基于限界模型检查的Web服务行为失配检测

戎 玫[1] 陈圣标[2] 张广泉[2,3]

(暨南大学深圳旅游学院 深圳 518053)[1] (苏州大学计算机科学与技术学院 苏州 215006)[2]

(中国科学院软件研究所计算机科学国家重点实验室 北京 100190)[3]

摘 要 在Web服务组合过程中,常因交互协议不一致等导致服务失配;Web服务失配检测可准确捕捉失配点,为实现服务的有效组合奠定基础。采用限界模型检查技术,提出一种基于可满足性模理论(SMT)的Web服务行为失配检测方法。该方法首先将服务失配检测问题转化为逻辑公式的可满足性判定问题,然后利用Yices工具实现Web服务行为失配检测,最后通过实例进一步阐述该方法的有效性。

关键词 限界模型检查,Web服务,行为失配检测,可满足性模理论

中图法分类号 TP311 **文献标识码** A

Detecting Behavioral Mismatch of Web Services Based on Bounded Model Checking

RONG Mei[1] CHEN Sheng-biao[2] ZHANG Guang-quan[2,3]

(Shenzhen Tourism College, Jinan University, Shenzhen 518053, China)[1]

(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)[2]

(State Key Laboratory of Computer Science, Chinese Academy of Sciences, Beijing 100190, China)[3]

Abstract Due to inconsistent of the interface type or interaction protocol, services can not be combined in the right way. Web services mismatch detection can accurately capture the mismatch points, which creates a foundation for realizing right interaction, avoid invalid composition. This paper presented a method for detecting mismatch of Web services based on Satisfiability Modulo Theories(SMT). The issue of detecting mismatch of Web services can be transformed into the problem of existence model checking that a deadlock is reachable or not between the interaction of the services, and the issue of existence model checking can be transformed into the problem that the logic formula is satisfiable or not. Finally, an example was given to explain the process of Web service mismatch detection.

Keywords Bounded model checking, Web service, Behavioral mismatch detection, Satisfiability modulo theories

面向服务的体系结构(Service Oriented Architectures, SOA)[1]作为一种新的软件开发范型,正逐步成为未来软件发展的趋势之一[2],作为其主流实现方式的Web服务技术目前得到了快速发展与应用。由于单个Web服务提供的功能有限,通常无法满足用户需求,因此有必要将多个服务组装成一个价值增值的、粒度更大的服务或系统,以满足复杂的应用需求。但在Web服务组合过程中,存在交互协议不一致等情况,导致服务失配而无法组合。Web服务失配检测可准确捕捉失配点,为实现服务的有效组合奠定基础。

Web服务失配可能发生在多个层次上,如签名层、行为层、语义层等,其中大部分的失配发生在行为层面。行为失配是诸多失配研究中的重点和难点[3]。在行为层,常见的失配类型有消息顺序失配、附加消息失配、缺失消息失配、一对多消息失配、多对一消息失配[4]。发生上述任何一种失配都可能导致服务交互过程产生死锁状态,死锁状态是不能向其它状态发生迁移的非终止状态。如果每个参与交互的服务最终都能到达各自的终止状态,即不存在死锁状态,则认为服务交互过程不存在失配;否则存在失配。

在Web服务行为失配检测方面,已有一些研究成果[5-8]。其中,文献[5]基于有限状态机模型,研究了两个Web服务交互的兼容性、等价性和可替换性问题。文献[6]提出了一种基于Petri网的Web服务失配检测方法,其将BPEL规范转化为Petri网描述,服务存在失配当且仅当Petri网模型中存在一个空的siphon。文献[7]使用π演算描述Web服务行为,通过π演算的操作语义以及自动推演实现服务兼容性的自动判定。由于π演算的进程是同步操作的,因此此方法无法解决异步通信模式下的服务兼容性判定问题。文献[8]使用模型检测工具SPIN验证了多个服务交互时是否满足兼容性。

本文研究Web服务行为层面的失配检测,将限界模型检查(Bounded Model Checking, BMC)[9]技术引入到Web服务

到稿日期:2011-07-21 返修日期:2012-03-06 本文受国家自然科学基金(60973149),江苏省自然科学基金(BK2011281),江苏省高校自然科学研究项目(08KJB520010,10KJB520019)资助。

