

一种刻画功能和时间空间性能的统一验证模型 atsFPM

钮 俊^{1),2),3)} 曾国荪^{1),2)} 陈 波^{1),2)}

¹⁾(同济大学计算机科学与技术系 上海 201804)

²⁾(嵌入式系统与服务计算教育部重点实验室 上海 201804)

³⁾(浙江工商职业技术学院信息工程学院 浙江 宁波 315012)

摘 要 对复杂信息系统的功能、性能进行组合验证,进而评估系统是否安全、可信,是当前的研究热点,但目前缺乏增加空间约束的验证模型.文章扩展已有的功能、性能验证模型,在状态空间上定义空间要求函数,提出一种刻画功能、时间和空间性能的统一验证模型 atsFPM.给出基于正则式的路径范式描述信息系统行为的功能属性,给出 atsFPM 模型的语法和语义,构造路径范式与系统模型的积自动机,证明积自动机与原始模型在功能刻画、时间和空间描述上的等价,提出针对 atsFPM 的功能性能组合模型验证算法.实例分析表明,atsFPM 统一验证模型能够有效解决信息系统功能和性能的组合分析问题,确保系统正确、安全、可信.

关键词 功能性能;空间约束;组合验证;安全可靠

中图法分类号 TP301

DOI号: 10.3724/SP.J.1016.2009.00740

An Integrated Verification Model atsFPM Based on Functional, Time and Spatial Constraints

NIU Jun^{1),2),3)} ZENG Guo-Sun^{1),2)} CHEN Bo^{1),2)}

¹⁾(Department of Computer Science and Technology, Tongji University, Shanghai 201804)

²⁾(Key Laboratory of Embedded System and Service Computing of Ministry of Education, Shanghai 201804)

³⁾(College of Information Engineering, Zhejiang Business Technology Institute, Ningbo, Zhejiang 315012)

Abstract Nowadays complex information system's integrated formal models of function verification and performance evaluation lack properties constraint about space aspect. This paper presents an integrated verification model atsFPM by defining a space requirement function over the states of the considered information system. The patterns of paths which are based on regular expressions is proposed in order to specify the functional specifications. The syntax and semantic of the model atsFPM is defined. A conversion product model is obtained by the combination of the system model and the automaton of the pattern of paths which expresses the functional specifications. The verification of the model atsFPM is tackled by the performance verification technique of Markov Reward Model. Experimental results show that the atsFPM model and its verification approach can satisfy the modeling of information system and verification of functional and performance specifications.

Keywords function and performance; spatial constraint; integrated verification; secure and trusted

收稿日期:2008-12-05;最终修改稿收到日期:2009-01-23. 本课题得到国家“八六三”高技术研究发展计划项目基金(2007AA01Z425)、国家“九七三”重点基础研究发展规划项目基金(2007CB316502)和国家自然科学基金(90718015,60673157)资助. 钮 俊,男,1976年生,博士研究生,讲师,主要研究方向为可信软件、软件验证. E-mail: tongjinj@gmail.com. 曾国荪,男,1964年生,博士,教授,博士生导师,主要研究领域为软件验证、信息安全. 陈 波,男,1963年生,博士研究生,副教授,主要研究方向为可信软件、软件验证.

1 引言

各种具有并发性、分布性、动态性的信息系统普遍存在,如并行计算机系统、通信系统、网络系统以及各种通信协议^[1]、安全协议^[2]等非实体系统.对信息系统的功能和性能进行分析和研究,一直是计算机科学领域中的重要课题^[3].已有的研究工作中,通常是从功能、时间性能、空间能力3个方面独立地对系统进行形式化建模和分析.在功能分析方面,用LTS建立系统的动态行为模型^[4],用时态逻辑如LTL、CTL、CTL*等刻画系统的功能属性,如安全性、活性、公平性、可达性或永动性等,用符号法、标记法或基于自动机的方法等验证系统是否满足给定性质.在时间分析方面,由于大量复杂信息系统往往具有时间、概率等特性,这些特性决定了系统的随机时序性能,一般用连续时间 Markov 链建立系统的时间性能模型,用逻辑 CSL(Continuous Stochastic Logic)刻画时序性质^[5].在空间分析方面,由于系统在运行中,要与上下文(运行环境)进行空间资源交互,这些资源可以是如存储空间、执行成本及各种量化的服务资源等伴随系统运行的非时间性信息,文献[6-8]用MRM(Markov Reward Model)建立系统的回报模型,用CSRL(Continuous Stochastic Reward Logic)逻辑刻画带有回报约束的系统性质,并给出模型验证方法.

以上3个方面的分析,常常是孤立进行的.但是,在实际应用中,既要保证系统的功能正确,又要关心诸如系统的平均响应时间、平均无故障时间等时间性能,还要关注诸如系统运行所需的网络带宽、内存空间的大小或能量消耗量等空间因素.因此,需要将功能、时间性能、空间性能组合起来,对系统进行统一分析和研究.

