时间行为协议状态空间约减算法

张振领[1] 贾仰理[1] 李舟军[2]
(聊城大学计算机学院 聊城 252059)[1] (北京航空航天大学计算机学院 北京 100191)[2]

摘 要 对复杂时间行为协议状态进行约减对于缓解形式化验证的状态空间爆炸问题,提高验证工具系统的效率、实用性等具有重要意义。分析了实时构件组合的几种形态,对基于时间行为协议的组合理论和状态空间爆炸问题进行了讨论,给出了时间行为协议的状态空间约减算法并进行了分析,给出了示例。

关键词 实时构件,时间行为协议,状态,约减
中图法分类号 TP311 **文献标识码** A

States Space Reduction Algorithm for Timed Behavior Protocol

ZHANG Zhen-ling[1] JIA Yang-li[1] LI Zhou-jun[2]
(School of Computer, Liaocheng University, Liaocheng 252059, China)[1]
(School of Computer Science & Engineering, Beihang University, Beijing 100191, China)[2]

Abstract It has great significance to relieve the state space explosion problem and improve the verification efficiency and practicality by state reduction. This paper analyzed the composition pattern of real-time components. The timed behavior protocols based composition and the state space explosion problem were discussed, and the state space reduction algorithm for timed behavior protocol was given, and the sample was presented finally.

Keywords Real-time component, Timed behavior protocol, States, Reduction

1 引言

随着实时构件系统在航空航天等安全、生命攸关系统中的广泛应用,在基于构件的软件开发过程中,从高层规约至最终实现,利用系统规约、模型检验等形式化方法来分析和验证构件系统和构件行为的正确性、构件交互的相容性,对提高构件及其组合的正确性、可靠性等可信性质具有重大意义[1]。形式化描述使用具有严格数学语法和语义定义的语言刻画系统行为,形式化验证则在形式化描述的基础上分析系统是否具有所期望的性质,主要分为模型检验(包括精化检验)和定理证明两种途径。通过形式化方法,可以确定复杂实时构件系统的边界,更精确地刻画系统的时序行为和各种非功能性需求,避免不同阶段开发人员对系统的理解歧义,并为系统的正确性、可靠性和安全性等各种可信性质的验证提供自动化手段。

模型检验是由美国卡耐基梅隆大学的 Clarke 等人在 20 世纪 80 年代首先提出的一种重要的形式化验证技术,它是通过搜索待验证软件系统模型的有穷状态空间来检验系统的行为是否具备预期性质的一种自动验证技术[2]。与传统的软件测试、仿真或定理证明等方法相比,模型检验技术具有自动化程度高、验证速度快等特点,当验证的性质不满足时,还能够给出反例,方便查错。但模型检验存在的一个难以解决的重要问题是"状态空间爆炸"问题,即随着所要验证的系统的规模的增大,模型检验算法所需的时间/空间开销往往呈指数级增长,因而极大限制了其实际应用范围。

文献[3]对基于模型检验的构件验证技术进行了分析和评述。文献[4,5]分别提出了基于时间行为协议的实时构件行为相容性和一致性验证理论及方法,主要是验证两个实时构件的组合行为是否存在错误、实时构件之间及构件各层次上的描述是否一致。这对于构件组合、精化、构件检索、复用(替代)等具有重要应用意义。为了检测组合行为是否满足相容性等特性,需要对组合的状态空间进行遍历搜索,在实际验证系统的开发中,对多个复杂时间行为协议进行验证时,由于需要遍历的状态数量巨大,验证等待时间就会过长。系统状态空间爆炸问题严重,从而制约了验证系统能够验证的系统的规模、效率。针对这一问题,本文围绕如何提高可验证的状态数量级,对构件组合中基于时间行为协议的状态空间约减算法进行了分析和研究,并给出了基于时间行为协议的状态空间约减算法。

