ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.2, February 2010, pp.248−260
doi: 10.3724/SP.J.1001.2010.03793

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于动态描述逻辑的网构软件系统故障诊断*

王竹晓[1,2+], 杨 鲲[1,2], 史忠植[1]

[1](中国科学院 计算技术研究所 智能信息处理重点实验室,北京 100190)

[2](中国科学院 研究生院,北京 100049)

Failure Diagnosis of Internetware Systems Using Dynamic Description Logic

WANG Zhu-Xiao[1,2+], YANG Kun[1,2], SHI Zhong-Zhi[1]

[1](Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100190, China)

[2](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: E-mail: wangzhuxiao@ics.ict.ac.cn

Wang ZX, Yang K, Shi ZZ. Failure diagnosis of Internetware systems using dynamic description logic. *Journal of Software*, 2010,21(2):248−260. http://www.jos.org.cn/1000-9825/3793.htm

Abstract: This paper proposes a way to analyze diagnosed systems using dynamic description logic. Syntax and semantics of dynamic description logic are suitable to describe both the normal and the failed behavior of the system. Then, it gives algorithms to test diagnosability of discrete-event systems by using dynamic description logic satisfiability-checking and for the construction of a diagnoser, which performs diagnostics using on-line observations of the system behavior. Throughout the paper, examples are given for illustration.

Key words: failure diagnosis; Internetware; dynamic description logic; trustworthy software; discrete-event system

摘 要: 提出了一种应用动态描述逻辑对被诊断系统进行分析的方法.动态描述逻辑的语法和语义非常适宜刻画系统的正常行为和异常行为.给出了算法使用动态描述逻辑的可满足性检测来测试离散事件系统的可诊断性.同时,给出了故障诊断器的构造算法.该故障诊断器通过实时地观察系统的行为完成故障诊断.以例子对算法进行了说明.

关键词: 故障诊断;网构软件;动态描述逻辑;可信软件;离散事件系统

中图法分类号: TP311 **文献标识码**: A

网构软件(Internetware)是为了适应开放、动态、多变的网络环境的发展趋势所呈现出的一种柔性可演化、多目标适应、连续反应式的新的软件系统形态[1−3].从技术角度来看,其形态应该是在软件构件等技术支持下的软件实体以Agent的软件服务形式存在于互联网的各个节点之上,各个软件实体相互间通过各种协同机制进行

* Supported by the National Natural Science Foundation of China under Grant No.60775035 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z132 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant Nos.2003CB317004, 2007CB311004 (国家重点基础研究发展计划(973)); the National Key Technology R&D Program of China under Grant No.2006BAC08B06 (国家科技支撑计划)

Received 2009-06-16; Revised 2009-09-11; Accepted 2009-12-07

跨网络的互连、互通、协作和联盟,从而形成一种与当前的信息Web相类似的软件Web(software Web).这样一种软件Web应能感知外部网络环境的动态变化,并随着这种变化按照某种指标进行调整和演化.

随着软件应用领域的不断拓展,越来越多的复杂技术系统借助软件来控制.人们更加注重软件系统的正确性(correctness)、安全性(safety)、性能(performance)、可靠性(reliability)、可用性(availability)、保密性(security)、隐私性(privacy)[4,5]等可信性质.

可信软件系统是稳定的具有容错功能的软件系统.容错功能(fault tolerance)是指在出现错误的情况下,软件系统依然能够提供正确的服务.容错技术在系统开发阶段实现,在系统的运行阶段发挥作用,从而提高了整个计算机系统的可靠性.软件系统容错功能实现时包括两个阶段:故障诊断阶段和系统恢复阶段.

软件系统故障诊断技术是构建可信软件系统的关键技术之一.软件的故障诊断问题也越来越受到人们的重视.通过故障诊断,自动发现系统的运行故障,确定故障的类型、部位,从而为进一步的系统恢复(例如软件自恢复)提供支持,是保证高可信软件系统的关键技术.本文主要研究基于动态描述逻辑的网构软件系统故障诊断.

系统故障是指背离系统正常和预期的行为[6].检测出系统中的故障,并将其从系统中分离出来的过程称为故障诊断(diagnosis)[7].随着复杂技术系统性能和可靠性要求的不断增加,故障诊断已在可靠性技术、控制、计算机科学等领域引起人们的广泛关注.大型复杂软件系统固有的复杂性,使其中的故障诊断问题十分复杂,相关的研究进展十分缓慢.由于能够在一定粒度上观测基于网构软件系统的行为,网构软件系统给软件系统的故障诊断问题带来了新的机遇和挑战.实践表明,动态系统中的故障诊断问题可以看成是离散事件系统(discrete-event system,简称DES)的状态估计和推理的问题[8−18].但是现有的故障诊断方法往往不能有效地描述软件系统的语义.

作为一类用于知识表示的形式化工具,史忠植等人[19]提出了一种动态描述逻辑(dynamic description logics),将描述逻辑与动态逻辑以及动作理论有机地结合起来,在描述逻辑ALCO的基础上构建了相应的动态描述逻辑.它具有清晰的语义特征,既提供了可判定的推理服务,又能有效地对动态过程和运行机制进行表示和推理.

本文将网构软件系统抽象成一个离散事件系统,提出一种新的故障诊断方法,称为基于动态描述逻辑的网构软件系统故障诊断(diagnosis of Internetware systems using dynamic description logic).它具有如下的特点:

(1) 基于动态描述逻辑,提出了一种系统的方法来分析待诊断的离散事件系统.可以看出,动态描述逻辑的语法和语义适宜对离散事件系统进行研究.

(2) 在动态描述逻辑框架下,研究了离散事件系统的可诊断性问题.可以看出,离散事件系统的可诊断性问题可以作为动态描述逻辑中公式集的可满足性问题来研究.

(3) 从动态描述逻辑的角度给出了构建诊断器的算法,并以一个例子给出了算法的执行过程.

本文第 1 节回顾相关的工作.第 2 节对动态描述逻辑进行介绍.第 3 节引入公式集、公式集变换函数,将动态描述逻辑引入到系统故障诊断建模中.第 4 节从动态描述逻辑的角度研究系统故障诊断问题,并给出网构软件中的一个应用场景.由此可以看出,离散事件系统的可诊断性问题可以转化为动态描述逻辑中公式集的可满足性问题.第 5 节提出故障诊断器的构造算法,并以一个例子给出其构造过程.第 6 节对全文进行总结.

1 相关工作

以COM/DCOM,CORBA3.0,J2EE,.NET等为代表的软件构件技术和中间件技术在 20 世纪末和 21 世纪初得到了长足的发展,并获得广泛的应用.随着Internet的进一步普及,未来的软件系统需要适应Internet发展的要求,具备开放的结构,拥有动态协同、在线演化、环境感知和自主适应的能力.对于这种新的软件形态,北京大学的杨芙清院士、南京大学的吕建教授、中国科学院数学研究所金芝研究员都作过表述,将其称为网构软件(Internetware)[1−3].网构软件系统中的实体元素被组织为分布、自治、异构的构件,具有独立性、主动性和自适应性,这给软件系统故障诊断问题的解决带来了新的机遇.基于网构软件的系统有明确的“软件边界”.这使得能够在一定粒度上观测软件系统的行为,同时也为软件系统的故障诊断问题打开了新的思路.