戎 玫(1966-),女,博士,副教授,主要研究方向为网络软件工程与服务计算、嵌入式软件与形式化方法;陈圣标 男,硕士,主要研究方向为软件服务与形式化方法;张广泉 博士,教授,CCF高级会员,主要研究方向为服务计算、云计算与CPS、网络与分布式计算等,E-mail:gqzhang@suda.edu.cn。

失配检测中。首先建立 Web 服务行为的形式化模型，在此基础上提出了基于可满足性模理论(Satisfiability Modulo Theories,SMT)的 Web 服务失配检测方法，最后采用本文方法检测一个股票分析系统中的服务交互是否失配。

1 Web 服务行为的形式化模型

定义 1 Web 服务行为的形式化模型 W 为一个五元组 $W=\langle S,s_0,F,A,E\rangle$，其中：

- S 是有限状态集；
- $s_0\in S$ 是初始状态；
- $F\subseteq S$ 是终止状态集；
- $A\subseteq M\times\{?,!\}$，其中×运算是笛卡尔积，$M$ 是消息集合，A 是迁移动作集，对于任意的消息 $m\in M$，$m?$ 表示接收消息 m，$m!$ 表示发送消息 m；
- $E\subseteq S\times A\times S$，对于任意的一条迁移 $e\in E$，假设有 $e=(s,a,s')$，其中 s 为源状态，s' 为目标状态，a 是触发迁移的消息动作。

定义 2(多个 Web 服务并发组合) 对于一组参与交互的 Web 服务 $W_1,W_2,\cdots,W_{n-1},W_n,W_i=\langle S_i,s_{0i},F_i,A_i,E_i\rangle(i=1,2,\cdots,n)$，组合模型 $W=W_1\parallel W_2\parallel\cdots\parallel W_{n-1}\parallel W_n,W=\langle S,s_0,F,A,E\rangle$，其中：

- S 是状态集，S 中每个状态 s 为一个向量，$s=(s_1,s_2,\cdots,s_n),s_i\in S_i$；
- $s_0=(s_{01},s_{02},\cdots,s_{0n})$ 是初始状态；
- $F=F_1\times F_2\times\cdots\times F_n$ 是终止状态集；
- $A=(M_1\cup\cdots\cup M_n)\times\{?,!\},M_i$ 是 W_i 的消息集；
- E 是迁移关系集，$E\subseteq S\times A\times S$。任意一条迁移边 $e=\langle s,a,s'\rangle$，s 是源状态，s' 是目标状态，a 是迁移发生的消息动作。

2 基于 SMT 的 Web 服务行为失配检测

判定一个布尔公式的可满足性的系统称为 SAT 求解器(或 SAT 工具)。SMT 工具与 SAT 工具不同的是，SAT 工具仅能求解只包含布尔变量的逻辑公式，而 SMT 可以求解包含整型变量、实数型变量、线性运算的逻辑公式的可满足性问题，比如逻辑公式 $(x+2<y)\wedge(y>5)$。Yices 是一种 SMT 求解器，由 SRI 机构开发，支持整数和实数线性运算、数组、非递归函数等。

Web 服务行为兼容性指每个参与交互的服务最终能到达各自的终止状态，用 f 表示服务兼容性的否定。BMC 限定搜索范围，采用由局部到全局渐进式检测，能快速寻找到反例，它是目前高效的模型检测方法之一。采用 BWC 技术判定多个 Web 服务交互是否发生失配的主要过程是先将模型 W 的迁移关系 T 表示为逻辑公式，然后将 T 进行 k 步路径展开，得到逻辑公式 $[[W]]_k$；再将性质 f 转换成逻辑公式 $[[f]]_k$；最后将 $[[W,f]]_k$ 输入 SWT 工具 Yices 来判定公式的可满足性。如果公式可满足，则说明在 W 中存在一个实例满足性质 f，由此得出服务交互发生失配；否则，增加 k 的值重新构造 $[[W,f]]_k$ 并进行可满足性判定，直到公式可满足或超过求解器的运算能力。

2.1 模型 W 的 k 步路径转换为逻辑公式

设 $W=W_1\parallel W_2\parallel\cdots\parallel W_n$，其中 $W_i=\langle S_i,s_{0i},F_i,A_i,E_i\rangle(i=1,2,\cdots,n)$。将模型 W 的 k 步路径转换为逻辑公式，可分如下 3 步完成：

