上述分析过程,容易得到对于所有 $i>k$,i 层诊断图均与 k 层诊断图相等.　　证毕.

定理 3.　对于模型中共有 n 个元件,诊断初始状态中有 p 个命题的诊断问题,假定该诊断问题中有 m 个元件工作模式,并且每个元件工作模式都有常数个形式参数.如果这些元件工作模式中最长的输出命题和连接命题数总和为 l,那么,运用上述构造诊断图的算法扩展得到的 t 层诊断图的大小与构造诊断图所需要的时间是关于 n, m, p, l 和 t 的多项式复杂的.

证明.　假定元件工作模式中具有最多的形式参数的数目为 k.由于对元件工作模式的使用不会产生新的系统元件,所以对元件工作模式进行例化最多能产生 $O(\ell n^k)$ 个不同的命题.因此,诊断图中任一命题层所能具有的最多命题数为 $O(p + m\ell n^k)$.此外,对于元件工作模式,最多有 $O(n^k)$ 种不同的例化方法,因此,诊断图中元件工作模式层最大的动作节点数为 $O(mn^k)$.由于 k 是常数,因此,诊断图的大小是关于 n, m, p, ℓ 和 t 多项式复杂的.

对于构造诊断图所需的时间,我们先来考察构造一层诊断图所需要的时间复杂度.构造新的一层诊断图所需要的时间主要由 3 部分组成:(1)根据诊断图前一层命题层中的命题,对所有可能的元件工作模式所需要的时间;(2)确定元件工作模式层中不一致的元件工作模式所需要的时间;(3)确定下一层命题层中不一致命题所需要的时间.显然,这些时间消耗是当前层诊断图中节点数目多项式复杂的.因此,构造 t 层诊断图所需的时间也是关于 n, m, p, ℓ 和 t 多项式复杂的.　　证毕.

4.3.2　诊断的计算

显然,对于某一诊断问题 P_D,我们可以根据前面介绍的算法构造诊断图,直到系统观察值 OBS 在诊断图上出现为止.当系统观察值 OBS 在诊断图上出现后,我们即可从诊断图出发,反向搜索所有可能导致系统观察值 OBS 的元件工作模式集合.在诊断图上进行解搜索需要充分地利用那些存在于命题和元件工作模式之间的不一致关系,以排除那些不可能的元件工作模式组合.诊断的过程是从系统观察值出发,沿诊断图反向逐层搜索可能导致系统观察值的元件工作模式的过程.因此,我们可以得到基于

诊断图的诊断计算算法.如算法 2 所示.首先算法输入诊断图 G,当前系统观察集合 OBS 以及诊断图的层数 i.算法通过递归调用子程序 Backward-Search,从诊断图元件工作模式层 T_i 中搜索能到达系统观察值 OBS 的元件工作模式集合 D_i(步 2).如果算法到达第 0 层,则算法返回空序列(步 1),并且将在紧接着到来的 Backward-Search 调用中返回诊断(步 3).

算法 2. 从诊断图上提取关于 OBS 的诊断.

Extract(G,OBS,i)

Input: the diagnostic graph G, the observation OBS, the level of diagnostic graph i

Output: the diagnosis D or fail

1. if $i=0$ then return($\langle\rangle$)
2. $D_i \leftarrow$ Backward-Search(G,OBS,$\varnothing$,i)
3. if $D_i \neq$ failure then return (D_i)
4. end

算法 3. 搜索到达观察值 OBS 所需元件工作模式 D_i 的算法.

Backward-Search(G,OBS,D_i,i)

Input: the diagnostic graph G, the observation OBS, the ith component working mode set, the level of diagnostic graph

Output: diagnosis D or fail

1. if $OBS=\varnothing$ then do
2. $D \leftarrow$ Extract(G,$\bigcup\{input(t) \mid \forall t \in D_i\}$,$i-1$)
3. if $D=$failure then return (failure)
4. return ($D \cdot \langle D_i \rangle$)
5. else do
6. select any $p \in OBS$
7. $resolvers \leftarrow \{t \in T_i \mid p \in output(t) \bigcup link(t)$ and $\forall t' \in D_i : (t,t') \notin \mu T_i$
8. if $resolvers=\varnothing$ then return (failure)
9. nondeterministically choose $t \in resolvers$
10. return(Backward-search(G,$OBS-output(t)-link(t)$,$D_i \bigcup \{t\}$,i))
11. end

4.3.3 基于诊断图的诊断算法

在分别介绍了诊断图构造算法和诊断搜索算法后,下面给出基于诊断图的诊断算法的整体描述(算法 4).