现有的故障诊断方法主要有基于规则推理、基于离散事件系统框架、基于模型、基于人工神经网络和基于案例的方法.其中,基于离散事件系统框架的诊断近年来已发展为故障诊断中的一个十分活跃的研究分支.离散事件系统中的故障诊断概念最初由文献[8]提出.其思想是将被诊断系统建模为有限状态自动机,将故障建模为不可观察到的状态转换.基于可观察到的状态转换,故障诊断器给出当前系统状态的估计,于是可以指出是否有一个故障发生.也就是说,如果DES执行了一个错误事件,那么在一个有限数量的状态转换或事件后,最终可以诊断到该错误事件的发生.文献[8]首先提出可诊断性的概念,并研究了其属性,本质上来说,一个可诊断的系统确保在一个有限的延迟内能够检测到所有已发生的故障,并确定其故障类型.文献[9]进一步扩展了基于状态的故障诊断方法,加上有关时间的信息,从而提高了故障诊断的精度.文献[10]研究了离散事件系统在分布式情况下的诊断问题,并提出了相关的分布式体系结构.文献[11]研究了在离散事件系统框架下如何设计控制系统和诊断系统.文献[12,13]研究了基于Petri网模型的故障诊断问题.文献[14,15]提出了多项式时间复杂度的算法用于测试系统的可诊断性.文献[16]研究了部分可观测的离散事件系统中的一些问题的计算复杂度.为了能够有效地描述复杂的系统行为,在离散事件系统的分析和控制中引入了时序逻辑[17].

随着软件应用领域的不断拓展,更多的复杂技术系统借助软件来控制,软件系统的可信问题(trustworthy)正越来越多地受到人们的重视,同时人们也更加注重软件系统的故障诊断问题.网构软件系统现有的故障诊断方法还缺乏相应的理论指导.我们发现,网构软件系统的失效常常是因为其组成的软件实体的行为并没有全面地遵循设计,而执行了设计者和软件编写者所禁止的行为,也就是违反了系统的语义.也就是说,软件故障同应用系统的语义相关,而现有的故障诊断方法对这种存在于软件系统中的语义考虑得较少.

基于以上分析,我们提出一种基于动态描述逻辑(dynamic description logic,简称 DDL)的故障诊断方法.动态描述逻辑在描述逻辑的基础上引入了动态维,用于描述和推理动态领域的知识.其主要特点是具有较强的描述能力,同时保证了相关推理问题的可判定性,具有有效的推理算法作为支撑,能够有效地刻画系统语义.该故障诊断方法以动态描述逻辑的动作建模软件系统的事件,以若干个体变元表示系统状态,能够有效地建模复杂系统的行为,并以此作为网构软件系统运行时故障诊断的参考依据.

2 动态描述逻辑 D-ALCO 简介

动态描述逻辑D-ALCO的基本符号包括[19]:① 由概念名组成的集合N_C;② 由角色名组成的集合N_R;③ 由个体名组成的集合N_I;④ 由个体变元组成的集合N_{IV};⑤ 由原子动作名组成的集合N_A;⑥ 概念构造符{ },¬,⊔,∀;⑦ 公式构造符¬,∨,〈 〉;⑧ 动作构造符“∪”,“;”,“*”,“?”;⑨ 其他符号,包括定义号“≡”、圆括号“( )”、逗号“,”.从这些符号出发可以构造出角色、概念、公式、以及动作.

定义 1. R是D-ALCO中的角色当且仅当$R\in N_R$.

定义 2. D-ALCO 中的概念由如下产生式生成:

$$C,D::=C_i|\{u\}|\neg C|C\sqcup D|\forall R.C,$$

其中,$C_i\in N_C$,$u\in N_I\cup N_{IV}$,R为角色.

将形如$\{u\}$,$\neg C$,$C\sqcup D$,$\forall R.C$ 的概念分别称为枚举、否定、析取、值限定概念.

令C_i为概念名,D为概念,则称$C_i\equiv D$为概念定义式.

对于由概念定义式组成的有限集合 $\mathcal{T}$,如果每个概念名最多在 $\mathcal{T}$中某个概念定义式的左边出现一次,则称 $\mathcal{T}$ 为 TBox.

相对于某个 TBox $\mathcal{T}$,如果概念名 C 在 $\mathcal{T}$中某个概念定义式的左边出现,则称 C 为被定义的概念名,否则称 C 为简单概念名.

定义 3. D-ALCO 中的公式由如下产生式生成:

$$\varphi,\psi::=C(u)|R(u,v)|\neg\varphi|\varphi\vee\psi|\langle\pi\rangle\varphi,$$

其中,$u,v\in N_I\cup N_{IV}$,C为概念,R为角色,π为动作.

将形如 $C(u)$,$R(u,v)$,$\neg\varphi$,$\varphi\vee\psi$,$\langle\pi\rangle\varphi$的公式分别称为概念断言、角色断言、否定式、析取式、动作存在性

断言.

给定某个 TBox $\mathcal{T}$,将形如 $R(u,v)$,$\neg R(u,v)$,$C(u)$和$\neg C(u)$(其中的 C 是相对于 $\mathcal{T}$的简单概念名)的公式都称为简单公式.

对于任一简单公式φ,如果φ形如$\neg R(u,v)$或$\neg C(u)$,则用$\varphi^{\neg}$表示相应的简单公式$R(u,v)$或$C(u)$;否则,如果φ形如$R(u,v)$或$C(u)$,C为简单概念名,则用$\varphi^{\neg}$相应地表示$\neg R(u,v)$或$\neg C(u)$.

由简单公式组成的有限集合$\varepsilon(v_1,...,v_n)=\{\psi_1(v_1),...,\psi_n(v_n)\}$称作简单公式集,其中$(v_1,...,v_n)$表示简单公式集$\varepsilon$中的所有个体变元组成的有限序列,在不产生歧义的情况下,可以将$\varepsilon(v_1,...,v_n)$简记为ε.

对于任一简单公式集$\varepsilon(v_1,...,v_n)=\{\psi_1(v_1),...,\psi_n(v_n)\}$,则用$\varepsilon^{\neg}$表示公式集$\varepsilon^{\neg}(v_1,...,v_n)=\{\psi_1^{\neg}(v_1),...,\psi_n^{\neg}(v_n)\}$.

定义在公式集上的操作符$\cup$和$\uplus$.简单公式集$\varepsilon(v_1,...,v_n)=\{\psi_1(v_1),...,\psi_n(v_n)\}$和$\varepsilon_0(v_1,...,v_n)=\{\varphi_1(v_1),...,\varphi_n(v_n)\}$,$\cup$表示普通的集合并操作$\varepsilon(v_1,...,v_n)\cup\varepsilon_0(v_1,...,v_n)=\{\psi_1(v_1),...,\psi_n(v_n),\varphi_1(v_1),...,\varphi_n(v_n)\}$,$\uplus$表示对公式集中的对应公式取并操作$\varepsilon(v_1,...,v_n)\uplus\varepsilon_0(v_1,...,v_n)=\{\psi_1\sqcup\varphi_1(v_1),...,\psi_n\sqcup\varphi_1(v_n)\}$.

定义 4. 给定某个TBox $\mathcal{T}$,将$\alpha(v_1,...,v_n)\equiv(P_\alpha,E_\alpha)\equiv(P_\alpha(v_1,...,v_n),E_\alpha(v_1,...,v_n))$称为一个原子动作定义式,其中:

(1) $\alpha\in N_A$,为所定义的原子动作名.

(2) $(v_1,...,v_n)$是由出现在P和E中的所有个体变元组成的有限序列.

(3) P_α是由公式组成的有限集合,表示动作执行前必须满足的前提条件.

(4) E_α是由简单公式组成的有限集合,表示执行该动作后将会发生的影响.

(5) P_α与E_α之间满足如下约束:对于任一$\varphi\in E_\alpha$,都有$\varphi^{\neg}\in P_\alpha$.

对于由动作定义式组成的任一有限集合 $\mathcal{A}c$,如果每个原子动作名最多在 $\mathcal{A}c$ 中某个动作定义式的左边出现一次,则称 $\mathcal{A}c$ 为 AActBox.

动作的执行可以改变简单公式集ε,动作α对简单公式集的变换定义为函数$\rho_\alpha(\varepsilon)::=(\varepsilon\backslash E_\alpha^{\neg})\cup E_\alpha$,其中的"\"为集合差运算.在构建公式的语义模型时通过简单公式集ε区分出不同的可能世界(或者说状态),并且通过简单公式集ε体现出这些可能世界之间存在的由原子动作引起的可达关系.