本文第 2 节介绍基于时间行为协议的实时构件组合;第 3 节讨论基于时间行为协议的状态空间约减理论,给出基于时间行为协议的状态空间约减算法;最后总结全文并指出下一步的工作。

到稿日期:2011-08-19 返修日期:2012-02-02 本文受国家自然科学基金项目(90718017),山东省自然科学基金项目(ZR2011FL023),山东省软科学项目(2010RKE16007),聊城大学自然科学重点项目(x09032),山东省高校智能信息处理与网络安全重点实验室(聊城大学)资助。
张振领(1977—),女,硕士,讲师,主要研究方向为形式化方法、智能信息处理,E-mail:zhangzhenling@lcu.edu.cn;贾仰理(1976—),男,博士,主要研究方向为形式化方法等;李舟军(1963—),男,博士生导师,主要研究方向为形式化方法、信息安全、智能信息处理等。

2 基于时间行为协议的实时构件组合

2.1 实时构件组合形态

构件只有经过组装才可以形成可运行的应用系统。构件组装的过程是由一些子构件连接、交互、生成功能更加强大的复合构件的过程。组装形成的复合构件又可以作为更高层次的复合构件的子构件参与组装。

对任两个构件接口之间的关系的分析有助于分析构件的组装机制。对于任意两个构件实体，我们认为它们的接口之间可以存在以下关系：

(1)绑定关系

绑定关系指构件接口请求服务接口与另一构件接口上提供的对应服务接口之间建立的连接关系，如图 1 所示。

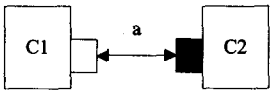

图 1 绑定关系图示

(2)委派关系

委派指复合构件某接口上的服务由子构件某接口来提供，两个接口之间形成了委派关系，如图 2 所示。

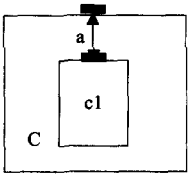

图 2 委派关系图示

(3)归并关系

归并指子构件某接口上的请求服务包含在复合构件接口上的请求服务接口中，两个接口之间形成了归并关系，如图 3 所示。

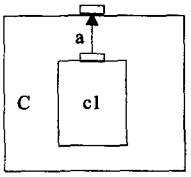

图 3 归并关系图示

(4)免除关系

免除是指子构件的某一接口在复合构件中不加以使用，即复合构件中不含有该接口。免除关系如图 4 所示。

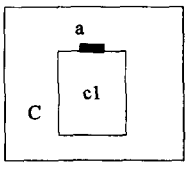

图 4 免除关系图示

上面是从构件接口的角度刻画构件之间的关系，刻画的是构件之间的连接交互关系。可以刻画处于同一层次上的构件间的关系，也可以刻画不同层次间的构件关系。

下面给出一个实际的数据库应用构件系统中构件接口之间关系的示意图。

图 5 为数据库应用构件系统的组成情况，它由数据库应用构件、数据构件、日志处理构件和数据存取构件组成，其中数据库构件又由数据处理构件、事务处理构件组成。数据处理构件在其接口 Ia 通过绑定关系使用事务处理构件的事务处理服务，在接口 Ib 通过委派关系向其父构件数据库构件提供应用服务，其请求服务在接口 Ic、Id 归并到了其父构件；Ie 接口上的方法则被免除，没有参与系统组合。

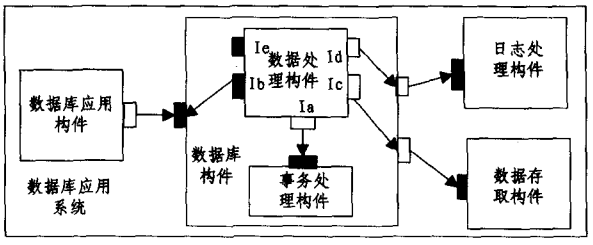

图 5 数据库应用系统构件接口关系图示

2.2 时间行为协议组合

首先给出时间动作交叠序列和时间行为迹的相关概念。

定义 1(动作交叠序列) 若 u,v,w 分别为 $U,V,(U\cup V)$ 上的有穷行为序列，n,m,l 分别是 u,v,w 的长度，若函数：

$$f(i)\left\{\begin{matrix}==0,\langle w_i\rangle=\langle u_j\rangle\\==1,\langle w_i\rangle=\langle v_k\rangle\end{matrix}\right\}$$