第一步　通过消息抽象处理消息；

第二步　描述模型的初始状态 I 和迁移关系 T；

第三步　由 T 得到模型的 k 步路径的表示。

2.1.1 消息抽象

定义 3(消息抽象) 设 $M=M_1\cup M_2\cup\cdots\cup M_n$，且 M 中共有 r 个消息。对于 $\forall m_i\in M$，定义函数 $Get(m_i)=i$，$Sent(m_i)=2*r+1-i$。

定义函数 Get 和 $Sent$，其分别用于描述消息的接收和发送情况，每个消息都与两个整数关联。对于 m_1 而言，$Get(m_1)=1$ 表示接收消息 m_1，$Sent(m_1)=2*r$ 表示发送消息 m_1。从上述定义可知，对任意一个消息，有 $Get(m)+Sent(m)=2*r+1$。消息 m_i 被抽象为整型变量 m_i,m_i' 为发生迁移后 m_i 的值。将消息抽象为变量，是为了模拟 Web 服务异步交互。变量的不同取值，表示消息在消息队列中的不同状态。如 $m_1=1$ 表示消息 m_1 已被接收，即消息已出队列；$m_1=2*r$ 表示已发送消息 m_1，此消息已入消息队列。

2.1.2 描述模型的初始状态 I 和迁移关系 T

设 Ω 为模型状态到整数集的映射。如果 $\Omega(s)=a$ 且 $a=Get(m)$，则表示状态 s 将要接收消息 m；如果 $\Omega(s)=b$ 且 $b=Sent(m)$，则表示状态 s 将要发送消息 m。如果 s 是终止状态，则 $\Omega(s)=0$。变量 at_i 和 at_i' 分别表示服务 W_i 的当前状态和迁移后的状态；整型变量 P_i 表示 W_i 将要接收或发送某个消息，P_i' 为发生迁移后 P_i 的值。令 $V=\{at_1,\cdots,at_n,P_1,\cdots,P_n,m_1,\cdots,m_r\}$。状态 s 为初始状态的表达式为：

$$I(s)=\bigwedge_{i=1}^{n}(at_i=s_{0i})\wedge\bigwedge_{i=1}^{r}(m_i=0)\wedge\bigwedge_{i=1}^{n}(P_i=\Omega(s_{0i}))$$

从上式可知，初始时，每个服务都处在初始状态，用整型变量 P_i 记录 W_i 在初始状态时将要接收或发送消息的情况，每个消息变量初始值为 0。

设 e 为 W_i 的任意一条迁移边，$e\in E_i$，$e=\langle s,a,s'\rangle$，则 $T(e)=(at_i=s)\wedge Con(a)\wedge(at_i'=s')\wedge(m'=d)\wedge(P_i'=\Omega(s'))\wedge Sam(V\backslash\{at_i,m,P_i\})$。如果 $a=m!$，则 $Con(a)=\text{true}$，$d=Sent(m)$；如果 $a=m?$，$Con(a)$ 就表示成 $m=Sent(m)$，此等式含义是判断 m 是否已被发送，$d=Get(m)$。$Sam(VAR)$ 表示 $v_1'=v_1\wedge v_2'=v_2\wedge\cdots\wedge v_q'=v_q$，其中 $VAR=\{v_1,v_2,\cdots,v_q\},VAR\subseteq V$。$Sam(V\backslash\{at_i,m,P_i\})$ 表示除变量 at_i,m,P_i 外，V 中其余变量值都不变。

迁移关系集 E_i 有 $T(E_i)=\bigvee_{e\in E_i}T(e)$。由此可得出 W 的迁移关系 $T(s,s')=T(E_1)\vee T(E_2)\vee\cdots\vee T(E_n)$。

如果模型的状态有不止一条接收消息动作的迁移(没有发送消息动作)，例如 $e_1=(s1,a?,s2)$ 和 $e_2=(s1,b?,s3)$ 是 W_i 中的两条迁移边，如果 $at_i'=s1$，则 P_i' 取值情况可表示为 $(P_i'=Get(a)\wedge P=Get(b))\vee(P_i'=Get(b)\wedge P=Get(a))$，$P$ 为新增加的变量。如果某个状态不止两条接收消息动作的出边，则相应地增加变量，采取类似处理。