首先,将诊断图层数 i 置 0,并将诊断问题的初始状态插入诊断图的第 0 层命题层(步 1～2).然后,根据算法 3.1 描述的诊断图扩展算法逐层扩展诊断图,直到到达诊断图的不动点层(步 3～6).如果在诊断图的不动点层,系统观察值不能被达到,或者观察值不能一致地达到,则算法失败返回(步 7).当观察值能一致地达到时,开始调用算法 2 描述的诊断计算算法进行诊断的搜索过程(步 8),并最终返回系统诊断(步 9).

定理 4. 算法 4 所描述的基于诊断图的诊断算法是可靠的.

证明. 首先考虑算法返回一个称为诊断的元件工作模式集合 $D=\{D_1,\cdots,D_n\}$ 的情况,根据算法 2 中的诊断计算算法,我们可知上述元件工作模式集合,D_i 中的元件工作模式应是相互一致的,并且元件工作模式集合 D_n 的输出或连接集合中包含有系统观察值 OBS,而元件工作模式集合 D_{n-1} 的输出或连接集合中包含 $input(D_n)$,如此类推,D 中前一层的元件工作模式集合的输出或连接集合中包含后一层的元件工作模式集合的输入集合,最终元件工作模式集合 D_1 的输入集合 $input(D_1)$ 将在诊断图的 P_0 层命题层(即诊断问题的初始状态中)中被满足.因此元件工作模式集合 D 满足诊断问题 $P_D=\langle s_0,M,OBS\rangle$ 的诊断的定义,因此,D 即为诊断问题 P_D 的诊断. 证毕.

算法 4. 基于诊断图的诊断算法整体描述.

Diagnosis(T,s_0,OBS)

Input: the set of component working modes T, the initial state s_0, the observation OBS

Output: the diagnosis D or failure

1. $i \leftarrow 0$, $P_0 \leftarrow s_0$
2. $G \leftarrow \langle P_0 \rangle$
3. do
4. $i \leftarrow i+1$
5. $G \leftarrow$ Expand(G)
6. while(Fixedpoint(G))
7. if $OBS \subset P_i$ or $OBS^2 \bigcap \mu P_i \neq \varnothing$ then return(failure)
8. $D \leftarrow$ Extract(G,g,i)
9. return (D)
10. end

5 实例分析

本小节以前面所介绍的全加器为例,说明基于诊断图的诊断算法的工作过程.

在图 2 表示的全加器的例子中,系统的观察值为 $OBS=\{out(X2)=1,out(O1)=0\}$,显然,该观察值能在诊断图 P_3 层上一致地出现.因此,我们开始从 P_3 反向搜索导致观察值的元件工作模式.从图 3 的诊断图中可以发现,在诊断图的元件工作模式层

T_3中能导致观察值的元件工作模式 NR_1($O1$)和 NO-MODE($out(X2)=1$),这两个元件工作模式是一致的,由此可得到 $D_3=\{$NR_1($O1$), NO-MODE($out(X2)=1$)$\}$.

现在到达了第 2 层,此时 D_3中的元件工作模式的输入$\{in1(O1)=0, in2(O1)=0, out(X2)=1\}$变成了系统观察值. 在 T_2层元件工作模式层中,以 $out(X2)=1$ 为输出的元件工作模式有 AB_4($X2$)和 NR_2($X2$). 产生 $in1(O1)=0$ 的元件工作模式有 AB_4($A2$)和 NR_3($A2$). 产生 $in2(O1)=0$ 的元件工作模式有 NO-MODE($in2(O1)=0$). 上述元件工作模式中,AB_4($X2$)和 NR_3($A2$)是不一致的,因为这两个工作模式具有不一致的输入. 同理,NR_2($X2$)和 AB_4($A2$)是不一致的. 由此,可得到

$$D_2=\{\text{NR\_2}(X2), \text{NR\_3}(A2), \text{NO-MODE}(in2(o1)=0)\}.$$

此时,$D2$ 中元件工作模式的输入集合$\{in1(X2)=0, in2(X2)=1, in1(A2)=1, in2(A2)=0, in2(O1)=0\}$变成了第 1 层的观察值. 这些观察值中,$in1(X2)=0$ 和 $in2(A2)=0$ 可由 T_1层元件工作模式层中的 AB_3($X1$)得到,$in2(O1)=0$ 可由 NR_3($A1$)得到. 而 $in1(A2)=1$ 和 $in2(X2)=1$ 可由对应的伪元件工作模式得到. 由此,可以得到

$$D_1=\{\text{AB\_3}(X1), \text{NR\_3}(A1), \text{NO-MODE}(in1(A2)=1), \text{NO-MODE}(in2(X2)=1)\}.$$

显然,D_1中所有的元件工作模式的输入集合均出现在命题层 P_0层,即这些输入能在诊断问题的初始状态中被满足,因此,算法成功地找到了系统的一个诊断,即 $D=\{D_1, D_2, D_3\}$.