下面介绍 D-ALCO 的语义定义.

对于描述逻辑(例如ALCO)来说,其语义解释一般形如$I=(\Delta,\bullet^I)$.其中的Δ是由个体组成的解释域;解释函数$\bullet^I$将每个概念名C_i解释为Δ的某个子集C_i^I,将每个角色名R_i解释为Δ上的某个二元关系R_i^I,将每个个体名p_i解释为Δ中的某个元素p_i^I.

动态描述逻辑在描述逻辑的基础上引入了动态维,使得语义模型从整体上体现为由多个可能世界构成的可能世界空间.在每个可能世界下都分别对概念名、角色名以及个体名进行解释.动作被解释为关于这些可能世界的二元关系.

定义 5. D-ALCO模型是一个三元组$M=(\Delta,W,I)$.其中,Δ是由个体组成的非空集合,作为该模型的论域;W是由可能世界组成的集合;I对W中的每个可能世界w赋予一个解释$I(w)=(\Delta,\bullet^{I(w)})$.

定义 6. 对应于任一D-ALCO模型$M=(\Delta,W,I)$,将函数$\gamma:N_{IV}\to\Delta^I$称为基于M的一个指派.

任一指派γ给出了N_{IV}中各个个体变元的一种赋值,分别将这些个体变元指派为Δ^I中的个体.如果$v\in N_{IV},\gamma_1,\ldots,\gamma_n$是基于$M$的所有指派,则$\mathcal{D}(v)=\cup_{1\le i\le n}\{\gamma_i(v)\}$,用$\mathcal{D}(v)$表示$v$的解释域.

一个解释I和一个指派γ可以确定一个从公式ψ到布尔值{True,False}的映射.例如,公式$R(u,a)$,其中,$u\in N_{IV}$,$a\in N_I$,R为角色,有且只有在$(\gamma(u),a^I)\in R^I$的情况下,可以得到$R(u,a)^{I,\gamma}$=True;考虑公式$C(v)$,其中,$v\in N_{IV}$,C为概念,有且只有在$\gamma(v)\in C^I$的情况下,可以得到$C(v)^{I,\gamma}$=True.

对应于任一D-ALCO模型$M=(\Delta,W,I)$和基于M的一个指派函数$\gamma:N_{IV}\to\Delta^I$,可以确定如下一个从简单公式集$\varepsilon(v_1,\ldots,v_n)=\{\psi_1(v_1),\ldots,\psi_n(v_n)\}$到布尔值{True,False}的映射$\Gamma_{M,\gamma}:\Gamma_{M,\gamma}(\varepsilon(v_1,\ldots,v_n))=\varepsilon(v_1,\ldots,v_n)^{I,\gamma}=\{\psi_1(v_1),\ldots,\psi_n(v_n)\}^{I,\gamma}=\psi_1(v_1)^{I,\gamma}\wedge\ldots\wedge\psi_n(v_n)^{I,\gamma}$.

3 基于 D-ALCO 的诊断模型

网构软件系统中的实体元素被组织为分布、自治、异构的构件,具有独立性、主动性和自适应性,基于网构软件的系统有明确的“软件边界”,这使得能够在一定粒度上观测软件系统的行为.从故障诊断的角度来看,网构软件系统可以被抽象为一个离散事件系统.

一个待诊断的离散事件系统G,是一个四元组$G=(X,\Sigma,\delta,X_0)$[8,11],其中:

- X表示状态的有限集合.
- Σ表示事件的有限集合.
- $\delta\subseteq X\times\Sigma\times X$表示状态转换的有限集合.
- $X_0\subseteq X$是初始状态的集合.

令Σ^*表示所有有限长度事件序列的集合,包括长度为 0 的序列,记作ϕ.集合Σ^*中的一个元素称为事件轨迹,Σ^*的子集记作$L(G)$,表示G中从初始状态开始的事件轨迹的集合.对于事件轨迹s和事件σ,用$\sigma\in s$表示σ是包含在事件轨迹s中的一个事件.

使用若干个体变元组成的有限序列$(v_1,\ldots,v_n)$表示状态,其中$v_i\in N_{IV}$,$1\le i\le n$.离散事件系统G的状态空间X等于所有个体变元解释域的笛卡尔乘积,即$X:=\prod_{i=1}^{n}\mathcal{D}(v_i)$,这里,$\mathcal{D}(v_i)$是$v_i$的解释域.

用简单公式集$\varepsilon(v_1,\ldots,v_n)=\{\psi_1(v_1),\ldots,\psi_n(v_n)\}$来描述状态空间不同的子集.令$\Theta(v_1,\ldots,v_n)$表示由若干简单公式集$\varepsilon$组成的集合,如果$\varepsilon\in\Theta(v_1,\ldots,v_n)$,那么在该公式集$\varepsilon$上的一个布尔值映射函数$\Gamma_{M,\gamma}(\varepsilon(v_1,\ldots,v_n)):X\to\{\text{True,False}\}$,对状态空间$X$中的每个状态,指定一个布尔值.对于每一个简单公式集$\varepsilon_0\in\Theta(v_1,\ldots,v_n)$,关联一个集合$\oplus\varepsilon_0\subseteq X$,在该集合上$\Gamma_{M,\gamma}(\varepsilon_0)$取值为True.因此公式集的集合$\Theta(v_1,\ldots,v_n)$和幂集$2^X$存在一一对应关系,也就是说公式集和状态集能够互相替换使用.例如,如果$X_{sub}\subseteq\oplus\varepsilon_0$,那么$\Gamma_{M,\gamma}(\varepsilon_0)$在$X_{sub}\subseteq\oplus\varepsilon_0$上取值为True.

使用动态描述逻辑D-ALCO中的一个原子动作α表示一个事件σ,由原子动作组成的有限集合N_A表示事件的有限集合Σ.

状态转换将一个状态映射到另一个状态.这样的转换可以自然地扩展到状态集和公式集,分别称为状态集转换和公式集变换.

考虑如上所述的离散事件系统G.G的初始状态集X_0通过初始公式集指定,即$\varepsilon_0,\oplus\varepsilon_0=X_0$.$G$的状态转换借助原子动作加以描述,原子动作的形式为

$$\alpha(v_1,\ldots,v_n)\equiv(P_\alpha,E_\alpha)\equiv(P_\alpha(v_1,\ldots,v_n),E_\alpha(v_1,\ldots,v_n)).$$

其中,动作$\alpha\in N_A$表示一个事件,P_α是由公式组成的有限集合,表示动作执行前必须满足的前提条件,E_α是由简单公式组成的有限集合,表示执行该动作后将会发生的影响,动作α对公式集的变换可以用公式集变换函数描述为$\rho_\alpha(\varepsilon)::=(\varepsilon\backslash E_\alpha^{-})\cup E_\alpha$,动作$\alpha$对个体变元$(v_1,\ldots,v_n)$的影响,可以描述为$\Gamma_{M,\gamma}(\rho_\alpha(\varepsilon(v_1,\ldots,v_n)))::=\Gamma_{M,\gamma}((\varepsilon(v_1,\ldots,v_n)\backslash E_\alpha^{-})\cup E_\alpha)$,也就是说,$\Gamma_{M,\gamma}(\rho_\alpha):X\to X$是一个定义在状态空间上的映射.如果动作没有前提条件,就认为前提条件满足.如果前提条件P_α满足,动作就可以执行.在执行时,按照函数ρ_α改变公式集ε,并给若干个体变元$(v_1,\ldots,v_n)$赋予新的解释.于是,在动作α上的状态转换发生.如果多个动作的前提条件同时满足,则随机选择它们中的一个执行.

另一个公式集变换函数用于定义向前一步的可达性.G的公式集变换函数Λ_R定义了当一个状态转换发生后所有的可达状态.