针对功能和性能的组合验证目前已经有一些研究工作.基本方法是在传统功能模型上添加随机参数,刻画时序性能,如随机 Petri 网、交互式 Markov 链、随机进程代数等.文献[9]为 Petri 网模型的迁移添加时间约束,对功能和时序性能进行组合建模.林闯教授^[10]等比较了进程代数和 Petri 网在性能评价上的差别,并讨论了二者的相互关系.文献[11-13]将进程代数、LTS 与连续时间 Markov 链正交结合,得到一种交互式 Markov 链模型 IMC(Interactive Markov Chains),将 CSL 扩展为 aCSL(action-based CSL),进行系统功能与时序性能的组合刻

画,并给出了相应的模型检测算法.基于 PDL(Propositional Dynamic Logic)^[14]的思想,Meyer-Kayser 博士在他的博士论文^[15]中提出用正则式表示系统的动作序列,将 aCSL 扩展为 aCSL+. Baier 教授等人提出一种同时带有动作、状态标记的 Markov 链模型^[16],系统的验证规范用 asCSL(CSL with actions, state labels)描述,用基于动作和状态组合符号的正则集表示功能属性,将系统模型与功能属性自动机进行整合,构造积 Markov 链,运用基于 CTMC(Continuous-Time Markov Chain)模型的 CSL 验证技术进行功能和时序性能的组合验证评估.文献[17]提出一种基于路径的 pathCSL 逻辑,用来刻画带有动作和时间约束的系统指标,并给出模型检测方法.

以上功能和时间性能组合的模型验证方法,缺点是没有将评估信息系统安全性、可靠性的另一重要属性参数——“空间性能”组合考虑进来,显然上述方法不够完善.在实际应用中,有时迫切需要对系统的空间性能进行刻画.例如,考虑某个移动信息系统,用户和系统所关心的是系统在一定的时间、空间约束下的正确响应,因此在考虑系统功能的基础上,还必须关注系统的响应时间、处理器占有率、存储器占有率等时间或空间上的性能指标.也就是说,系统动态行为是基于特定的时间、空间约束下的结果.又例如,在网络实体信任协商中,除了考虑协商协议的正确性之外,还要考虑协商过程中,协商步的数目和交换证书的数量等非功能、非时间性的空间指标.再举例说,在评估基于冗余的容错组合 Web 服务时,由于各个子服务的加载将会给响应时间、带宽等带来性能上的瓶颈,因此必须考虑参与组合的子服务的数量等非时间上的性能指标.因此,必须将系统的功能、时间性能和空间性能组合描述,否则将导致系统验证或评估结果不可信.在实体系统中,空间因素用来表示系统的实体资源消费或生产;在虚拟系统中,空间因素可以表示更为抽象的概念,如系统的“资源的能力,过程的数量”等.

不同于已有的刻画功能和时序性能的系统模型,本文给出一种刻画操作功能、时间性能及空间性能的统一验证模型 atsFPM(Function and Performance Model with actions, time and space labels).基于已有的功能、性能验证模型,给状态间的转移添加动作、转移率、空间要求的组合标记,用以刻画功能和描述性能,为信息系统的形式化验证提供了新的思路和方法.

2 实例要求和问题提出

评价一个系统的运行是否安全、可信,不仅要看系统是否满足功能需求,同时,对其时间、空间性能进行评估,是非常有必要的.事实上,一个系统的行为特性,往往与其时间、空间性能密切相关,它们之间的刻画互为条件.为了表述客观系统分析上的要求,提出问题,我们通过一个实例来帮助描述和解释.

例 1. 容错多处理器系统^[6]. 设一个容错多处理器系统由 3 个处理器、3 个存储器及 1 个通信部件组成,通信部件完成处理器与存储器之间的通信.系统在初始状态时,所有组件都能正常工作,每个处理器可并行执行元任务.但是在运行过程中,各个组件都有可能出现故障而不能正常工作,故障组件可在一定时间内得到修复.当 3 个处理器均出现故障时,系统将进入瘫痪状态,同样地,当 3 个存储器均出现故障或者通信部件出现故障时,系统也瘫痪,此时可通过重置操作让系统回到其初始状态.

为了后面分析方便,设单个处理器的故障率、修复率期望分别为每小时 p_f 、 p_r 次,类似地,单个存储器的故障率、修复率期望分别为 m_f 、 m_r ,通信部件的故障率、修复率期望分别为 c_f 、 c_r ,记系统从瘫痪状态回到初始状态的重置率为 s_r .

很显然,对于上述系统,除了关心其功能是否正确,也迫切需要把握该系统在工作过程中是否满足某些性能指标,从而得出其是否安全、可信的判断.时间性能方面,如正常工作的平均时间等;空间性能方面,如在正常工作过程中的平均存储消耗量、平均维护成本等.具体来说,我们可能会考虑以下类似问题:

- (1) 通信部件是临界独占资源,是否出现死锁;
- (2) 系统瘫痪前,故障组件能否得到修复;
- (3) 系统正常工作 1h 的概率为多少;
- (4) 在正常工作情况下,平均内存消耗量小于 2M 的可能性如何;
- (5) 系统正常工作 2h,维护成本小于 100 个单位成本的概率为多少;
-

问题(1)、(2)描述了系统的功能特性,可通过传统的基于功能的模型验证技术回答.问题(3)描述了该系统的功能和空间性能,可通过建立系统的随机模型,进行功能和空间性能的组合验证.问题(4)描

述了系统的功能和空间性能,可通过基于随机回报模型的验证方法得到解决.问题(5)要求在功能和空间性能的基础上,还需要考虑系统的空间约束,这是目前系统功能验证和性能评估的热点方向,要回答它们,需要建立形式化验证模型,统一刻画系统功能及时间空间性能指标,并运用模型检测技术,对其进行自动的分析、验证和评估.