将 D_1, D_2, D_3中元件工作模式进行适当化简,删除其中的表示元件正常工作的元件工作模式以及用于在命题层之间传递命题的伪元件工作模式,即可得到最终出故障的元件,即 $D=\{$AB_3($X1$)$\}$,这就是系统的诊断.

6　我们的诊断技术在卫星领域中的应用

为了说明我们的诊断技术的有效性,我们选取了 IPC2002 规划大赛上的 Satellite 问题作为我们的测试例子. Satellite 问题的任务是要求用航天器上的摄像机拍摄卫星图像,并将这些高质量的星云图像传回地面. 为了拍摄到目标卫星图像,航天器首先必须转动适当角度(动作 turn_to),使安装在航天器上的摄像机镜头能指向待拍摄的目标,然后打开摄像机(动作 switch_on),并进行校准工作(动作 calibrate),最后进行卫星图像拍摄(动作 take_image). 对于远程 Agent 的体系结构,可简单地用图 5 进行表示. 对于给定的任务,远程 Agent 系统通常能够调用相应的规划器,求得能完成相应任务的规划动作序列,然后,系统会将产生的规划动作序列提交给系统的执行部件,实现系统的状态迁移,并使系统最终达到目标状态. 然而,当系统出现某些故障的时候,往往会导致任务的失败,具体表现在系统并不能如预期那样成功到达目标状态,这就形成了所谓的诊断问题,我们对于诊断问题的描述,主要分为两个部分,一是诊断模型的描述,一是系统观察值的描述. 为了能运用我们的图诊断方法进行诊断,需要根据规划领域,给出异常情况下的各种工作模式,在这里对应各种异常情况下的动作模型,为此,我们引入两个谓词 NR 和 AB,谓词 NR 表示正常情况下的动作模型,而 AB 则表示出故障时的动作模型. 同时,引入 fail 谓词,表示各种故障现象. 例如,对于动作 turn_to,我们有以下两个动作模型:

```
NR_turn_to
    :parameters(?s-satellite ?d_new-direction
                ?d_prev-direction)
    :input (and (pointing ?s ?d_prev)
                (not (=?d_new ?d_prev))
           )
    :output (and (pointing ?s ?d_new)
                 (not (pointing ?s ?d_prev))
            )
AB1_turn_to
    :parameters(?s-satellite ?d_new-direction
                ?d_prev-direction)
    :input (and(pointing ?s ?d_prev)
           (not(=?d_new ?d_prev))
           )
    :output (OR(fail_pointing ?s ?d_new)
                (pointing ?s ?d_prev)
            )
)

AB2_turn_to
    :parameters(?s-satellite ?d_new-direction
                ?d_prev-direction)
    :input (OR(fail_pointing ?s ?d_prev)
               (=?d_new ?d_prev)
           )
    :output (OR(fail_pointing ?s ?d_new)
                (pointing ?s ?d_prev)
            )
```

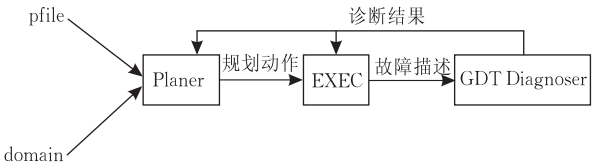

图 5 自主 Agent 的体系结构

上述动作模型中，NR_turn_to 动作模型表明，在正常情况下执行动作 turn_to 后，航天器将指向新的方向，即(pointing ?s ?d_new)，此时，原来的(pointing ?s ?d_prev)不再成立，即(not (pointing ?s ?d_prev))．而在故障动作模型 AB1-turn_to 作用下，航天器的结果状态可能有以下两种情况发生，要么航天器并没有按预期那样指向新的方向，即(fail_pointing ?s ?d_new))(比如可能指向其他方向)，要么航天器仍然停留在原来的状态下，即(pointing ?s ?d_prev)．故障模式 AB2_turn_to 表明，执行动作 turn_to 时，航天器本来就没有指向 d_prev 方向，即(fail_pointing ?s ?d_prev)，或者，d_new 和 d _ prev 本来就是同一个方向，即(=?d_new ?d_prev)，因此，就不可能得到正常情况下的结果状态．类似地，我们可以按照上述方式设计其他动作，比如 switch_on，swith_off，calibrate 以及 take_image 等动作的正常模型和异常模型，从而形成故障诊断所需要的模型．

有了诊断模型的描述文件后，我们即可根据诊断问题的观察值，运用我们的图诊断工具 GDT 进行诊断．，如果通过系统观察值，我们发现，系统并没有到达我们预期的规划目标状态，此时就产生了诊断问题．由于条件限制，我们没法完全模拟 Agent 实际运行时执行规划的情况，因此，我们通过手工修改规划目标状态中的谓词，比如原来规划目标中要求：

```
(:goal (and (have_image Phenomenon4 thermograph0)
            (have_image Star5 thermograph0)
            (have_image Phenomenon6 thermograph0)
       )
)
```