$i\in\{1,2,\cdots,l\}$，$j\in\{1,2,\cdots n\}$，$k\in\{1,2,\cdots,m\}$

则 w 是 u,v 的动作交叠序列(interleaving)。即 u,v 序列的交叠是它们中的动作元素在保持相对顺序的情况下的交错而形成的序列。

定义 2(时间行为迹) 对于一个时间行为协议 P，在事件集合 $A=\{?\ a[t_1],!\ a[t_2],?\ b[t_3],!\ b[t_4]\cdots\}$ 上，如果存在一个长度为 n 的动作交叠序列$\langle ?\ a[t_1],!\ b[t_4],\cdots\rangle$，每个动作事件都是可推进的，使得 P 能够一直推进到某个结束事件，则称 P 的变迁过程中所产生的动作序列为一个时间行为迹，记为：$T(P)/A$，且 $T(P)/A=\langle ?\ a[t_1],!\ b[t_4],\cdots\rangle$。

时间行为迹 $T(P)$描述了构件(时间行为协议为 P)的一个动态行为：经过有限步变迁后达到结束状态。

由于选择操作符和并行操作符的存在，使得一个构件的时间行为迹不是唯一的。

例如，考虑一个时间行为协议描述 P：

$P=?$ db. add{! tm. begin$[t_1]\uparrow$;(! tm. commit$\uparrow$ + ! tm. rollback$\uparrow$)}

在事件集合 $A=\{!$ db. add,? tm. begin$[t_1]$;? tm. commit,? tm. rollback$\}$上，则该构件的时间行为迹如下：

$T(P)/A=\langle ?$ db. add$\uparrow$,! tm. begin$[t_1]\uparrow$,! tm. commit$\uparrow$,! db. add$\downarrow\rangle$

或

$T(P)/A=\langle ?$ db. add$\uparrow$,! tm. begin$[t_1]\uparrow$,! tm. rollback$\uparrow$,! db. add$\downarrow\rangle$

定义 3(时间行为语言) 设 A 为时间行为协议 P 上的事件集合，则时间行为协议 P 在事件集合 A 上定义的时间行为语言 $L(P)/A$ 为时间行为协议在事件集 A 上所产生的所有的行为迹的集合。即：$L(P)/A=\{T(P)/A\}$。

定义 4(时间行为协议的组合) 时间行为协议的组合需

要考虑两协议之间的交互。假设 A 和 B 分别为时间行为协议 P 与 Q 的事件集合。在交互情况下，时间行为协议 P 或 Q 执行 send a 事件，时间行为协议 Q 或 P 则执行 receive a 事件，时间行为协议 P 与 Q 对事件进行交互，在组合操作下，则生成内部事件。

时间行为协议交互动作可以分为进化动作和迁移动作。经过进化动作，系统从状态 p 进化(evolution)到状态 p'，记为 $p \xrightarrow[t]{} p'$；如果时间标记迁移系统在 p 状态处理特定的事件(内部事件 τ 或外部事件 a)，并进入下一状态 p'，这种动作称为迁移动作。经过迁移动作，系统从状态 p 迁移到 p'，记为 $p \xrightarrow{(t,\tau)} p'$(内部迁移)或 $p \xrightarrow{(t,a)} p'$(外部迁移)。我们用结构化操作语义(SOS)的风格定义时间行为协议的组合规则如下。

从进化角度看，

$$\text{Composition}\ \frac{P \xrightarrow[t]{} P', Q \xrightarrow[t]{} Q'}{P \bigcap\nolimits_X Q \xrightarrow[t]{} P' \bigcap\nolimits_X Q'}$$