3 验证模型的基本概念

为了表述清楚以及理解方便,本节先分别阐述功能验证模型和性能验证模型.本论文将提出的 atsFPM 模型是在现有模型基础上的扩展和完善.

定义 1. 功能验证模型,是一个标记变迁系统 LTS^[4],即四元组 (S, Act, T, s_0) . 其中, S 是非空的状态集合; Act 是非空的动作集合(包括空动作); $T = S \times Act \rightarrow S$ 是状态转移关系; s_0 是初始状态.

下面以例 1 为对象,建立容错多处理器系统的 LTS 模型.通过分析,系统存在如下状态:所有组件均正常工作,即初始状态,某些处理器或存储器出故障状态以及所有组件均出故障的瘫痪状态.为了表达简洁,令状态集合 $S = \{s_{3,3}, s_{3,2}, s_{3,1}, s_{2,3}, s_{2,2}, s_{2,1}, s_{1,3}, s_{1,2}, s_{1,1}, s_{0,0}\}$, $s_{3,3}, s_{0,0}$ 分别表示初始状态、瘫痪状态,其余状态 $s_{i,j}$ 表示 i 个处理器和 j 个存储器正常工作.用命题“p_on”表示至少一个处理器正常工作,命题“m_on”表示至少一个存储器正常工作,“c_on”表示通信部件正常工作,则状态 $s_{3,3}$ 被标记为 $\{\text{start}\}$, 状态 $s_{0,0}$ 被标记为 $\{\neg \text{p_on}, \neg \text{m_on}, \neg \text{c_on}, \text{off}\}$, 其余所有状态的状态标记为 $\{\text{p_on}, \text{m_on}, \text{c_on}, \neg \text{off}\}$, 其中,命题“starting”、“off”分别表示初始状态、瘫痪状态.系统的动作有处理器出故障、存储器出故障、修复处理器、修复存储器、重置系统,即 $Act = \{\text{Pfail}, \text{Mfail}, \text{Prepair}, \text{Mrepair}, \text{Reset}\}$. 该系统的功能验证模型如图 1 所示,图中方向一致(水平向左、水平向右、垂直向上、垂直向下)的所有变迁的动作标记相同,为了简略图形,这里略掉部分同名动作标记.

定义 2. 性能验证模型,是一个带标记的连续时间 Markov 链(Labeled-CTMC)^[5],即四元组 $(S, AP, \mathcal{L}, \mathcal{R})$. 其中, S 是状态集合; AP 是原子命题集合; $\mathcal{L}: S \rightarrow 2^{AP}$ 为状态标记函数; $\mathcal{R}: S \times S \rightarrow \mathcal{R}_{\geq 0}$ 为转移函数, $\mathcal{R}_{\geq 0}$ 为正实数集.

对于 $s_1 \in S, s_2 \in S$, 如果存在 $\lambda = \mathcal{R}(s_1, s_2) > 0$, 则表示在状态 s_1 和 s_2 之间存在转移率为 λ 的转移,

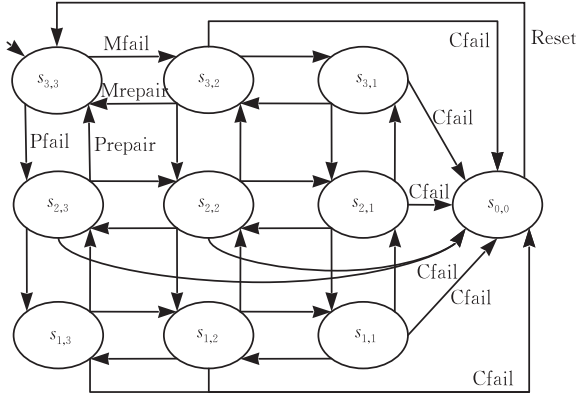


图 1 容错多处理器系统的功能验证模型

记为 $s_1 \xrightarrow{\lambda} s_2$, 该转移在时间 t 内发生的概率为 $1 - e^{-\mathcal{R}(s_1, s_2) \cdot t}$. 对 $s \in \mathcal{S}$, 令 $E(s) = \sum_{s' \in \mathcal{S}} \mathcal{R}(s, s')$ 表示从状态 s 出发的总转移率. 如果存在 $s^* \in \mathcal{S}$, 满足 $E(s^*) = 0$, 则称状态 s^* 为吸收态, 即不存在从该状态出发的任何转移. 用 $s \xrightarrow{t} s'$ 表示系统运行过程中的单步状态转移, 其中 t 表示该转移的延迟时间, 即在状态 s 的停留时间. 多个转移构成的转移序列表示一条执行路径. 转移序列

$$s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \dots \xrightarrow{t_{n-2}} s_{n-1} \xrightarrow{t_{n-1}} s_n \dots$$

表示一条无限路径, 其中, 对所有的非负整数 i , 满足 $s_i \in \mathcal{S}$, $\mathcal{R}(s_i, s_{i+1}) > 0$ 且 $t_i \in \mathcal{R}_{\geq 0}$. 如果 s_j 为吸收态, 则称转移序列

$$s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} \dots \xrightarrow{t_{j-2}} s_{j-1} \xrightarrow{t_{j-1}} s_j$$

为一条有限路径.