对于一个动作$\alpha\in N_A$和一个公式集$\varepsilon(v_1,\ldots,v_n)\in\Theta(v_1,\ldots,v_n)$,正式定义如下:$\Lambda_R(\varepsilon(v_1,\ldots,v_n),\alpha)::=\varepsilon'$,$\varepsilon'$满足$[P_\alpha\subseteq\rho_\alpha^{-1}(\varepsilon')]\wedge[\varepsilon\subseteq\rho_\alpha^{-1}(\varepsilon')]$.

对于$\Omega\subseteq\Sigma$,定义$\Lambda_R(\varepsilon(v_1,\ldots,v_n),\Omega):=\uplus_{\alpha\in\Omega}\Lambda_R(\varepsilon(v_1,\ldots,v_n),\alpha)$.最后,定义$\Lambda_R^*(\varepsilon,\Omega)$表示从一个在$\oplus\varepsilon$中的状态,执行$\Omega$中的 0 个或多个事件转换后的所有可达状态集.

给定Λ表示公式集变换函数和$\varepsilon\in\Theta(v_1,\ldots,v_n)$,$\varepsilon$对$\Lambda$的限制记为$\Lambda|_\varepsilon$,是一个公式集变换函数,定义如下:$\Lambda|_\varepsilon(\varepsilon')::=\Lambda(\varepsilon\cup\varepsilon')\cup\varepsilon,\forall\varepsilon'\in\Theta(v_1,\ldots,v_n)$.

4 系统的可诊断性分析

可诊断性是指从一个有限数量的观测事件,推理出过去发生的、没有观测到的错误事件的能力.令$\Sigma_F\subseteq\Sigma$表示错误事件集,$\Sigma_o\subseteq\Sigma$表示可观测到的事件集合,$\Sigma_{uo}=\Sigma\backslash\Sigma_o$表示不可观测事件集合.事件可以部分地被观测描述为一个事件观测掩码函数$M:\Sigma\to\Sigma_o\cup\{\phi\}$,不失一般性,对于任意$\sigma\in\Sigma_F$,$M(\sigma)=\phi$.$M$的定义可以从$\Sigma$扩展到$\Sigma^*$:对于$s\in\Sigma^*$,$\sigma\in\Sigma$,$M(s\sigma)=M(s)M(\sigma)$.

文献[8]给出了离散事件系统的可诊断性,如下(不失一般性,仅考虑单个错误类型):

定义 7. 一个系统G被称为相对于观察掩码M和错误事件集Σ_F是可诊断的,如果满足:

$$(\exists n\in\mathrm{N})(\forall s\in L(G),s_f\in\Sigma_F)(\forall t\in L(G)/s)(\|t\|\geq n\Rightarrow D).$$

其中,可诊断性条件 D 为

$$(\forall w\in M_L^{-1}(M(st)))(\exists u\in prefix(\{w\}),u_f\in\Sigma_F).$$

其中,N表示自然数集合,$s_f,u_f\in\Sigma$分别表示在事件轨迹$s,u\in\Sigma^*$中的最后一个事件,$L(G)/s=\{t\in\Sigma^*|st\in L(G)\}$,$M_L^{-1}(y)=\{s\in L(G)|M(s)=y\}$,$prefix(\{w\})\subseteq\Sigma^*$是$w\in\Sigma^*$的所有前缀的集合.可诊断性的定义直观上如图 1 所示[20].

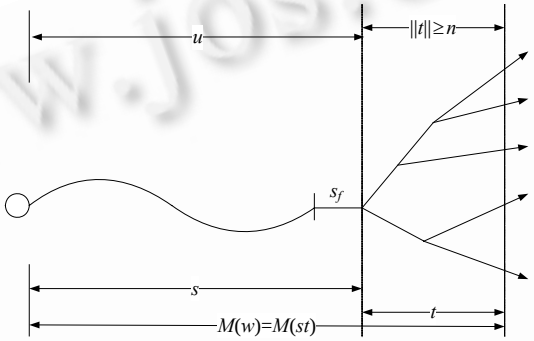

Fig.1 Definition of diagnosability

图 1 可诊断性的定义

可诊断性就是要求每个错误事件能够产生足够明确的观测,从而能够在一个有限的延迟内标识这个错误.

使用动态描述逻辑,完成可诊断性测试如下:

算法 1.

考虑$G=(X,\Sigma,\delta,X_0)$,用若干个体变元$(v_1,...,v_n)$表示状态,用动态描述逻辑的原子动作集N_A表示事件集Σ(为了表示方便,在不产生歧义的情况下,不区分N_A和Σ),给出如下模型:

初始条件:$\varepsilon_0\in\Theta(v_1,...,v_n)$.

事件发生规则:

$$\forall\alpha\in N_A:\alpha(v_1,...,v_n)\equiv(P_\alpha(v_1,...,v_n),E_\alpha(v_1,...,v_n)).$$

(1)以布尔变量f扩充表示状态的个体变元$(v_1,...,v_n)$,以标识过去是否有错误发生.扩充后的状态变量形如$(v_1,...,v_nf)$.扩充后的系统模型$G_0=(X\times\{\mathrm{True,False}\},\Sigma,\delta_0,X_{0l})$如下:

初始条件:$ini(v_1,...,v_nf)=\varepsilon_0\cup\{\neg err(f)\}$.

事件发生规则:

$$\forall\alpha\in\Sigma_F:(P_\alpha(v_1,...,v_nf),E_\alpha(v_1,...,v_nf))\equiv(P_\alpha(v_1,...,v_n),E_\alpha(v_1,...,v_n)\cup\{err(f)\}).$$

$$\forall\alpha\notin\Sigma_F:(P_\alpha(v_1,...,v_n,f),E_\alpha(v_1,...,v_n,f))\equiv(P_\alpha(v_1,...,v_n),E_\alpha(v_1,...,v_n)).$$

使用一个公式集η表示在一个错误发生后所访问的状态的集合,$\eta(v_1,...,v_nf)=\{err(f)\}$.

(2) 计算$G_d=(G_0\|G_0)=((X\times\{\mathrm{True,False}\})^2,\Sigma,\delta_d,X_{dl})$,表示$G$与自身的组合,用$(v_1,...,v_nf)$和$(u_1,...,u_nf')$分别表示前一个$G_0$和后一个$G_0$的状态:

初始条件:$ini(v_1,...,v_n,f)\cup ini(u_1,...,u_n,f')$.

事件发生规则:

$\forall(\alpha,\alpha')\in\{(\Sigma\cup\{\phi\})^2\setminus\{\phi,\phi\}\}$,并且$M(\alpha)=M(\alpha')$:

如果$\alpha\neq\phi$并且$\alpha'\neq\phi$,那么$(P_{M(\alpha)}(v_1,...,v_n,f)\cup P_{M(\alpha')}(u_1,...,u_n,f'),E_{M(\alpha)}(v_1,...,v_n,f)\cup E_{M(\alpha')}(u_1,...,u_n,f'))$;

如果$\alpha\neq\phi$并且$\alpha'=\phi$,那么$(P_{M(\alpha)}(v_1,...,v_n,f),E_{M(\alpha)}(v_1,...,v_n,f))$;

如果$\alpha=\phi$并且$\alpha'\neq\phi$,那么$(P_{M(\alpha')}(u_1,...,u_n,f'),E_{M(\alpha')}(u_1,...,u_n,f'))$.

(3) 使用动态描述逻辑 D-ALCO 的 Tableau 判定算法判定下面的公式集的可满足性:

如果对于任意的状态$((v_1^0,...,v_n^0,f),(u_1^0,...,u_n^0,f'))$,$\exists((v_1,...,v_n,f),(u_1,...,u_n,f'))$,有公式集:

$$eq((v_1^0,...,v_n^0,f),(v_1,...,v_n,f))\cup eq((u_1^0,...,u_n^0,f'),(u_1,...,u_n,f'))\cup\eta(v_1,...,v_n,f)\cup\eta^{\neg}(u_1,...,u_n,f'),$$

那么G是可诊断的当且仅当该公式集在G_d中不可满足,其中,公式集$eq((x_1^0,...,x_n^0,f),(x_1,...,x_n,f))::=\{equal(x_1^0,x_1),...,equal(x_n^0,x_n)\}$,$equal$为角色.