从迁移角度看，

$$\text{Composition}\ \frac{P \xrightarrow{(t,!\ a)} P', Q \xrightarrow{(t,?\ a)} Q'}{P \bigcap\nolimits_X Q \xrightarrow{(t,!\ a \wedge ?\ a)} P' \bigcap\nolimits_X Q'}$$

$$[!\ a \in A, ?\ a \in B, a \in X]$$

设两个时间行为协议 P,Q，它们组合形成的时间事件序列为 $L(P)$和 $L(Q)$中事件标识的交叠序列，要求在给定事件集合 X 上同步，在满足时序要求的情况下，任意子序列！interface. method [t]，? interface. method [t] 或? interface. method[t]，! interface. method[t]组合为 τinterface. method。

即时间行为协议的组合序列为两协议形成事件序列的交叠序列，对指定事件集合内的事件，该交叠序列需要完成发出事件(!)和接收事件(?)的匹配，生成内部(τ)事件，如果部分事件 $x \in X$ 没有完成匹配，则从组合后的路径中删除。

在理想情况下，事件 $x \in X$ 完全匹配，即时间行为协议 P 或 Q 行为序列执行 send a 事件，时间行为协议 Q 或 P 相对应的行为序列正好执行 receive a 事件，则这两个时间行为协议 P 与 Q 对事件进行交互，生成内部事件。除特殊情况外，可以不考虑内部事件的时间要求。

例如：

P=！tm. begin[1];(！tm. commit[1]+！tm. rollback[1])

Q=？tm. begin[1];？tm. rollback[1]

则 P 和 Q 在集合 X={begin,commit,rollback}上的组合为：

$P \bigcap_x Q = \tau$tm. begin;τtm. rollback

组合后的时间行为协议不包含事件 tm. commit，这是因为组合操作符右面要求必须出现 rollback 事件，而 rollback 和 commit 事件只能选择其一执行。

3 时间行为协议约减技术

3.1 时间行为协议约减的必要性

与行为协议类似，对于一些复杂时间行为协议，用状态迁移形式表示后系统状态空间是惊人的[6]。例如：表示时间行为协议 $a[t]$;$b[t]$和时间行为协议 $c[t]$;$d[t]$各需要 3 个状态(假设不考虑时间)，而表示它们的与并行操作后的结果，则需要 9 个状态。

一般来说，两个分别需要 m 和 n 个状态来表示的协议做与并行组合操作，结果状态数量将达到 $m \times n$ 个，而迁移数量将达到 $m \times n' + n \times m'$($m'$,$n'$分别为原协议的迁移数目)，如图 6 所示。

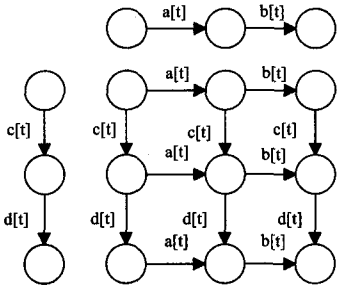

图 6 与并行操作后状态空间图示

对于复杂的时间行为协议，它们组合后的状态空间将呈指数级增长。为了检测组合行为是否满足相容性或可替换性，需要对组合成的自动机进行状态空间搜索，在对由多个复杂时间行为协议组合而成的时间行为协议进行验证时，由于需要遍历搜索的状态数量巨大，因此验证等待时间会过长。所以，必须考虑状态空间约减问题，以求在一定程度上缓解状态空间爆炸问题。

为解决这一问题，考虑在组合验证前对单一或组合时间行为协议先进行精简。

3.2 时间行为协议约减算法

考虑时间行为协议的约减问题，首先看一个例子，考虑构件 A、B 的时间行为协议 P、Q 分别如下：

P=？$a[t_1]$(！$x[t_2]$)* |(？$b[t_3]$)*

Q=！$a[t_1]$(？$x[t_2]$+？$y[t_3]$)*

显然，对两个构件的行为进行分析可以发现，构件 A 中的 b 方法和构件 B 中的 y 方法在 A,B 组合时不会用到，在这一特定组合中(不考虑其它构件调用 b 方法或 y 方法)，时间行为协议可以简化为：