在性能验证模型中, 令 σ 为该模型中的某条有限路径, 记 $\sigma[i] = s_i$ 为路径中的第 $i+1$ 个状态. 用 $\delta(\sigma, i)$ 表示路径在状态 s_i 的停留时间, 即 $\delta(\sigma, i) = t_i$. 设 $\tau(\sigma) = \sum_{m=0}^{j-1} t_m$, 表示路径 σ 的花费时间. 当 $t \leq \tau(\sigma)$ 时, 路径 σ 在时刻 t 所处的状态用 $\sigma @ t$ 表示, 即 $\sigma @ t = \sigma[k]$, 其中, $k = \min_i (t \leq \sum_{j=0}^i t_j)$. 设 M 表示系统的性能验证模型, 记 $Path^M$ 表示其所有的有穷和无穷路径集合, $Path^M(s)$ 表示在 M 中, 从状态 s 出发的所有路径的集合. 通过构造集合 $Path^M(s)$ 上的 Borel 空间, 为路径集合定义概率测度 Pr_s^M , 简记为 Pr^M , 详细说明见参考文献[5-7]. 给定 M , 从状态 s 出发, 在时刻 t 处于状态 s' 的概率, 用 $\pi^M(s, s', t)$ 表示, 即 $\pi^M(s, s', t) = Pr^M\{\sigma \in Path^M(s) \mid \sigma @ t = s'\}$, 表示瞬时概率, $\pi^M(s, s') = \lim_{t \rightarrow \infty} (\pi^M(s, s', t))$ 表示稳态概率, 即一直停留于状态 s' 的概率. 令 $\mathcal{S}' \in 2^{\mathcal{S}}$, 记

$$\pi^M(s, \mathcal{S}') = \sum_{s' \in \mathcal{S}'} \pi^M(s, s'). \quad \sigma(i, j) \quad (0 \leq i \leq j) \text{ 表示路径 } \sigma \text{ 中介于状态 } s_i \text{ 和 } s_j \text{ 之间的路径片段.}$$

基于前面的相关描述, 例 1 中的容错多处理器系统的性能验证模型如图 2 所示.

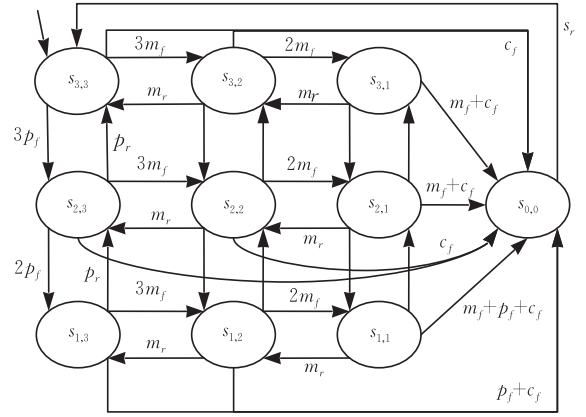


图 2 容错多处理器系统的性能验证模型

系统在运行过程中, 需要从运行环境中接收支撑运行的“输入”, 同时, 还将产生某些“输出”. 在本文中, 我们通过这些可观测的输入输出信息, 来刻画系统的另外一个重要方面: 空间性能, 下面给出空间信息的概念.

定义 3. 空间信息. 在运行过程中, 系统与环境交互的一切可观测资源信息, 包括生产和消耗两种情况.

上述定义说明, 我们可以通过运行过程中空间信息的生产或消耗来评估一个信息系统的空间性能, 刻画系统空间资源的消耗程度或生产能力, 得出在空间约束下是否安全、可信的判断. 这些空间信息可以是实体的, 比如电能、存储空间、网络带宽等, 也可以是非实体的, 如需要的活动组件数、交易证书数、事务步骤数、出错状态数等, 还可以是一些相对量, 如过载率、容错率、数据传输率等. 为了简洁, 本文中, 空间信息的生产或消耗统称为空间要求.

4 功能时间空间组合的统一验证模型 atsFPM

由前面的分析可知, 已有的分析模型侧重于功能和时间性能的分析, 缺乏空间要求信息的描述. 因此, 我们在已有模型的基础上进行扩展, 添加空间信息标记, 提出组合统一的 atsFPM 模型.

定义 4. 功能时间空间组合的统一验证模型 atsFPM, 是一个带有空间约束标记的随机分析模

型,即六元组 $(\mathcal{S}, Act, AP, \mathcal{L}, \mathcal{R}, \rho)$. 其中, $\mathcal{S}, Act, AP, \mathcal{L}$ 的含义与定义 1、2 中的类似, $\mathcal{R}: \mathcal{S} \times Act \times \mathcal{S} \rightarrow \mathcal{R}_{\geq 0}$ 为转移函数, $\rho: \mathcal{S} \rightarrow \mathcal{R}$ 表示状态的空间要求函数 (\mathcal{R} 表示实数集).