即原来规划目标中要求能拍摄到 Phenomenon4，Star5 和 Phenomenon6 方向的 thermograph0 图像，但执行完规划动作后，我们通过观察发现，航天器上的摄像机没能成功地拍摄到 Phenomenon4 方向的 thermograph0 图像，即此时系统观察值为

```
(:OBS(and (fail_have_image Phenomenon4 thermograph0)
          (have_image Star5 thermograph0)
          (have_image Phenomenon6 thermograph0)
     )
)
```

此时，表明系统在某个方面存在故障，由此产生了所谓的诊断问题．我们根据 IPC2002 规划大赛上的 pfile1～pfile10 这 10 个 Satellite 规划问题，分别设计了 10 个诊断问题，对应为 obsfile1～obsfile10，如表 2 所示．有关 Satellite 诊断问题的详细的模型描述文件以及相应的故障描述文件，限于篇幅，本文无法详细列出．有了系统的模型描述和故障问题描述文件后，我们就可将这 10 个诊断问题连同系统模型一起，形成故障描述文件，输入到我们的图诊断工具 GDT 中进行求解．最终，GDT 将返回那些带故障标识谓词 AB 的异常动作模型，作为系统的诊断解．我们的实验系统都是在 Linux 平台下进行的，测试平台为 CPU(C Ⅱ 850MHz)＋RAM(128MB)＋Redhat9．0 的 Linux 平台，编译器为 Redhat9．0 自带的 gcc．

表 3 显示的是用 GDT 对这 10 个问题进行诊断的求解结果．下面我们来分析一下我们求得的诊断结果，看是否能有效地解释为何未能按预期达到我们的规划目标．

表 2 根据 pfile1～pfile10 所表示的规划问题设计的诊断问题(fail 开始的谓词表示故障)

Planning Problem	Planning Goal	Diagnostic Problem	Observations
pfile1	(have_image Phenomenon4 thermograph0) (have_image Star5 thermograph0) (have_image Phenomenon6 thermograph0)	obsfile1	(fail_have_image Phenomenon4 thermograph0)
pfile2	(have_image Planet3 infrared0) (have_image Planet4 infrared0) (have_image Phenomenon5 image2) (have_image Phenomenon6 infrared0) (have_image Star7 infrared0)	obsfile2	(fail_have_image Planet3 infrared0) (fail_have_image Star7 infrared0)
pfile3	(pointing satellite0 Phenomenon5) (have_image Star3 infrared0) (have_image Star4 spectrograph2) (have_image Phenomenon5 spectrograph2) (have_image Phenomenon7 spectrograph2)	obsfile3	(fail_pointing satellite0 Phenomenon5) (fail_have_image Phenomenon5 spectrograph2)

（续　表）

Planning Problem	Planning Goal	Diagnostic Problem	Observations
pfile4	(pointing satellite1 Planet5) (have_image Planet3 infrared1) (have_image Star4 infrared1) (have_image Planet5 thermograph2) (have_image Star6 infrared1) (have_image Star7 infrared0) (have_image Phenomenon8 thermograph2) (have_image Phenomenon9 infrared0)	obsfile4	(fail_pointing satellite1 Planet5) (fail_have_image Star6 infrared1)
pfile5	(pointing satellite0 Phenomenon5) (pointing satellite1 GroundStation2) (have_image Star3 thermograph0) (have_image Phenomenon5 image2) (have_image Phenomenon6 image2) (have_image Star7 thermograph0) (have_image Phenomenon8 image2) (have_image Planet9 spectrograph1)	obsfile5	(fail_pointing satellite1 GroundStation2) (fail_have_image Phenomenon5 image2) (fail_have_image Planet9 spectrograph1)
pfile6	(have_image Planet4 thermograph2) (have_image Planet5 spectrograph0) (have_image Star6 thermograph2) (have_image Star7 infrared3) (have_image Phenomenon8 spectrograph0) (have_image Star9 infrared1) (have_image Star10 infrared3)	obsfile6	(fail_have_image Star6 thermograph2) (fail_have_image Phenomenon8 spectrograph0) (fail_have_image Star10 infrared3)
pfile7	(pointing satellite1 Star1) (pointing satellite2 Phenomenon5) (have_image Phenomenon5 image0) (have_image Star6 image1) (have_image Star7 image0) (have_image Planet8 image0) (have_image Planet9 image3) (have_image Planet10 image0) (have_image Planet11 image2)	obsfile7	(fail_pointing satellite2 Phenomenon5) (fail_have_image Planet8 image0)
pfile8	(have_image Phenomenon5 thermograph1) (have_image Star6 thermograph1) (have_image Star7 spectrograph3) (have_image Phenomenon8 image0) (have_image Phenomenon9 image0) (have_image Star10 spectrograph3) (have_image Planet11 thermograph2) (have_image Phenomenon12 image0) (have_image Phenomenon13 thermograph1) (have_image Phenomenon14 thermograph2)	obsfile8	(fail_have_image Phenomenon5 thermograph1)
pfile9	(pointing satellite0 Phenomenon7) (pointing satellite3 Star9) (pointing satellite4 Planet5) (have_image Planet5 image2) (have_image Phenomenon6 image3) (have_image Phenomenon7 infrared1) (have_image Phenomenon8 image2) (have_image Star9 image3) (have_image Planet10 image4) (have_image Planet11 spectrograph0) (have_image Phenomenon12 image3) (have_image Phenomenon13 spectrograph0) (have_image Star14 image4)	obsfile9	(fail_pointing satellite0 Phenomenon7)
pfile10	(pointing satellite4 Planet9) (have_image Planet5 image4) (have_image Star6 infrared3) (have_image Star7 image4) (have_image Phenomenon8 image4) (have_image Planet9 infrared0) (have_image Planet10 infrared3) (have_image Star12 image4) (have_image Phenomenon13 image4) (have_image Phenomenon14 spectrograph1) (have_image Star15 spectrograph1) (have_image Star16 image2)	obsfile10	(fail_have_image Phenomenon8 image4) (fail_have_image Star12 image4) (fail_have_image Star15 spectrograph1)