2.1.3 模型 k 步路径的表示

变量 at_i_k、m_i_k、P_i_k 分别表示 at_i、m_i、P_i 在第 k 步的值。在 $I(s)$ 中，用 Var_0 替换 Var 得到 I_0，其中 $Var\in V$，I_0 表示第 0 步迁移。在 $T(s,s')$ 中，用 Var_0 替换 Var，Var_1 替换 Var' 得到 T_0，其中 $Var\in V$，T_0 表示第 1 步迁移。在 $T(s,s')$ 中，用 Var_1 替换 Var，Var_2 替换 Var' 得到 T_1；依次计算 $T_2,\cdots,T_{k-1}$。至此得到表示模型 W 的 k 步路径的逻辑

公式$[[W]]_k=I_0 \wedge T_0 \wedge T_1 \wedge \cdots \wedge T_{k-1}$。

2.2 将服务兼容性质转换为逻辑公式

本文的服务兼容性质用 f 表示，指服务交互过程存在死锁状态。如果公式$[[f]]_k$ 为真，则表示当前状态为死锁状态。死锁状态是非终止状态，用逻辑公式$\neg((at_1_k=F_1) \wedge (at_2_k=F_2) \wedge \cdots \wedge (at_n_k=F_n))$表示；对于服务 W_i，如果它当前即将发送消息或者所要接收的消息已发送，则 W_i 能发生状态迁移，用 $Con(W_i)$表示 W_i 能发生状态迁移的条件，则$Con(W_i)=(m_1_k+P_i_k=2*r+1) \vee \cdots \vee (m_r_k+P_i_k=2*r+1) \vee (P_i_k>r) \vee ExtraCon$。

等式 $m_j\_k+P_i\_k=2*r+1$ 成立，表示 W_i 将要接收的消息 m_j 已发送。由前面定义可知，m_j_k 的取值范围是 0、$Get(m_j)$、$Sent(m_j)$。当 P_i_k 表示接收消息时，有 $0 \leqslant P_i\_k \leqslant r$：

(1)若 $m_j\_k=0$，则 $m_j\_k+P_i\_k$ 取值范围是$[0,r]$，等式 $m_j\_k+P_i\_k=2*r+1$ 不可能成立；

(2)若 $m_j\_k=Get(m_j)=j$，此时 $1 \leqslant m_j\_k \leqslant r$，则 $1 \leqslant m_j\_k+P_i\_k \leqslant 2*r$，等式 $m_j\_k+P_i\_k=2*r+1$ 不成立；

(3)要使等式 $m_j\_k+P_i\_k=2*r+1$ 成立，m_j_k 只能是 $Sent(m_j)$，这时 $P_i\_k=2*r+1-Sent(m_j)=2*r+1-(2*r+1-j)=j=Get(m_j)$。因此，根据 $m_j\_k+P_i\_k$ 的值可以判定 W_i 所要接收的消息 m_j 是否已发送。

如果 W_i 中存在一个状态，它有不止一条接收消息动作的迁移边(没有发送消息的迁移边)，则在 $Con(W_i)$中通过增加 $ExtraCon$ 来处理。对于 2.1 节中的 e_1 和 e_2，$ExtraCon=(a\_k+P=2*r+1) \vee (b\_k+P=2*r+1)$。在当前状态下不能发生迁移的条件是$[[f]]_k=\neg(\bigwedge_{i=1}^{n}(at_i_k=F_i)) \wedge \neg(\bigvee_{i=1}^{n} Con(W_i))$。

2.3 利用 SMT 工具 Yices 判定公式的可满足性

将逻辑公式$[[W]]_k \wedge [[f]]_k$ 输入工具 Yices，判断其可满足性。如果公式可满足，则 Yices 输出 sat，并给出满足 f 的实例；如果公式不可满足，则 Yices 输出 unsat，此时需要做进一步转换：在 $T(s,s')$中，分别用 Var_k、$Var_(k+1)$替换 Var、Var'得到 T_k，其中 $Var \in V$，于是有$[[W]]_{k+1}=[[W]]_k \wedge T_k$；在$[[f]]_k$ 中，用 $k+1$ 替换 k 就得到$[[f]]_{k+1}$。这样就得到$[[W]]_{k+1} \wedge [[f]]_{k+1}$，$k$ 初值为 0。