对于 $s_1 \in \mathcal{S}, s_2 \in \mathcal{S}$, 设 $\lambda = \mathcal{R}(s_1, a, s_2)$, 如果存在 $a \in Act$ 并满足 $\lambda > 0$, 则表示在状态 s_1 和 s_2 之间存在标记为 a 的转移, 即 $s_1 \xrightarrow{a, t} s_2$, 称 s_2 为 s_1 的直接后继, s_1 为 s_2 的直接前驱, s_1 为当前转移的出发状态, s_2 为当前转移的目标状态, 该转移的延迟时间 t 满足参数为 λ 的指数分布.

上述定义中的空间要求函数 ρ 的作用是为了描述在运行过程中, 当系统处于某状态时, 系统空间信息的要求率, 即单位时间内空间信息的要求量. 参照前面相关定义, 直观上, 如系统驻留于某个状态时单位时间内的耗电量、维护成本、数据传输量、信道带宽的占用量、必需活动构件的数量等, 都可以作为该函数的函数值, 刻画系统在运行过程中空间信息的要求情况. 它也可描述某些非具体资源的要求情况, 用来刻画如单位时间内处理服务的正确率、运行中经过某状态的频率等. 注意, 当系统发生转移时, 也会产生空间信息要求, 但在本文中, 我们只考虑基于状态的空间要求情况.

atsFPM 模型中路径及其相关定义与定义 2 中类似. 设 σ 为模型中一条路径, σ 在时刻 t 的累积空间资源要求总和定义为 $v(\sigma, t)$, 即当 $t = \sum_{j=0}^{k-1} t_j + t'$ 且 $t' \leq t_k$ 时, $v(\sigma, t) = \sum_{j=0}^{k-1} t_j \cdot \rho(s_j) + t' \cdot \rho(s_k)$, 其中 t_j 为

在状态 s_j 的停留时间.

由定义 4 可以看出, 统一分析模型 atsFPM 除了能够准确地描述动态系统的操作行为、时间性能, 同时也能刻画系统空间资源的生产或消耗. 在实际应用中, 这是非常有必要的. 相比以往的系统模型, 我们提出的模型 atsFPM 具有更加丰富的表达能力, 能从功能上、时序上、空间上准确地刻画应用系统, 从而为准确分析和验证系统奠定了强有力的基础. 仍以第 2 节中的例 1 为对象, 可定义空间资源要求函数如下:

(1) 当处于瘫痪状态时, 空间资源要求值为 1, 否则为 0, 即 $\rho(s_{0,0}) = 1, \rho(s_{i,j}) = 0 (1 \leq i, j \leq 3)$. 因此, 对于某条路径, 其空间要求值为 5, 表示沿着该路径系统进入瘫痪状态的次数累计为 5;

(2) 令 $\rho(s_{0,0}) = 0, \rho(s_{i,j}) = m \cdot (1 - (1 - 1/m)^l) (1 \leq i, j \leq 3)$, 其中, $l = \min(i, j)$, $m = \max(i, j)$, 则该要求函数表示, 当处于某状态时, 系统期望的可用带宽^[8];

(3) 令 $\rho(s_{0,0}) = 0, \rho(s_{i,j}) = 6 - (i + j) (1 \leq i, j \leq 3)$, 表示状态的维护成本;

.....

为了描述方便, 对于例 1, 这里定义其空间资源要求函数定义为上述 (3) 的情形, 对应的 atsFPM 模型如图 3 所示. 为了方便表达, 转移上的标记由 3 部分构成, 分别表示当前转移的动作标记、转移率以及该转移目标状态的空间要求值 (注意图中只标出了部分标记).

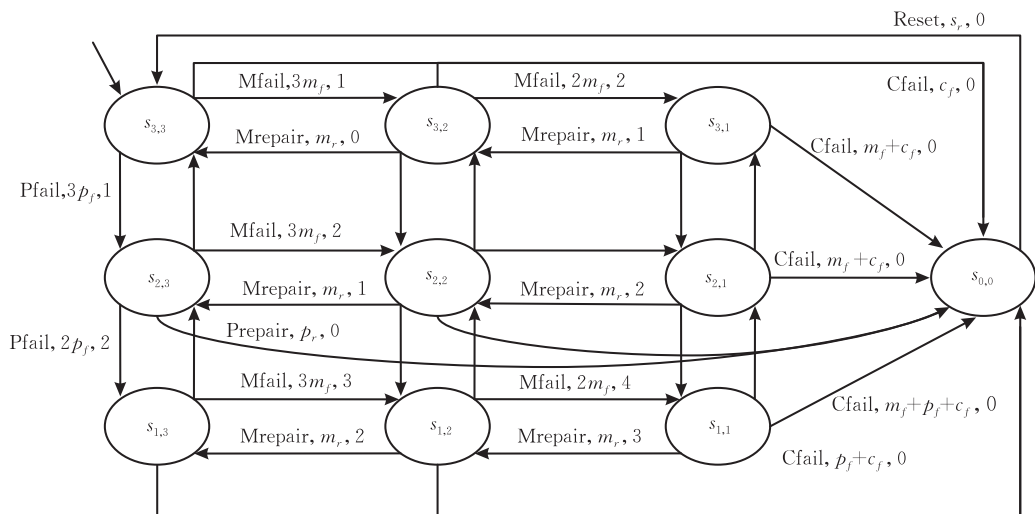


图 3 带有空间约束的功能性能统一验证模型 atsFPM

5 atsFPM 的语法和语义

通过沿用和扩展连续随机逻辑 CSL^[5]、连续随机回报逻辑 CSRL^[6]，并基于文献[17]的思想，下面给出在模型 atsFPM 中，路径公式和状态公式的语法和形式语义。