算法 1 结束.

上式检测具有如下属性的状态对$((v_1,...,v_n,f),(u_1,...,u_n,f'))\in(X\times\{\text{True,False}\})^2$的存在:

- $(v_1,...,v_n,f)$是一个错误状态:$\Gamma_{M,\gamma}(\eta(v_1,...,v_n,f))$成立.
- $(u_1,...,u_n,f')$是一个非错误状态:$\neg\Gamma_{M,\gamma}(\eta(u_1,...,u_n,f'))$成立.
- 从初始条件$ini(v_1^0,...,v_n^0,f)\cup ini(u_1^0,...,u_n^0,f')$沿着一个事件轨迹到达状态$((v_1,...,v_n,f),(u_1,...,u_n,f'))$,形成一个循环,即,$\Gamma_{M,\gamma}(eq((v_1^0,...,v_n^0,f),(v_1,...,v_n,f))\cup eq((u_1^0,...,u_n^0,f'),(u_1,...,u_n,f')))$为真.

上面的公式无论何时满足,在G中都存在一对任意长度的有错误和无错误的路径,这一对路径是不可区分的,于是系统是不可诊断的.我们已经开发了动态描述逻辑推理机,可以用来检测G_d中上面的公式集的可满 足性.

例 1:为了示例上述算法的结论,考虑一个简单的例子,在Internet开放环境下,多个软件实体相互协作,完成给定任务的一个网构软件系统.如图 2 所示,该网构软件系统由 6 个具有自治能力的软件实体所构成,软件实体之间存在数据的交互,如图中有向箭头所示,表示数据从一个软件实体传输到另一个软件实体,但系统故障诊断者只能检测到其中部分的数据交互,如图中实线所示.待处理的数据最初在软件实体A_0上,数据可以在软件实体之间沿箭头方向任意传输,但当数据传输到软件实体A_1,A_4时,一个错误发生.故障诊断的目标就是通过观察网构软件系统中可观察到的部分数据的交互,以获知是否有错误发生.

上面的问题可以被形式化为基于D-ALCO的诊断模型中的一个故障诊断问题.待诊断的系统G,有单一个体变元v表示网构软件系统中待处理数据的位置,v能够取值$\{A_0,A_1,A_2,A_3,A_4,A_5\}$;初始状态是$\varepsilon_0(v)=\{A_0(v)\}$;事件集是$\Sigma=\{o_0,o_1,o_2,o_3,uo_0,uo_1,uo_2,uo_3,uo_4,uo_5\}$;事件观测掩码$M$定义为$M(uo_i)=\phi$,其中,$0\le i\le 5$,并且$M(o_i)=o_i$,其中,$0\le i\le 3$.待诊断问题的模型如下:

(1) 初始条件:$\varepsilon_0(v)=\{A_0(v)\}$.

(2) 事件发生规则:

$$o_0(v)\equiv(\{A_0(v)\},\{\neg A_0(v),A_3(v)\}),$$
$$o_1(v)\equiv(\{A_3(v)\},\{\neg A_3(v),A_0(v)\}),$$
$$o_2(v)\equiv(\{A_5(v)\},\{\neg A_5(v),A_3(v)\}),$$
$$o_3(v)\equiv(\{A_2(v)\},\{\neg A_2(v),A_3(v)\}),$$
$$uo_0(v)\equiv(\{A_0(v)\},\{\neg A_0(v),A_1(v)\}),$$
$$uo_1(v)\equiv(\{A_0(v)\},\{\neg A_0(v),A_4(v)\}),$$
$$uo_2(v)\equiv(\{A_4(v)\},\{\neg A_4(v),A_5(v)\}),$$
$$uo_3(v)\equiv(\{A_1(v)\},\{\neg A_1(v),A_2(v)\}),$$
$$uo_4(v)\equiv(\{A_3(v)\},\{\neg A_3(v),A_2(v)\}),$$
$$uo_5(v)\equiv(\{A_3(v)\},\{\neg A_3(v),A_5(v)\}).$$

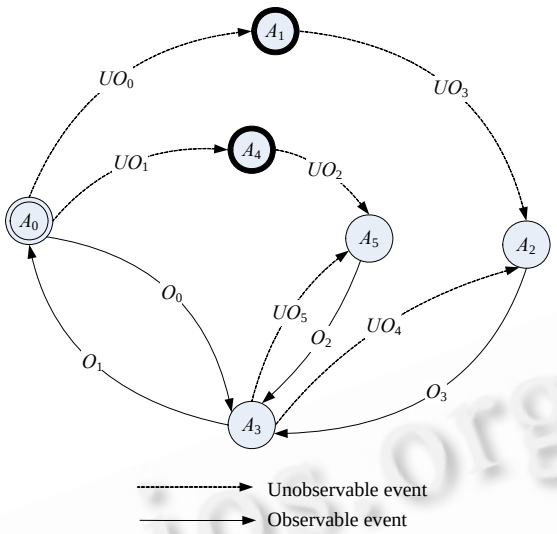

Fig.2 Diagnosed Internetware system

图 2 待诊断的网构软件系统

当数据传输到软件实体A_1,A_4时,一个错误就会发生,于是$\Sigma_F=\{uo_0,uo_1\}$.注意,此时uo_0,uo_1是不可观察到的事件.

为了验证可诊断性,依据算法,以布尔变量f扩充个体变元v得到(v,f).G_0的两个拷贝,分别使用(v,f)和(u,f')表示状态,通过组合G_0的两个拷贝,可以得到G_d.

使用动态描述逻辑推理机验证可诊断性条件,得到该网构软件系统是可诊断的.动态描述逻辑D-ALCO将描述逻辑ALCO、动态逻辑以及基于可能模型途径的动作理论有机地结合了起来;既可以基于描述逻辑刻画和推理静态的领域知识,又可以在这些知识的基础上将动作也作为一类知识进行刻画和推理.D-ALCO的Tableau判定算法具有可终止性、可靠性和完备性.应用该算法,可以在采用开世界假设的情况下对D-ALCO中任一公式的可满足性进行判定[19].

在Internet开放环境下的网构软件系统中,可以通过部署在路由器上的监控程序获知软件实体之间存在的数据交互.使用上述算法可知,事件o_2无须被观察,系统依然可以保持可诊断性.于是,可以据此减少部署相关监控程序的路由器数量,从而降低部署成本和维护代价,获得现实意义.

5 诊断器的构建

本节描述诊断器的构造过程,并讨论如何使用该诊断器实时地诊断系统的故障.如果系统被判断为是可诊断的,则可以进行诊断器的构建.像可诊断性测试一样,我们开发了相关算法,用于从系统模型 G 构建诊断器.

系统模型G的诊断器为$G_{\text{diagnoser}}=(X_{\text{diagnoser}},\Sigma,\delta_{\text{diagnoser}},D_0)$,其中诊断器的初始状态$D_0$定义为

$$S_0(v_1,...,v_n,f)=ini(v_1,...,v_n,f),S'_0(v_1,...,v_n,f)=ini(v_1,...,v_n,f)\cup\eta^{\neg}(v_1,...,v_n,f).$$

状态空间$X_{\text{diagnoser}}$是由从初始状态D_0通过状态转换函数$\delta_{\text{diagnoser}}$产生出的一切可达的状态所组成.