3 实例分析

一个股票分析系统(Stock Analysis Service，SAS)包括 StockBroker、Investor 和 ResearchDept 3 个服务，对应的模型为 W_s、W_a 和 W_r，分别如图 1—图 3 所示。它们的交互过程如下：首先，Investor 发送 regist 消息请求给 StockBroker。若 StockBroker 拒绝该请求，则返回 reject 消息；若 StockBroker 接受该请求，则返回 accept 消息，并且发送 request 消息请求给 ResearchDept，之后 ResearchDept 将股票的分析结果 report 发送给 Investor。Investor 收到报告后决定是否继续接受服务，若继续，则发送 ack 确认消息给 StockBroker；否则发送 cancel 消息。StockBroker 收到 ack 消息后，发送一个 bill 账单消息给 Investor，并向 ResearchDept 发送 terminate 结束整个服务。

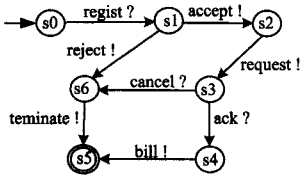

图 1 服务 StockBroker 模型

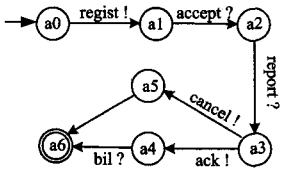

图 2 服务 Investor 模型

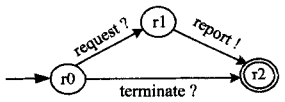

图 3 服务 ResearchDept 模型

3.1 W_s，W_a 和 W_r 的迁移关系到逻辑公式的转换

消息 regist(REG)、accept(ACC)、report(REP)、cancel(CAN)、ack(ACK)、bill(BIL)、reject(REJ)、request(REQ)、terminate(TEM)对应的有序对分别为(1,18)、(2,17)、(3,16)、(4,15)、(5,14)、(6,13)、(7,12)、(8,11)、(9,10)。变量 at，st，rs 分别表示 W_a，W_s 和 W_r 的当前状态，at'，st'，rs'为对应发生迁移后的状态；Pa，Ps，Pr，Pa'，Ps'，Pr'分别表示 W_a，W_s 和 W_r 将要接收或发送某个消息。变量集 V 为{at，st，rs，Pa，Ps，Pr，REG，ACC，REP，CAN，ACK，BIL，REJ，REQ，TEM}。

状态 s 的初始状态表达式为 $I(s)=\wedge(at=0)(st=0)(rs=0)(REG=0)(ACC=0)(REP=0)(CAN=0)(ACK=0)(BIL=0)(REJ=0)(REQ=0)(Pa=18)(Ps=1)(TEM=0)(Pr=8 \wedge P1=9 \vee Pr=9 \wedge P1=8)$。

W_a 中 a_0 到 a_1 的迁移 $T(a_0,a_1)$可表示为：

$T(a_0,a_1)=\wedge(at=0)(at'=1)(st'=st)(rs'=rs)(REG'=18)(ACC'=ACC)(REP'=REP)(CAN'=CAN)(ACK'=ACK)(BIL'=BIL)(REJ'=REJ)(REQ'=REQ)(TEM'=TEM)(Pa'=2)(Ps'=Ps)(Pr'=Pr)$。

将 W_a，W_s 及 W_r 的所有迁移边都转换为逻辑表达式后，得到系统迁移关系 $T(s,s')$：

$T(s,s')=\vee(\wedge(at=0)(at'=1)(REG'=18)(Pa'=2) \wedge Sam(V\backslash\{at,REG,Pa\}))$

$(\wedge(at=1)(ACC=17)(at'=2)(ACC'=2)(Pa'=3) \wedge Sam(V\backslash\{at,ACC,Pa\}))$

$(\wedge(at=2)(REP=16)(at'=3)(REP'=3)((Pa'=14) \vee (Pa'=15)) \wedge Sam(V\backslash\{at,REP,Pa\}))$

$(\wedge(at=3)(at'=4)(Pa'=6)(ACK'=14) \wedge Sam(V\backslash\{at,ACK,Pa\}))$

$(\wedge(at=3)(at'=5)(Pa'=19)(CAN'=15) \wedge Sam(V\backslash\{at,CAN,Pa\}))$

$(\wedge(at=5)(at'=6)(Pa'=0) \wedge Sam(V\backslash\{at,Pa\}))$

$(\wedge(at=4)(BIL=13)(at'=6)(Pa'=0) \wedge Sam(V\backslash\{at,BIL,Pa\}))$

$(\wedge(rs=0)(REQ=11)(rs'=1)(REQ'=8)(Pr'=16) \wedge Sam(V\backslash\{at,REQ,Pa\}))$