根据前面的定义，atsFPM 模型中的路径 σ 是指由形如 $s_{i-1} \xrightarrow{b_{i-1}, t_{i-1}} s_i$ 的转移所构成的转移序列。换句话说，一个转移表示在满足某条件 Φ 的状态上，经过一定的延迟时间，执行某个动作 b 而转移到另外一个状态。因此转移可由一个标识当前状态的公式 Φ 和一个动作 b 的有序对 (Φ, b) 表示，简记为 Φb 。将 Φb 看作一个迁移“符号”，一个路径便可用所有的这些符号构成的符号序列表示。在 atsFPM 模型中，形如 Φb 的“符号”表，用 Σ 表示，其中， Φ 表示状态应该满足的条件， b 表示在满足 Φ 条件的状态上将要执行的动作，即 $\Sigma = \Phi \times (Act \cup \{\sqrt{\cdot}\})$ ，标记为“ $\sqrt{\cdot}$ ”的动作作为一个内部动作，其执行不改变当前状态。在这里，为了便于处理，暂时忽略路径上转移的延迟时间和状态上的空间资源要求，我们给出路径范式的形式化定义。

定义 5. 路径范式 α ，表示模型 atsFPM 中路径上的操作应该满足的性质，即路径上的动作序列应该遵循的规范，表示如下：

$$\alpha ::= \epsilon \mid (\Phi b) \mid \alpha; \alpha \mid \alpha \cup \alpha \mid \alpha^*,$$

其中， ϵ 表示一条不包含任何转移的空路径， $\Phi b \in \Sigma$ ，“ $;$ ”表示顺序执行，“ \cup ”表示选择，“ $*$ ”表示任意次顺序执行。

在模型 atsFPM 中，一个路径范式 α 规定了路径上的操作序列应该满足的性质，它对应模型中的某个路径集合。若用 $Path^M(\alpha)$ 表示满足该路径规范 α 所限定的所有路径的集合，则通过判断一条路径 ζ 是否属于该集合，可确定该路径是否满足 α 指定的性质。

定义 6. 路径公式，刻画 atsFPM 模型中的执行路径应该满足的性质，用 α_S^T 表示。其中， α 为路径范式，区间 $T = [t, t'] \subseteq \mathcal{R}_{\geq 0}$ 表示路径上的时间约束区间， $S = [r, r'] \subseteq \mathcal{R}_{\geq 0}$ 表示空间约束区间。

具体来说，对于模型 atsFPM 中的某条路径 σ ，它满足路径公式 α_S^T ，意味着路径 σ 上的操作序列满足 α 指定的规则，且执行时间、累积空间要求值分别落在 T 区间和 S 区间内。实质上，一个路径公式对

应一个满足该公式的所有路径的集合。

定义 7. 路径公式的语义，表示路径与路径公式所表示的路径集合之间的从属关系。记 M 为当前讨论的 atsFPM 模型， ζ 为 M 中的无限路径， α_S^T 为路径公式，则路径 ζ 满足 α_S^T ，表示为 $M, \zeta \models \alpha_S^T$ ，当且仅当存在 ζ 的前缀 σ （即存在正整数 j ， σ 为 ζ 的路径片段 $\zeta(0, j)$ ），满足 $\sigma \in Path^M(\alpha)$ ，且 σ 分别满足时间约束 T 和空间约束 S 。其中，集合 $Path^M(\alpha)$ 递归定义如下：

$$\sigma \in Path^M(\epsilon) \text{ iff } length(\zeta) = 0,$$

$$\sigma \in Path^M(\Phi, b) \text{ iff } \sigma = s \xrightarrow{b, t} s' \text{ 且 } M, s \models \Phi,$$

$$\sigma \in Path^M(\Phi, \sqrt{\cdot}) \text{ iff } \sigma = s \text{ 且 } M, s \models \Phi,$$

$$\sigma \in Path^M(\alpha^1; \alpha^2) \text{ iff } \exists i (i \leq length(\sigma)), \sigma(0, i) \in Path^M(\alpha^1), \sigma(i, length(\sigma)) \in Path^M(\alpha^2),$$

$$\sigma \in Path^M(\alpha^1 \cup \alpha^2) \text{ iff } \sigma \in Path^M(\alpha^1) \cup Path^M(\alpha^2),$$

$$\sigma \in Path^M(\alpha^*) \text{ iff } \exists i > 0, \sigma \in Path^M(\alpha^i),$$

其中， $length(\sigma)$ 表示 σ 的长度，即路径 σ 中的转移个数。 α^i 表示 α 的 i 次顺序执行。

定义 8. 状态公式，表示模型 atsFPM 中，状态应该满足的性质，用 Φ 表示。定义如下：

$$\Phi ::= q \mid \neg \Phi \mid \Phi \wedge \Phi \mid S_{\bowtie p}(\Phi) \mid P_{\bowtie p}(\alpha_S^T),$$