诊断器的一个状态含有两个公式集:一个记为$S_k(v_1,...,v_n,f)\in\Theta(v_1,...,v_n,f)$,表示在系统模型$G$中出现了第$k$个可观测的事件后的可能状态;另一个记为$S'_k(v_1,...,v_n,f)\supseteq S_k(v_1,...,v_n,f)$,且$S'_k(v_1,...,v_n,f)\in\Theta(v_1,...,v_n,f)$表示$\oplus S'_k$是$\oplus S_k$的子集,该子集不包括$\oplus\eta(v_1,...,v_n,f)$中的任何状态,表示在系统模型$G$中沿着那些无错误的事件轨迹执行后的状态,在$\oplus S'_k$中的状态使$\neg\Gamma_{M,\gamma}(\eta(v_1,...,v_n,f))$恒为真.

诊断器的状态转换函数$\delta_{\text{diagnoser}}$描述如下:在出现第$(k+1)$个可观测的事件$\alpha(k\geq0)$时,更新当前值对$(S_k(v_1,...,v_n,f),S'_k(v_1,...,v_n,f))$以获得新的值对$(S_{k+1}(v_1,...,v_n,f),S'_{k+1}(v_1,...,v_n,f))$.由于在系统模型$G$中,第$(k+1)$个观

察到的结果相当于执行在$M^{-1}(\phi)\cap\Sigma$中的若干不可观察到的事件,之后执行在$M^{-1}(\alpha)$中的一个可观察到的事件.于是,通过从$S_k(v_1,...,v_n f)$使用可达性计算,经过$M^{-1}(\phi)\cap\Sigma$中的若干不可观察的事件以及$M^{-1}(\alpha)$中的单个事件,得到$S_{k+1}(v_1,...,v_n f)$.同理,以一个相似的方式计算$S'_{k+1}(v_1,...,v_n f)$,不同之处是将向前可达性公式集变换函数Λ_R替换为其上的限制$\Lambda_R|_{\eta^{\neg}(v_1,...,v_n f)}$.

需要注意的是,在以上诊断器状态转换的过程中,无论何时,$\oplus S'_k(v_1,...,v_n f)$都是$\oplus S_k(v_1,...,v_n f)$的一个子集.$\Gamma_{M,\gamma}(S'_k(v_1,...,v_n f))\neq$false,表示系统执行了一些轨迹,该轨迹上可能存在一个错误状态,但同时存在其他不可区分的不存在错误状态的轨迹.若同时$\Gamma_{M,\gamma}(S_k(v_1,...,v_n f))\neq$false,那么在这种情况下并不能判断是否存在错误的发生.但如果

$$[\Gamma_{M,\gamma}(S_k(v_1,...,v_n f))\neq\text{false}]\wedge[\Gamma_{M,\gamma}(S'_k(v_1,...,v_n f))=\text{false}],$$

则表示沿着系统所能执行的所有路径,都已经访问过一个错误状态.而状态对

$$[\Gamma_{M,\gamma}(S_k(v_1,...,v_n f))=\text{false}]\wedge[\Gamma_{M,\gamma}(S'_k(v_1,...,v_n f))=\text{false}]$$

表示不可能的情况.在以上两种情况下,诊断器都会报告一个错误.

总之,诊断器$G_{\text{diagnoser}}$构建算法如下:

算法 2.

初始步骤:

(初始情况下,没有观测任何事件,即当 k=0 时)

$$S_0(v_1,...,v_n f)=ini(v_1,...,v_n f),$$

$$S'_0(v_1,...,v_n f)=ini(v_1,...,v_n f)\cup\eta^{\neg}(v_1,...,v_n f).$$

迭代步骤:

在第(k+1)次观察到$\alpha\in M(\Sigma)\backslash\{\phi\}$时:

$$S_{k+1}(v_1,...,v_n f)=\Lambda_R((\Lambda_R^{*}(S_k(v_1,...,v_n f),M^{-1}(\phi)\cap\Sigma)),M^{-1}(\alpha)),$$

$$S'_{k+1}(v_1,...,v_n f)=\Lambda_R|_{\eta^{\neg}(v_1,...,v_n f)}((\Lambda_R|_{\eta^{\neg}(v_1,...,v_n f)})^{*}(S'_k(v_1,...,v_n f),M^{-1}(\phi)\cap\Sigma)),M^{-1}(\alpha)).$$

申明一个错误,如果满足以下条件之一:

$$[\Gamma_{M,\gamma}(S_{k+1}(v_1,...,v_n f))\neq\text{false}]\wedge[\Gamma_{M,\gamma}(S'_{k+1}(v_1,...,v_n f))=\text{false}];$$

$$[\Gamma_{M,\gamma}(S_{k+1}(v_1,...,v_n f))=\text{false}]\wedge[\Gamma_{M,\gamma}(S'_{k+1}(v_1,...,v_n f))=\text{false}].$$

算法 2 结束.

考虑先前给出的网构软件系统中故障诊断的例子,示例上面的诊断器构建算法.

例 2:参考第 4 节的例 1,计算诊断器如下:

(1) k=0:由于$ini(v,f)=\{A_0(v),\neg err(f)\}$并且$\eta(v,f)=\{err(f)\}$,设置

$$S_0(v,f)=\{A_0(v),\neg err(f)\},S'_0(v,f)=\{A_0(v),\neg err(f)\}.$$

(2) k=1:存在 4 个可观察到的事件$o_0,o_1,o_2,o_3\in\Sigma$.

如果第 1 个观察到的是o_0,那么

$$S_1(v,f)=\{A_3(v),\neg err(f)\},S'_1(v,f)=\{A_3(v),\neg err(f)\}.$$

如果第 1 个观察到的是o_1,那么

$$\Gamma_{M,\gamma}(S_1(v,f))=\text{false},\Gamma_{M,\gamma}(S'_1(v,f))=\text{false}.$$

这就是说,o_1作为第 1 个可观测到的事件是不可能的.

如果第 1 个观察到的事件是o_2或者o_3,那么

$$S_1(v,f)=\{A_3(v),err(f)\},\Gamma_{M,\gamma}(S'_1(v,f))=\text{false}.$$

这表明在过去的某个时刻,一个错误已经发生.

(3) k=2:根据 k=1 时所引入的 3 个状态对,分为 3 种情况.

(3-1) 假设第 1 个观察(k=1)是o_0,即当前状态对

$$S_1(v,f)=\{A_3(v),\neg err(f)\},S'_1(v,f)=\{A_3(v),\neg err(f)\}.$$

如果下一个观察是o_0(即观测序列o_0o_0),那么

$$\Gamma_{M,\gamma}(S_2(v,f))=\text{false},\Gamma_{M,\gamma}(S'_2(v,f))=\text{false}.$$

这表明观测序列o_0o_0是不可能的.

如果下一个观察是o_1(即观测序列o_0o_1),那么

$$S_2(v,f)=\{A_0(v),\neg err(f)\}=S_0(v,f),S'_2(v,f)=\{A_0(v),\neg err(f)\}=S'_0(v,f).$$

如果下一个观察是o_2(即观测序列o_0o_2),那么

$$S_2(v,f)=\{A_3(v),\neg err(f)\},S'_2(v,f)=\{A_3(v),\neg err(f)\}.$$

如果下一个观察是o_3(即观测序列o_0o_3),那么

$$S_2(v,f)=\{A_3(v),\neg err(f)\},S'_2(v,f)=\{A_3(v),\neg err(f)\}.$$

(3-2) 假设第 1 个观察(k=1)是o_1.由于$\Gamma_{M,\gamma}(S_1(v,f))=\text{false},\Gamma_{M,\gamma}(S'_1(v,f))=\text{false}$,那么接下来,

$$\Gamma_{M,\gamma}(S_2(v,f))=\text{false},\Gamma_{M,\gamma}(S'_2(v,f))=\text{false}.$$

(3-3) 最后假设第 1 个观察(k=1)是o_2或者是o_3,即当前状态对

$$S_1(v,f)=\{A_3(v),err(f)\},\Gamma_{M,\gamma}(S'_1(v,f))=\text{false}.$$

如果下一个观察是o_0(即观测序列o_2o_0或者o_3o_0),那么

$$\Gamma_{M,\gamma}(S_2(v,f))=\text{false},\Gamma_{M,\gamma}(S'_2(v,f))=\text{false}.$$

这表明观测序列o_2o_0或者o_3o_0是不可能的.