表 3 用 GDT 诊断工具对诊断问题进行求解的结果(solution 表示返回的诊断,time 表示求解时间)

Problem	Diagnosis solution	time/s
obsfile1	(AB2_SWITCH_ON INSTRUMENT0 SATELLITE0) (AB6_TAKE_IMAGE SATELLITE0 PHENOMENON4 INSTRUMENT0 THERMOGRAPH0)	0.10
obsfile2	(AB1_TURN_TO SATELLITE0 STAR7 PLANET4) (AB7_TAKE_IMAGE SATELLITE0 STAR7 INSTRUMENT1 INFRARED0) (AB4_CALIBRATE SATELLITE0 INSTRUMENT0 PLANET4) (AB2_TAKE_IMAGE SATELLITE0 PLANET3 INSTRUMENT0 INFRARED0)	0.05
obsfile3	(AB2_SWITCH_ON INSTRUMENT2 SATELLITE0) (AB2_TURN_TO SATELLITE0 PHENOMENON5 STAR4) (AB6_TAKE_IMAGE SATELLITE1 PHENOMENON5 INSTRUMENT2 SPECTROGRAPH2)	0.05
obsfile4	(AB2_SWITCH_ON INSTRUMENT2 SATELLITE1) (AB1_TURN_TO SATELLITE1 PLANET5 STAR2) (AB6_TAKE_IMAGE SATELLITE1 STAR6 INSTRUMENT2 INFRARED1)	0.09
obsfile5	(AB1_SWITCH_ON INSTRUMENT6 SATELLITE2) (AB6_TAKE_IMAGE SATELLITE2 PHENOMENON5 INSTRUMENT6 IMAGE2) (AB1_TURN_TO SATELLITE0 PLANET9 PHENOMENON8) (AB2_TURN_TO SATELLITE1 GROUNDSTATION2 PHENOMENON8) (AB7_TAKE_IMAGE SATELLITE0 PLANET9 INSTRUMENT8 SPECTROGRAPH1)	0.11
obsfile6	(AB2_SWITCH_ON INSTRUMENT1 SATELLITE1) (AB6_TAKE_IMAGE SATELLITE2 PHENOMENON8 INSTRUMENT1 SPECTROGRAPH0) (AB1_TAKE_IMAGE SATELLITE1 STAR6 INSTRUMENT2 THERMOGRAPH2) (AB6_TAKE_IMAGE SATELLITE2 STAR10 INSTRUMENT1 INFRARED3)	0.11
obsfile7	(AB1_TURN_TO SATELLITE2 PHENOMENON5 STAR6) (AB1_TURN_TO SATELLITE3 PLANET8 PLANET10) (AB7_TAKE_IMAGE SATELLITE3 PLANET8 INSTRUMENT7 IMAGE0)	0.14
obsfile8	(AB1_CALIBRATE SATELLITE1 INSTRUMENT4 STAR4) (AB2_TAKE_IMAGE SATELLITE3 PHENOMENON5 INSTRUMENT4 THERMOGRAPH1)	0.28
obsfile9	(AB2_TURN_TO SATELLITE0 PHENOMENON7 STAR0)	0.45
obsfile10	(AB2_SWITCH_ON INSTRUMENT7 SATELLITE3) (AB6_TAKE_IMAGE SATELLITE4 STAR15 INSTRUMENT7 SPECTROGRAPH1) (AB6_TAKE_IMAGE SATELLITE4 STAR12 INSTRUMENT7 IMAGE4) (AB1_TURN_TO SATELLITE2 PHENOMENON8 PLANET9) (AB7_TAKE_IMAGE SATELLITE2 PHENOMENON8 INSTRUMENT10 IMAGE4)	0.40