$(\wedge(rs=1)(rs'=2)(REP'=16)(Pr'=0)) \wedge Sam(V\backslash\{rs,REP,Pr\}))$

$(\wedge(rs=0)(TEM=10)(rs'=2)(TEM'=9)(Pr'=0) \wedge Sam(V\backslash\{rs,TEM,Pr\}))$

$(\wedge(st=0)(REG=18)(st'=1)(REG'=1)(Ps'=12 \vee Ps'=17) \wedge Sam(V\backslash\{st,REG,Ps\}))$

$(\wedge(st=1)(st'=2)(ACC'=17)(Ps'=11) \wedge Sam(V\backslash\{st,ACC,Ps\}))$

$(\wedge(st'=6)(st=1)(REJ'=12)(Ps'=10) \wedge Sam(V\backslash\{st,REJ,Ps\}))$

$(\wedge(st=2)(st'=3)(REQ'=11)((Ps'=4)\wedge(P2=5)\vee(Ps'=5)\wedge(P2=4))\wedge Sam(V\backslash\{st,REQ,Ps\}))$

$(\wedge(st=3)(ACK=14)(st'=4)(ACK'=5)(Ps'=13)\wedge Sam(V\backslash\{st,ACK,Ps\}))$

$(\wedge(st=3)(CAN=15)(st'=6)(CAN'=4)(Ps'=10)\wedge Sam(V\backslash\{st,CAN,Ps\}))$

$(\wedge(st=6)(st'=5)(TEM'=10)(Ps'=0)(at'=at)\wedge Sam(V\backslash\{st,TEM,Ps\}))$

$(\wedge(st=4)(st'=5)(BIL'=13)(Ps'=0)\wedge Sam(V\backslash\{st,BIL,Ps\}))$

其中，表达式 $Sam(V\backslash\{at,REG,Pa\})$ 是指 V 中除变量 at、REG、Pa 外，其余的变量值不变，$Sam(V\backslash\{at,REG,Pa\})=\wedge(st'=st)(rs'=rs)(ACC'=ACC)(REP'=REP)(CAN'=CAN)(ACK'=ACK)(BIL=BIL')(REJ=REJ')(REQ'=REQ)(TEM'=TEM)(Ps'=Ps)(Pr'=Pr)$。

3.2 股票分析系统的兼容性质到逻辑公式的转换

在本例中，死锁状态不是终止状态的逻辑公式表达为 $\neg((at\_k=6)\wedge(st\_k=5)\wedge(rs\_k=2))$。对于服务 W_a，在当前状态下能发生迁移的条件是 $Con(W_a)=\vee(Pa\_k+REG\_k=19)(ACC\_k+Pa\_k=19)(REP\_k+Pa\_k=19)(CAN\_k+Pa\_k=19)(ACK\_k+Pa\_k=19)(BIL\_k+Pa\_k=19)(REJ\_k+Pa\_k=19)(REQ\_k+Pa\_k=19)(Pa\_k>9)(Pa\_k+TEM\_k=19)$。

对于服务 W_s，在当前状态下能发生状态迁移的条件是 $Con(W_s)=\vee(REG\_k+Ps\_k=19)(ACC\_k+Ps\_k=19)(REP\_k+Ps\_k=19)(CAN\_k+Ps\_k=19)(ACK\_k+Ps\_k=19)(BIL\_k+Ps\_k=19)(REJ\_k+Ps\_k=19)(Ps\_k+REQ\_k=19)(TEM\_k+Ps\_k=19)(CAN\_k+P2=19)(ACK\_k+P2=19)(Ps\_k>9)$。增加 $P2$ 变量的是针对 W_s 中状态 $s3$ 有两条接收消息的出边。

对于服务 W_r，在当前状态下能发生状态迁移的条件是 $Con(W_r)=\vee(REG\_k+Pr\_k=19)(ACC\_k+Pr\_k=19)(REP\_k+Pr\_k=19)(CAN\_k+Pr\_k=19)(ACK\_k+Pr\_k=19)(BIL\_k+Pr\_k=19)(REJ\_k+Pr\_k=19)(REQ\_k+Pr\_k=19)(TEM\_k+Pr\_k=19)(CAN\_k+P_1=19)(ACK\_k+P1=19)(Pr\_k>9)$。