如果下一个观察是o_1(即观测序列o_2o_1或者o_3o_1),那么

$$S_2(v,f)=\{A_0(v),err(f)\},\Gamma_{M,\gamma}(S'_2(v,f))=\text{false}.$$

如果下一个观察是o_2(即观测序列o_2o_2或者o_3o_2),那么

$$S_2(v,f)=\{A_3(v),err(f)\},\Gamma_{M,\gamma}(S'_2(v,f))=\text{false}.$$

如果下一个观察是o_3(即观测序列o_2o_3或者o_3o_3),那么

$$S_2(v,f)=\{A_3(v),err(f)\},\Gamma_{M,\gamma}(S'_2(v,f))=\text{false}.$$

在上面 3 种情况下,可以得知过去一个错误已经发生.

(4) k=3:仅需考虑下面的观测序列o_2o_1,即当前状态对:

$$S_2(v,f)=\{A_0(v),err(f)\},\Gamma_{M,\gamma}(S'_2(v,f))=\text{false}.$$

于是检验这个谓词对.如果下一个观察是o_0(即观测序列$o_2o_1o_0$),那么

$$S_3(v,f)=\{A_3(v),err(f)\},\Gamma_{M,\gamma}(S'_3(v,f))=\text{false}.$$

如果下一个观察是o_1(即观测序列$o_2o_1o_1$),那么

$$\Gamma_{M,\gamma}(S_3(v,f))=\text{false},\Gamma_{M,\gamma}(S'_3(v,f))=\text{false}.$$

这表明观测序列$o_2o_1o_1$是不可能的.

如果下一个观察是o_2(即观测序列$o_2o_1o_2$),那么

$$S_3(v,f)=\{A_3(v),err(f)\},\Gamma_{M,\gamma}(S'_3(v,f))=\text{false}.$$

如果下一个观察是o_3(即观测序列$o_2o_1o_3$),那么

$$S_3(v,f)=\{A_3(v),err(f)\},\Gamma_{M,\gamma}(S'_3(v,f))=\text{false}.$$

由于第 3 次迭代没有引入新的状态对,于是不需要进一步迭代.在这种情况下,经过上面的步骤,生成一个离线诊断器,表示为一个自动机,如图 3 所示.

诊断器通过实时地观察系统的行为进行故障的诊断.在图 3 中,当存在一个从状态对$[\Gamma_{M,\gamma}(S_k(v_1,...,v_n,f))\neq\text{false}]\wedge[\Gamma_{M,\gamma}(S'_k(v_1,...,v_n,f))\neq\text{false}]$到状态对$[\Gamma_{M,\gamma}(S_k(v_1,...,v_n,f))\neq\text{false}]\wedge[\Gamma_{M,\gamma}(S'_k(v_1,...,v_n,f))=\text{false}]$的转换时,诊断器报告一个错误.由于状态对$[\Gamma_{M,\gamma}(S_k(v_1,...,v_n,f))=\text{false}]\wedge[\Gamma_{M,\gamma}(S'_k(v_1,...,v_n,f))=\text{false}]$表示不可能的情况,当存在一个从状态对$[\Gamma_{M,\gamma}(S_k(v_1,...,v_n,f))\neq\text{false}]\wedge[\Gamma_{M,\gamma}(S'_k(v_1,...,v_n,f))\neq\text{false}]$到状态对$[\Gamma_{M,\gamma}(S_k(v_1,...,v_n,f))=\text{false}]\wedge[\Gamma_{M,\gamma}(S'_k(v_1,...,v_n,f))=\text{false}]$的转换时,诊断器也会报告一个错误发生.

至此,我们已经使用动态描述逻辑D-ALCO作为建模工具,用于网构软件系统的故障诊断.与以往的诊断方

法[8−18]相比, 该方法具有如下的优点:

(1) 可以与网构软件系统同步运行,实时监控,对被诊断系统透明.

(2) 具有坚实的数学基础和有效的推理算法作为支撑,在一定粒度上观测基于网构软件系统的行为,从而可以诊断大型随时间变化的动态系统.

(3) 无须人工干预,自动发现系统的运行故障,确定故障的类型、部位,从而为进一步的系统恢复提供支持.

(4) 有较强的通用性和独立性,使用简洁的模型,有效刻画网构软件系统中的实体元素间复杂的交互,方法不依赖于具体网构软件系统,从而将诊断系统的推理内核与被诊断系统模型分开,缩短了诊断系统的开发周期,提高了诊断效率,只要诊断模型抽象得正确,就能诊断所有可能的故障,并能给出令人信服的解释.

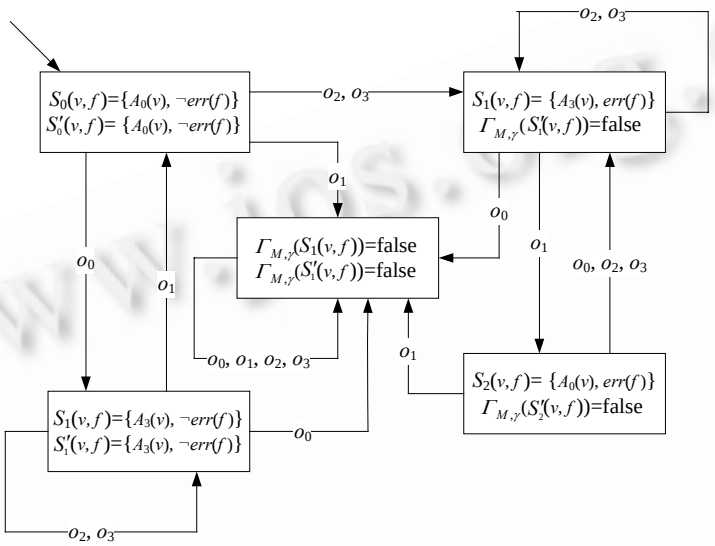

Fig.3 Diagnoser automaton of the Internetware system

图 3 网构软件系统的诊断器自动机

6 总结及未来的工作

本文从分析网构软件系统的特点入手,鉴于网构软件系统的行为能够在一定粒度上被观测,从故障诊断的角度,将网构软件系统抽象为一个离散事件系统.之后,通过引入动态描述逻辑对这个被诊断的离散事件系统进行分析.动态描述逻辑在描述逻辑的基础上引入了动态维,其主要特点在于具有较强的描述能力,同时保证了相关推理问题的可判定性,具有有效的推理算法作为支撑,适于描述和推理动态领域的知识,适于刻画系统的语义,即系统的正常行为和异常行为.在动态描述逻辑的框架下研究了离散事件系统的可诊断性问题,提出了相关的算法,将离散事件系统的可诊断性测试转化为动态描述逻辑中公式集的可满足性检测.借助公式集及其变换函数,给出了故障诊断器的构造算法,并配以例子说明其构造过程.该故障诊断器通过实时地观察系统的行为完成故障诊断.

近来可信软件系统在学术界和工业界已引起广泛的关注.从技术角度来讲,“可信”意味着可以充分相信其行为会严格地遵循设计,而执行设计者所禁止的行为的概率很低.软件系统故障诊断是构建可信软件系统的关键技术之一.下一步,我们将研究如何应用动态描述逻辑的测试、选择、顺序以及迭代动作刻画软件系统的复杂行为,并在动态描述逻辑的基础上引入对不确定信息的考虑,形成概率动态描述逻辑,以建模软件系统中不可重现的错误.

致谢 感谢审稿专家提出的宝贵意见.

References:

[1] Yang FQ, Mei H, Lü J, Jin Z. Some discussion on the development of software technology. Acta Electronica Sinica, 2002,30(12A): 1901−1906 (in Chinese with English abstract).