在 obsfile1 中,观察值为(fail_have_image Phenomenon4 thermograph0),也就是说,系统在执行完规划动作后,发现并没有按预期拍摄到 Phenomenon4 方向的 thermograph0 图像. 对于该诊断问题,GDT 返回的诊断结果是(AB2_SWITCH_ON INSTRUMENT0 SATELLITE0)和(AB6_TAKE_IMAGE SATELLITE0 PHENOMENON4 INSTRUMENT0 THERMOGRAPH0). 为了搞清楚这两个诊断结果的物理含义,我们将上述诊断结果展开如下:

```
AB2_switch_on(INSTRUMENT0 SATELLITE0)
    :input (and (on_board INSTRUMENT0 ?s)
                (power_avail SATELLITE0)
           )
    :output(and (fail_power_on INSTRUMENT0)
                (not (calibrated INSTRUMENT0))
                (power_avail SATELLITE0)
           )
AB6_take_image (SATELLITE0 PHENOMENON4
                INSTRUMENT0 THERMOGRAPH0)
    :input(fail_power_on INSTRUMENT0)
    :output(fail_have_image PHENOMENON4
            THERMOGRAPH0)
```

显然,从这两个诊断结果中,我们可以知道,导致(fail_have_image Phenomenon4 thermograph0)的原因是因为用航天器中的仪器 INSTRUMENT0 进行拍摄的前提条件不满足,即动作 take_image 的一个输入 input=(fail_power_on INSTRUMENT0),进行拍摄的仪器 INSTRUMENT0 没有处于就绪状态. 而导致(fail_power_on INSTRUMENT0)的原因是因为动作 switch_on 出现了故障. 类似地,我们可以对其它诊断问题的诊断结果进行分析,限于篇幅,本文在此不一一赘述. 显然,这些分析结果表明,GDT 返回的诊断能合理地解释出现故障的原因.

7 相关工作

基于模型的诊断研究开始于 20 世纪 70 年代中期,其开创性的工作是 de Kleer 的 INTER[19] 系统.

到 80 年代中期这个领域热了起来,因基于模型的诊断具有诸多优点及很高的实用价值,吸引了许多研究者,到 90 年代已经成为一个十分活跃的研究分支. 基于模型的诊断对整个人工智能领域的研究起着重要推动作用. 一方面,人工智能领域其他分支的许多研究成果可以在基于模型的诊断中得到有效的应用,促进了这一分支的发展. 另一方面,基于模型的诊断研究暴露了许多人工智能技术中存在的问题,激发了人们对它们研究的热情. 当前在基于模型的诊断中有两个主要流派,一个是基于一致性的诊断(consistency-based diagnosis)[1-4],另一个是溯因诊断(abduction diagnosis)[5-7]. 前者把诊断概念建立在不正常工作的元件与观察结果的相容性上,这相当于解释的弱概念,持这种观点的代表人物是 de Kleer[19]. 后者把诊断概念建立由不正常工作的元件和诊断模型可以逻辑地推出观察结果,这相当于解释的强概念,持这种观点的代表人物是 Poole[5]. 最近,人们开始研究如何将两种方法比较与融合. Console 和 Friedrich[20] 把诊断问题看成是带有一致性约束的溯因问题. 他们提出了一个统一的定义,使得不同的诊断逻辑定义都可以用这个统一的定义表示,并且可以把不同的定义加以比较. 在这个统一定义下,基于一致性的诊断和溯因诊断仅仅是两个极端的情形.

我们的基于模型的诊断可以看作是溯因诊断的一种,因为对于任何不正常的观察结果,通过我们的诊断方法,都能找到可能出故障的元件,并给出导致系统异常观察合理解释. 但我们的方法与传统的基于模型的诊断方法又存在着本质的不同,传统的基于模型的诊断方法是基于逻辑推理的,因此其诊断计算方法要使用到定理证明器,这导致传统的基于模型诊断方法的效率过于低下. 事实上,基于模型诊断方法自提出以来,就由于其具有很多优良性质而吸引了众多研究者对其进行研究,但基于模型诊断效率过于低下这一问题却一直困扰着相关研究者们,成为模型诊断研究领域公认的而又不得不面对的难题. 我们认为基于模型诊断效率难以提高的根本原因在于原来的关于诊断的计算过多地依赖定理证明器的使用. 基于这一认识,我们提出了重新修正原来的关于模型的描述方法,并给出了我们的用诊断图分析进行诊断的方法,显然,我们的方法属于一种过程化的方法,这使得我们的方法能有效地克服经典模型诊断中效率过于低下的难题.