因此可得到 $[[f]]_k=\neg((at\_k=6)\wedge(st\_k=5)\wedge(rs\_k=2))\wedge\neg(Con(W_a)\vee Con(W_s)\vee Con(W_r))$。

3.3 Web 服务失配检测实验

图 4 为运行 Yices 判定逻辑公式 $[[W]]_k\wedge[[f]]_k$ 的可满足性的实验截图。表 1 为股票分析系统中服务失配检测情况。

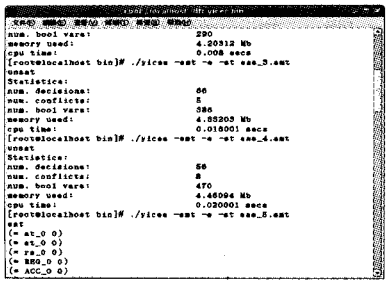

图 4 股票分析系统的失配检测过程

表 1 股票分析系统的失配判定实验结果

	内存消耗(Mb)	时间代价(S)	结果
k=0	3.949	0.002	unsat
k=1	4.074	0.004	unsat
k=2	4.203	0.008	unsat
k=3	4.332	0.016	unsat
k=4	4.460	0.02	unsat
k=5	4.589	0.028	sat

在实验中，一个 SMT 实例存放在一个文件中，实例 $[[W]]_0\wedge[[f]]_0$ 存在 sas_0.smt 文件中。从实验结果可知，当 $k=5$ 时，Yices 输出 sat，说明 $k=5$ 的实例是可满足式，由此可知服务 W_a、W_s 和 W_r 的交互过程存在失配。当 $at\_5=1$、$st\_5=5$、$rs\_5=2$ 时，W_s 和 W_r 已到达各自的终止状态，而 W_a 等待接收的消息 ACC 还没发送，所以此状态为死锁状态。从初始状态到当前死锁状态的交互过程为：

$$(a_0,s_0,r_0)\xrightarrow{REG!}(a_1,s_0,r_0)\xrightarrow{REG?}(a_1,s_1,r_0)\xrightarrow{REJ!}(a_1,s_6,r_0)\xrightarrow{TEM!}(a_1,s_5,r_0)\xrightarrow{TEM?}(a_1,s_5,r_2)。$$

结束语 本文围绕 Web 服务失配检测问题展开研究，采用限界模型检查技术，提出一种基于可满足性模理论(SMT)的 Web 服务行为失配检测方法，其将失配检测问题转化为逻辑公式的可满足性判定问题。最后举例进一步说明了本文方法的可行性和有效性。由于在一些应用场景中，服务间的交互通常具有时间约束，因此下一步工作将考虑具有时间约束的 Web 服务失配检测问题。

参 考 文 献

[1] Papazoglou M, Heuvel W. Service Oriented Archite-ctures: Approaches, Technologies and Research Issues[J]. VLDB Journal, 2007, 16(3): 389-415

[2] Gacek C, Gamble C. Mismatch Avoidance in Web Services Software Architectures[J]. Journal of Universal Computer Science, 2008, 14(8): 1285-1313

[3] Camara J, Salaun G, Canal C. Composition and Run-time Adaptation of Mismatching Behavioural Interfaces[J]. Journal of Universal Computer Science, 2008, 14(13): 2182-2211

[4] Kongdenfha W, Nezhad H, Benatallah B. Mismatch Patterns and Adaptation Aspects: A Foundation for Rapid Development of Web Service Adapters[J]. IEEE Transactions on Services Computing, 2009, 2(2): 94-107

[5] Benatallah B, Casati F, Toumani F. Representing, Analyzing and Managing Web Service Protocols[J]. Data & Knowledge Engineering, 2006, 58(3): 327-357

[6] Xiong P, Zhou M, Calton P. A Petri Net Siphon Based Solution to Protocol-level Service Composition Mismatch[C]// IEEE International Conference on Web Services(ICWS). Los Alamitos: IEEE Computer Society, 2009: 952-958

[7] 邓水光，李莹，吴健，等. Web 服务行为兼容性的判定与计算[J]. 软件学报，2007，18(12)：3001-3014

[8] Dong R, Zhao W, Luo X. Model Checking Behavioral Specification of BPEL Web Services[C]// The World Congress on Engineering. Springer, 2008: 198-203

[9] Biere A, Cimatti A, Clarke E, et al. Symbolic Model Checking Without BDDs[C]// International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS). Berlin: Springer, 1999: 193-207