其中， Φ 表示状态公式， q 是原子命题， p 表示概率值， \bowtie 表示比较运算算子，即 $\bowtie \in \{<, \leq, >, \geq\}$ ， T, S 代表非负实区间，含义与定义 6 中一致。 α 为路径范式， α_S^T 表示路径公式。

连接符 \neg 、 \wedge 的涵义与命题逻辑中一致，通过这两个连接符，可以表示其它的几个连接符或逻辑常量 true 或 false，比如 $false = \Phi \vee \neg \Phi$ 等。稳态算子 $S_{\bowtie p}(\Phi)$ ^[5] 表示最终处于 Φ 状态的概率满足限定 $\bowtie p$ ， $P_{\bowtie p}(\alpha_S^T)$ 表示其前缀满足路径公式 α_S^T 的路径的概率满足限定 $\bowtie p$ 。实际上，一个状态公式对应了一个状态集合，该集合中的所有状态均满足该状态公式。

定义 9. 状态公式的语义，表示状态和状态公式之间的可满足关系。记 M 为当前考虑的 atsFPM 模型， s 为 M 中的状态， M 中从状态 s 出发的所有路径集合定义为 $Path^M(s)$ 。 s 满足状态公式 Φ ，记为 $M, s \models \Phi$ ，定义如下：

$$M, s \models q \text{ iff } q \in \mathcal{L}(s),$$

$$M, s \models \neg \Phi \text{ iff } M, s \not\models \Phi,$$

$$M, s \models \Phi_1 \wedge \Phi_2 \text{ iff } M, s \models \Phi_1 \text{ 且 } M, s \models \Phi_2,$$

$$M, s \models S_{\bowtie p}(\Phi) \text{ iff } \pi^M(s, Sat^M(\Phi)) \bowtie p,$$

$$M, s \models P_{\bowtie p}(\alpha_S^T) \text{ iff } Prob^M(s, \alpha_S^T) \bowtie p,$$

其中 $Prob^M(s, \alpha_S^T) = Pr^M\{\zeta \in Path^M(s) \mid M, \zeta \models \alpha_S^T\}$ ，

表示从 s 出发并满足路径公式 α_s^T 的概率, $Sat^M(\Phi) = \{s \in S \mid M, s \models \Phi\}$ 代表公式 Φ 的可满足集合.

6 基于 atsFPM 的模型检测

对于所分析的系统,建立带有空间约束的功能性能统一验证 atsFPM 模型 $M = (S, Act, AP, \mathcal{L}, \mathcal{R}, \rho)$, 用状态公式刻画基于功能、时间、空间的系统规范 Φ , 其模型验证思想与 CTL 模型验证的思想类似^[4]: 对于状态 s , 计算状态公式 Φ 的可满足集合 $Sat(\Phi)$, 再判断 $s \in Sat(\Phi)$ 是否成立. 对于简单命题算子, 可以很方便地计算出其可满足集合, $Sat(S_{\approx p}(\Phi))$ 的计算与针对 CSL^[5] 的模型验证中的计算方法一致, 而 $Sat(P_{\approx p}(\alpha_s^T))$ 的处理则比较复杂: 对每个状态 s , 需要计算 $Prob^M(s, \alpha_s^T) = Pr^M\{\zeta \in Path^M(s) \mid M, \zeta \models \alpha_s^T\}$, 然后检查其是否在范围 $\approx p$ 内. 本文中我们的解决思路是将路径规则 α 用自动机表示, 与要考虑的模型 M 一起生成积 atsFPM, 然后运用基于 CSRL 的模型验证技术进行组合验证^[6].

6.1 构造路径范式的自动机

定义 10. 路径范式自动机. 路径范式 α 对应的非确定的自动机模型是一个五元组 $A_\alpha = (Z, \Sigma', \delta, Z_0, F)$. 其中, Z 为有限状态集, Σ' 为定义 5 中的符号表 Σ 的子集, $\delta: Z \times \Sigma' \rightarrow 2^Z$ 为转移函数, $Z_0 \subseteq Z$ 为初始状态集, $F \subseteq Z$ 为接受状态集.

设 M 为当前考虑的 atsFPM 模型, 由路径范式 α 的定义可知, α 对应于符号表 Σ 上的某个正则集 α' . 设 $L(\alpha')$ 表示正则集 α' 所接受的语言, 直观上容易得到 $L(\alpha')$ 等同于 $Path^M(\alpha)$, 只不过前者是操作序列的字符串表示. 设 σ 为 M 中的某条路径, 暂时忽略该路径上的时间标记, 可获得 σ 在 A_α 中的运行情况, 即路径所表达的行为的一次匹配判断^[16], 路径 σ 可能被 A_α 接受, 也可能被拒绝. 令 $Path^M(A_\alpha)$ 表示 M 中的某条路径 σ 在 A_α 中的可接受运行 (从 Z_0 中某状态出发, 最终到达终止状态 $z' \in F$) 路径的集合, $L(A_\alpha)$ 表示由路径模式 α 产生的自动机 A_α 的可接受语言, 则 $L(A_\alpha) = Path^M(A_\alpha)$, 由自动机理论, 可得到 $L(\alpha') = L(A_\alpha)$, 因此, 下式成立:

$$Path^M(\alpha) = Path^M(A_\alpha) \quad (1)$$

令 σ 表示 M 中的一条路径, $Z' \subseteq Z$ 表示自动机 A_α 中的状态子集, 从状态 $z \in Z'$ 出发, 可以直观地得到路径 σ 在 A 中的一次运行后最终可达的状态 z' .