熟悉规划领域的读者不难发现,我们的基于诊断图分析的诊断方法与规划领域中用规划图[21-22]分析进行规划的方法非常类似. 事实上,智能规划的研究初期也主要是采用逻辑推理的手段进行研究,智能规划系统的前身 QA3 系统就是一个典型的逻辑系统. 但后来,规划领域研究者们发现,单纯用逻辑推理的方法来求解规划的效率是非常低下的,远不能满足求解实际规划问题的需要. 此后,Fikes 和 Nilsson 提出了 STRIPS[23] 规划描述方法,并设计了 STRIPS 规划系统,这标志着规划领域研究者们开始逐渐抛弃纯逻辑的手段来研究规划的路线,改用一种过程化的方法来研究规划问题,以期提高规划效率. 在 STRIPS 规划系统的影响下,规划系统的效率较之前的基于逻辑推理的规划系统有了极大的提高,并且规划研究领域得到了迅猛发展. 各种成熟高效的规划系统不断涌现,基于规划图分析的图规划系统更是在原来规划效率有了较大提高的基础上,又将规划效率提高了几个数量级. 此后,智能规划研究领域逐渐形成了统一的规划描述语言,并举行了规划领域的比赛,极大地促进了规划技术的发展. 然而,与蓬勃发展的规划技术相比,智能诊断研究领域尚停留在单纯依靠逻辑推理进行研究的阶段,这导致智能诊断的效率难以得到有效提高. 我们的工作则在过程化方法求解诊断方面进行了尝试,以期能摆脱诊断研究领域的困境,推动智能诊断研究的发展.

8　小　结

我们在 Reiter 的基于模型诊断的基础上,重新定义了诊断. 在此基础上,我们提出了一种基于诊断图分析的快速诊断方法. 我们的诊断过程分成两个阶段,首先是根据系统的模型构造诊断图,然后在诊断图上根据系统观察值反向计算诊断. 我们的方法是一种过程化的方法,这与传统的模型诊断方法有着本质的不同. 我们已经在 Graphplan 和 LPG 规划器的基础上,设计并实现了基于我们方法的图诊断工具(GDT),初步实验结果表明了我们方法的有效性. 当然,我们的结果也仅仅是初步的,比如,在我们的方法下,每次诊断求解只能产生一个诊断解. 此外,在诊断领域,一个令人感兴趣的研究问题是关于极小诊断和核心诊断问题的讨论以及诊断测试和诊断修复问题的讨论,我们对这一部分的工作尚未来得及开展,这将会是我们下一步研究工作的重点.

参 考 文 献

[1] Reiter R. A theory of diagnosis from first principles. Artificial Intelligence, 1987, 32(1): 57-96

[2] Davis R. Diagnostic reasoning based on structure and behavior. Artificial Intelligence, 1984, 24(3): 347-410

[3] Kleer D J, Williams B C. Diagnosing multiple faults. Artificial Intelligence, 1987, 32(1): 97-130

[4] Genesereth M R. The use of design descriptions in automated diagnosis. Artificial Intelligence, 1984, 24(1-3): 411-436

[5] Poole D, Goebel R, Aleliunas R. THEORIST: A logic reasoning system for defaults and diagnosis//Cercone N J, McCalla G eds. The Knowledge Frontier: Essays in the Representation of Knowledge. New York: Springer Verlag, 1987: 331-352

[6] Console L, Theseider Dupre D, Torasso P. A theory of diagnosis for incomplete causal models//Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, 1989: 1311-1317

[7] Cox P T, Pietrykowski T. General diagnosis by abductive inference//Proceedings of the 1987 Symposium on Logic Programming. San Francisco, California, 1987: 183-189

[8] Console L, Torasso P. A spectrum of logical definitions of model-based doagnosis. Computational Intelligence, 1991, 7(3): 133-141

[9] Chittaro Luca. Hierarchical model-based diagnosis based on structural abstraction. Artificial Intelligence, 2004, 155(1-2): 147-182

[10] Baroni Pietro, Lamperti Gianfranco, Pogliano Paolo, Zanella Marina. Diagnosis of large active systems. Artificial Intelligence, 1999, 110(1): 135-183

[11] Li Zhan-Shan, Jiang Yun-Fei, Wang Tao. The decomposition for model-based diagnosis problem and its algorithm. Chinese Journal of Computers, 2003, 26(9): 1177-1182(in Chinese)
(李占山, 姜云飞, 王涛. 基于模型的诊断问题分解及其算法. 计算机学报, 2003, 26(9): 1177-1182)