P=？$a[t_1]$(！$x[t_2]$)*

Q=！$a[t_1]$(？$x[t_2]$)*

这是因为在针对一些特定应用的构件组合时，有些方法一直不会用到，所以可以在组合构件的时间行为协议中将和这些方法相关的状态进行约减，由此简化时间行为协议，从而降低时间行为协议的复杂性，提高其可读性。针对一个具体的特定应用，约减后的时间行为协议和原时间行为协议应满足可替换性。

显然，如图 5 所示，接口 Ie 在构件组装中被“免除”，对这类构件接口，完全不必考虑其上的时间行为协议。

定义 5(约减时间行为协议) 设 P,Q 为时间行为协议，如果 $L(P) \subseteq L(Q)$，且 P 可以替代 Q，则称 P 为 Q 的约减时间行为协议。

时间协议 Q 能否约减为另一协议 P，主要需要考虑协议 P 在与 Q 的环境行为协议组合时能否代替 Q，这是时间行为协议约减的核心准则。

时间行为协议约减算法如下：

算法输入：约减前的时间行为协议

算法输出：约减后的时间行为协议

生成两个子协议可产生的所有迹

遍历两子协议组合可产生的所有迹 T(c)

(1)If 任一事件 $e_i \in S(T(c1))$ 或 $e_i \in S(T(c2))$，$e_i \in S(T(C))$ then visited(e_i)：=1；

(2)else visited(e_i)：=0；

(3)对于任一由'+','|',';'连接的子协议 b1，Trace(b1)∈L(b1)

(4)If 任意 Firstofchar(Trace(b1))！='?' then onemit(b1)：=false，

(5)If 〈〉∈Trace(b1) then nullable(b1)：=true

(6)读取标识后的子协议串，按下面的规则约减：

(7)Case 1：if 所有 $e_i \in S(T(b1))$ visited(e_i)：=0，并且 onemit(b1)：=false then b1+b2 中 b1 可以约减掉；

(8)Case 2：if 所有 $e_i \in S(T(b2))$ visited(e_i)：=0 并且 onemit(b2)：=false then b1+b2 中 b2 可以约减掉；

(9)Case 3：if 所有 $e_i \in S(T(b1))$ visited(e_i)：=0 并且 onemit(b1)：=false，nullable(b1)：=true then b1|b2 中 b1 可以约减掉；

(10)Case 4：if 所有 $e_i \in S(T(b2))$ *visited*(e_i)：=0 并且 onemit(b2)：=false，nullable(b2)：=true then b1|b2 中 b2 可以约减掉；

(11)Case 5：if 所有 $e_i \in S(T(b1))$ visited(e_i)：=0 并且 onemit(b1)：=false，nullable(b1)：=true then b1；b2 中 b1 可以约减掉；

(12)Case 6：if 所有 $e_i \in S(T(b2))$ visited(e_i)：=0 并且 onemit(b2)：=false，nullable(b2)：=true then b1；b2 中 b2 可以约减掉；

(13)Case 7：if 所有 $e_i \in S(T(b))$ visited(e_i)：=0 并且 onemit(b)：=false，nullable(b)：=true b！=NULL then b 可以约减掉。

算法的基本思想是：为了验证组合中的两个时间行为协议是否可以约减，首先把组合中用到的事件标识出，标识 visited 为 1 的肯定不能约减掉，而其它的则可以考虑是否可以约减，对于用'+'、'|'和';'连接的子协议的协议片断，分别考察它们是否可产生空迹或能否直接发出调用事件，然后根据情况判定是否可以将它们约减掉。