[2] Yang FQ. Thinking on the development of software engineering technology. Journal of Software, 2005,16(1):1−7 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/1.htm

[3] Lü J, Tao XP, Ma XX, Hu H, Xu F, Cao C. Research on agent-based Internetware. Science in China (Series E), 2005,35(12): 1233−1253 (in Chinese).

[4] Hasselbring W, Reussner R. Toward trustworthy software systems. Computer, 2006,39(4):91−92.

[5] Wang HM, Tang YB, Yin G, Li L. Trustworthiness of Internet-based software. Science in China (Series E), 2006,36(10): 1156−1169 (in Chinese).

[6] Willsky AS. A survey of design methods for failure detection in dynamic-systems. Automatica, 1976,12(6):601−611.

[7] Frank PM. Fault-Diagnosis in dynamic-systems using analytical and knowledge-based redundancy—A survey and some new results. Automatica, 1990,26(3):459−474.

[8] Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D. Diagnosability of discrete-event systems. IEEE Trans. on Automatic Control, 1995,40(9):1555−1575.

[9] Zad SH, Kwong RH, Wonham WM. Fault diagnosis in discrete-event systems: Incorporating timing information. IEEE Trans. on Automatic Control, 2005,50(7):1010−1015.

[10] Debouk R, Lafortune S, Teneketzis D. Coordinated decentralized protocols for failure diagnosis of discrete event systems. Discrete Event Dynamic Systems—Theory and Applications, 2000,10(1-2):33−86.

[11] Sampath M, Lafortune S, Teneketzis D. Active diagnosis of discrete-event systems. IEEE Trans. on Automatic Control, 1998,43(7): 908−929.

[12] Ushio T, Onishi I, Okuda K. Fault detection based on Petri net models with faulty behaviors. In: Zadeh LA, ed. IEEE Int'l Conf. on Systems, Man, and Cybernetics. New York: IEEE Computer Society Press, 1998. 113−118.

[13] Aghasaryan A, Fabre E, Benveniste A, Boubour R, Jard C. Fault detection and diagnosis in distributed systems: An approach by partially stochastic Petri nets. Discrete Event Dynamic Systems—Theory and Applications, 1998,8(2):203−231.

[14] Jiang SB, Huang ZD, Chandra V, Kumar R. A polynomial algorithm for testing diagnosability of discrete-event systems. IEEE Trans. on Automatic Control, 2001,46(8):1318−1321.

[15] Yoo TS, Lafortune S. Polynomial-Time verification of diagnosability of partially observed discrete-event systems. IEEE Trans. on Automatic Control, 2002,47(9):1491−1495.

[16] Yoo TS, Lafortune S. On the computational complexity of some problems arising in partially-observed discrete-event systems. In: AACC, ed. Proc. of the American Control Conf. New York: IEEE Computer Society Press, 2001. 307−312.

[17] Lin F. Analysis and synthesis of discrete-event systems using temporal logic. Control-Theory and Advanced Technology, 1993, 9(1):341−350.

[18] Pandalai DN, Holloway LE. Template languages for fault monitoring of timed discrete event processes. IEEE Trans. on Automatic Control, 2000,45(5):868−882.

[19] Chang L, Shi ZZ, Qiu LR, Lin F. A Tableau decision algorithm for dynamic description logic. Chinese Journal of Computers, 2008, 31(6):896−909 (in Chinese with English abstract).

[20] Contant O, Lafortune S, Teneketzis D. Diagnosis of intermittent faults. Discrete Event Dynamic Systems—Theory and Applications, 2004,14(2):171−202.

附中文参考文献:

[1] 杨芙清,梅宏,吕建,金芝.浅论软件技术发展.电子学报,2002,30(12A):1901−1906.

[2] 杨芙清.软件工程技术发展思索.软件学报,2005,16(1):1−7. http://www.jos.org.cn/1000-9825/16/1.htm

[3] 吕建,陶先平,马晓星,胡昊,徐锋,曹春.基于 Agent 的网构软件模型研究.中国科学(E 辑),2005,35(12):1233−1253.

[5] 王怀民,唐扬斌,尹刚,李磊.互联网软件的可信机理.中国科学(E 辑),2006,36(10):1156−1169.

[19] 常亮,史忠植,邱莉榕,林芬.动态描述逻辑的 Tableau 判定算法.计算机学报,2008,31(6):896−909.

王竹晓(1981－),男,四川自贡人,博士生,主要研究领域为分布式计算,知识表示和推理.

史忠植(1941－),男,研究员,博士生导师,CCF 高级会员,主要研究领域为人工智能,机器学习,神经计算.

杨鲲(1978－),女,博士生,主要研究领域为语义 Web,Web 服务,知识表示和推理.

第 7 届全国 Web 信息系统及其应用学术会议(WISA 2010)

征文通知

Internet 和 Web 给我们的社会带来了深刻影响。Web 信息系统已成为分布式应用系统的主流形式之一，在公众计算、企业计算和行业信息化中发挥日益重要的作用。全国 Web 信息系统及其应用会议(WISA)是中国计算机学会暨电子政务与办公自动化专委会主办的系列会议。前 6 届会议分别在武汉、沈阳、南京、北京、西安与徐州召开，最多时收到论文近 1000 篇。

WISA 2010 会议将于 2010 年 8 月 20 日~22 日在内蒙古大学召开。WISA 2010 会议将继续这一良好的传统，在 Web 信息系统核心技术、Web 信息系统应用框架和体系结构、Web 信息系统应用等方面进行深入广泛的探讨和交流。会议期间将同时举办两个研讨会：全国第 5 次语义 Web 与本体论学术研讨会(SWON 2010)和全国第 4 次电子政务技术及应用学术研讨会(EGTA 2010)。

大会及研讨会录用论文中主要论文以英文方式由 IEEE Computer Society Press (EI)正刊出版，其余论文将由核心期刊《计算机科学》专刊、《计算机与数字工程》正刊出版。会议期间除进行会议论文交流外，还将邀请著名专家做特邀报告，并继续评选大会优秀学生论文。

一、征文范围

征文范围包括(但不限于)：Web 信息挖掘与检索，Web2.0 与社会信息，Web 与数据库技术，XML 与半结构化数据管理，Web 信息集成，语义 Web 与智能 Web，Web 应用框架和体系结构，组件与中间件技术，Web 服务、SOA 与网格计算，工作流管理，Web 站点逆向工程与维护技术，多媒体数据管理，Web 测试与 Web 应用的质量保证，决策支持与分析技术，Deep Web 技术，Web 信息系统工程方法，Web 与信息系统安全性，基于 Web 的电子政务与电子商务框架及应用，Web 系统度量与分析技术，Web 信息系统实际应用经验。

二、来稿要求

1．本次会议主要通过网上投稿，尽量不要通过 E-mail 投稿，拒收纸质稿件。严禁一稿多投。

2．中英文稿均可，一般不超过 8000 字，为了便于出版论文集，来稿必须附中英文摘要、关键词、资助基金与主要参考文献，注明作者及主要联系人姓名、工作单位、详细通信地址（包括 E-mail 地址）与作者简介。稿件要求采用 WORD 或 PDF 格式。

三、联系信息

1．投稿地址：http://www.easychair.org/conferences/?conf=wisa2010
投稿联系人：华东师范大学　周傲英　周敏奇(mqzhou@sei.ecnu.edu.cn)

2．大会网站：www.neu.edu.cn/wisa2010

3．会务情况：内蒙古大学　高光来(cszjtao@imu.edu.cn)　周建涛(wisa2010@imu.edu.cn)

四、重要日期

征文截止日期：2010 年 3 月 31 日
录用通知发出日期：2010 年 4 月 30 日
正式论文提交截止日期：2010 年 5 月 20 日
会议召开日期：2010 年 8 月 20 日~22 日