[12] Ouyang Dan-Tong, Jiang Yun-Fei. Model-based diagnosis of a general causal theory. Journal of Computer Research and Development, 1999, 36(1): 31-35(in Chinese)
(欧阳丹彤, 姜云飞. 广义因果理论的基于模型的诊断. 计算机研究与发展, 1999, 36(1): 31-35)

[13] Chen Rong, Jiang Yun-Fei. Model-based diagnosis system with constraints. Chinese Journal of Computers, 2001, 24(2): 127-135(in Chinese)
(陈荣, 姜云飞. 含约束的基于模型的诊断系统. 计算机学报, 2001, 24(2): 127-135)

[14] FattahY E, Dechter R. Diagnosing tree-decomposable circuits//Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95). Montréal, Québec, Canada, 1995: 572-578

[15] Stumptner M, Wotawa F. Diagnosing tree-structured systems. Artificial Intelligence, 2001, 127(1): 1-29

[16] Luan Shang-Min, Dai Guo-Zhong. An approach to diagnosing a system with structure information. Chinese Journal of Computers, 2005, 28(5): 801-808(in Chinese)
(栾尚敏, 戴国忠. 利用结构信息的故障诊断方法. 计算机学报, 2005, 28(5): 801-808)

[17] Mozetic I. A polynomial-time algorithm for model-based diagnosis//Proceedings of the 10th European Conference on Artificial Intelligence (ECAI 92). Vienna, Austria, 1992: 729-733

[18] Childress R L, Valtorta M. Polynomial-time model-based diagnosis with the critical set algorithm//Proceeding of the 4th International Workshop on Principles of Diagnosis. University of Wales, Aberystwith, 1993: 166-177

[19] de Kleer J, Mackworth A K, Reiter R. Charactering diagnoses and systems. Artificial Intelligence, 1992, 56(2-3): 197-222

[20] Console L, Friedrich G et al. Model-Based Diagnosis. Basel Switzerland: Science Publishers, 1994

[21] Blum Avrim L, Merrick L. Fast planning through planning graph analysis. Artificial Intelligence, 1997, 90(1-2): 281-300

[22] Jiang Yun-Fei, Chen Ai-Xiang et al. Automated Planning: Theory and Practice (Translated version in Chinese). Beijing: Tsinghua University Press, 2008(in Chinese)
(姜云飞, 陈蔼祥等译. 自动规划:理论与实践. 北京: 清华大学出版社, 2008)

[23] Fikes R, Nilsson N. STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence, 1971, 2(3): 189-203

CHEN Ai-Xiang, born in 1978, Ph. D., lecturer. His research interests include artificial intelligence and planning.

CHEN Qing-Liang, born in 1980, Ph. D., lecturer. His research interests include artificial intelligence and reasoning.

PAN Jiu-Hui, born in 1956, professor, Ph. D. supervisor. His research interests include artificial intelligence and knowledge engineering.

JIANG Yun-Fei, born in 1945, professor, Ph. D. supervisor. His research interests include artificial Intelligence and planning.

YANG Jin-Ji, born in 1968, associate professor. His research interests include artificial intelligence and reasoning.

Background

Informally, diagnosis can be defined as the task of explaining why a given physical system does not exhibit its nominal behavior. Due to its generality, its dramatic importance in many application domains, and its intrinsic complexity, automated diagnosis has received constant and considerable attention in AI research. Formally speaking, given the logical model of system and its input, when an observation of system's behavior conflicts with the way the system meant to behave, we can determine those components of the system which, when assumed to be functioning abnormally, will explain the discrepancy between the observed and correct system behavior. That is the main idea of model-based diagnosis. However the classic diagnostic method is very inefficient because its computing procedure is based on logical reasoning. We in this paper redefine the idea of diagnosis based on the original concept of Reiter's model-based diagnosis. And then we introduce a fast diagnosis method by the data structure of diagnostic graph. In fact, our model is procedure-theoretic rather than logic-theoretic. In our diagnosis model, we have two phases in the processing, first we construct the diagnostic graph according to our system model, and then we implement computations on the graph conversely by the system observations. This is obvious a procedural approach that differs inherently from the traditional pure logical one. We have implemented our method based on Graphplan and LPG and have our tool GDT (Graph Diagnosis Tool). The preliminary experimental results justify the effectiveness of our method.

This work are supported by the National Basic Research Program (973 Program) of China under grant No. 2005CB321902; the National Natural Science Foundation of China under grant No. 60773201; the International Joint Research Project of National Science Foundation under grant No. 60911130005; the Start-up Research Fund for Introduced Talents in Jinan University; the Distinguished Young Researcher Nurturing Program in Universities of Guangdong No. LYM08017; the Research Foundation of Science and Technology Plan Project in Guangdong Province of China under grant No. 2007B010400068 and the Industry, Education and Research Joint Research Fund of Guangdong Province under grant No. 2007B090400095.