算法考虑了时间行为协议涉及的';'、'+'和'|'，即事件的顺序、选择和并发操作，因此该约减算法只可以解决由';'、'+'和'|'表示的协议的约减问题，但由于其他操作符可以由这 3 种基本操作符表示，例如事件的循环操作就可以表示为事件的顺序操作序列，因此如果要对包含其它复杂操作符的协议进行约减时，其必须先转换为仅由';'、'+'和'|'表示的协议。

算法首先对组合行为迹进行遍历，对组合涉及到的行为协议中的所有事件进行标识，然后只需对这些协议的操作符和标识符进行分析、判断即可，因此算法可以在有限时间内完成协议的约减检验。

下面利用协议语法树以图形方式给出一个时间行为协议约减的示例。

对于时间行为协议 $P=!\,a[t_1];(?\,x[t_2]+?\,y[t_3])^*$，给出协议的语法树，如图 7 所示。

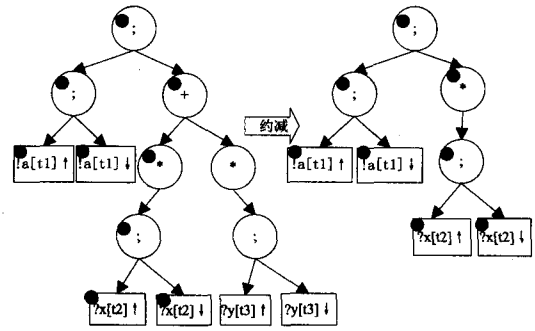

图 7　时间行为协议约减图示

相对于另一协议 $Q=?\,a[t_1];(!\,x[t_2])^*$，协议片断 $(?\,y[t_3])^*$ 没有被访问（*visited*(e_i)：=0），P 协议可以约减为 $P=!\,a[t_1];(?\,x[t_2])^*$。

显然，对于时间行为协议 P 和 Q，如果两个协议产生的行为迹（语言）满足包含关系，且 P 可以替代 Q，则 P 为 Q 的约减时间行为协议。即约减协议一般仅能够产生被约减协议在特定应用中的部分行为迹，这些迹往往都是由外围环境触发，约减协议可以代替被约减协议来响应这些触发事件。

常见的状态空间约减技术包括程序切片、偏序约简、数据抽象等，文献[3]对构件形式化验证涉及的状态约减问题进行了简述，本文提出的约减算法是专门针对时间行为协议的约减算法，它综合使用了程序切片和偏序约减的思想。目前尚未见到其它针对时间行为协议的约减算法。

结束语　实时构件系统在航空航天、医疗、交通管理等高可靠性需求领域有着广阔的应用前景。基于形式化描述和验证方法，在构件开发、选择、组装等环节保障这类复杂实时构件系统在功能和性能上的正确性与可靠性，防止灾难的发生，是我们所面临的主要挑战之一。本文提出的时间行为协议状态约减算法为基于时间行为协议的形式化验证提供了可行性。在今后工作中，将进一步开发构件实时行为验证实用工具。

参 考 文 献

[1] 李建中. 信息物理融合系统(CPS)的概念、特点、挑战和研究进展[R]. 2009 年中国计算机科学技术发展报告，2010：1-17

[2] Clarke E M，Grumberg O，Peled D A. Model Checking [M]. Cambridge，Massachusetts，USA：The MIT Press，2000

[3] 贾仰理，李舟军，邢建英，等. 基于模型检验的构件验证技术研究进展[J]. 计算机研究与发展，2011，48(06)：913-922

[4] 贾仰理，张振领，李舟军. 构件行为协议实时性扩展及相容性验证[J]. 计算机科学，2010，37(10)：143-1147

[5] 张振领，贾仰理，李舟军. 基于协议的实时构件行为一致性验证[J]. 计算机科学，2010，37(10)：143-147

[6] Mach M，Plasil F，Kofron J. Behavior Protocol Verification：Fighting State Explosion [J]. Int Journal of Computer and Information Science，2005，6(1)：22-30