定义 11. 最终可达状态集, 表示从自动机 A_α

的状态子集 Z' 中的某些状态出发, 模型 atsFPM 中路径 σ 在 A_α 中的所有可接受运行, 其运行轨迹中最后一个状态 z' 的集合. 用 $\delta^M(Z', \sigma)$ 表示. 于是

$$Path^M(\alpha) = \{\sigma \in Path^M \mid \delta^M(Z_0, \sigma) \cap F \neq \emptyset\} \quad (2)$$

6.2 系统模型与路径自动机的整合

定义 12. 积 atsFPM 模型. 令 M 表示一个 atsFPM 模型, 即 $M = (S, Act, AP, \mathcal{L}, \mathcal{R}, \rho)$, α 为路径范式, A_α 为由 α 构造出的自动机, 即 $A_\alpha = (Z, \Sigma', \delta, Z_0, F)$, M 与 A_α 的积 atsFPM 模型定义为

$$M \times A_\alpha = (S^{M \times A_\alpha}, Act^{M \times A_\alpha}, AP^{M \times A_\alpha}, L^{M \times A_\alpha}, R^{M \times A_\alpha}, \rho^{M \times A_\alpha}),$$

其中各元素的定义描述如下:

$$S^{M \times A_\alpha} = \{\langle s, Z' \rangle \mid s \in S \wedge Z' \in 2^Z\};$$

$Act^{M \times A_\alpha} = Act$, $AP^{M \times A_\alpha} = AP \cup \{accept\}$ ($accept \notin AP$, 标志命题, 标记积 atsFPM 中的终止状态);

如果 $Z' \cap F \neq \emptyset$, 则 $L^{M \times A_\alpha}(\langle s, Z' \rangle) = L(s) \cup \{accept\}$, 否则, $L^{M \times A_\alpha}(\langle s, Z' \rangle) = L(s)$;

设路径 σ' 为从状态 s_1 到状态 s_2 的一个动作标记为 b 的转移, 从 Z_1 中状态出发, 如果该转移在 A 中的最终可达状态集为 Z_2 , 则 $R^{M \times A_\alpha}(\langle s_1, Z_1 \rangle, b, \langle s_2, Z_2 \rangle) = R(s_1, a, s_2)$, 否则 $R^{M \times A_\alpha}(\langle s_1, Z_1 \rangle, b, \langle s_2, Z_2 \rangle) = 0$;

$$\rho^{M \times A_\alpha}(\langle s, Z' \rangle) = \rho(s).$$

从积 atsFPM 模型的定义中可以看出, 它是在系统 atsFPM 模型的基础上, 有机地融合了路径范式 α 所代表的操作行为规则, 产生的新的 atsFPM 模型, 而模型的时间、空间配置参数未变. 直观上, 我们得到下面两个结论.

命题 1. 模型 M 与模型 $M \times A_\alpha$ 在功能刻画上是等价的.

命题 2. 模型 M 与模型 $M \times A_\alpha$ 在时间、空间刻画上是等价的.

文献[16]对于只包含有时间约束的连续时间 Markov 链, 定义状态的等价关系, 进一步得到模型 M 与模型 $M \times A_\alpha$ 的路径之间的一一对应关系. 很显然, 带有空间约束的连续时间 Markov 链, 亦类似. 通过该方法, 我们得出 atsFPM 模型 M 与相应的积 atsFPM 模型 $M \times A_\alpha$, 二者在路径上也存在着映射关系, 这说明经过整合后的积 atsFPM 模型并未改变系统本身的功能特性. 限于篇幅, 具体细节可参考相关文献. 基于上述结论, 得到了下面两个命题.

命题 3.

$$Prob^M(s, \alpha_s^T) = Prob^{M \times A_\alpha}(\langle s, Z_0 \rangle, \alpha_s^T) \quad (3)$$

证明. 由前面的定义知 $Prob^M(s, \alpha_S^T) = Pr^M\{\zeta \in Path_\omega^M(s) \mid M, \zeta \models \alpha_S^T\}$, 对 $M \times A_a$ 来说, $Prob^{M \times A_a}(\langle s, Z_0 \rangle, \alpha_S^T) = Pr^{M \times A_a}\{\zeta \in Path_\omega^M(\langle s, Z_0 \rangle) \mid M, \zeta \models \alpha_S^T\}$ 显然成立. 要证明 $Prob^M(s, \alpha_S^T) = Prob^{M \times A_a}(\langle s, Z_0 \rangle, \alpha_S^T)$, 根据路径公式的语义, 只需说明在 M 中, 从 s 出发并满足 α 的路径集合与在 $M \times A_a$ 中, 从 $\langle s, Z_0 \rangle$ 出发, 并满足 α 的路径集合相同就可以了. 而由 $R^{M \times A_a}$ 的定义可知, M 中的路径与 $M \times A_a$ 中路径存在一一对应关系, 因此结论显然成立, 命题得证.

证毕.

命题 3 说明要计算 M 中从 s 出发并满足路径公式 α_S^T 的概率, 可通过在 $M \times A_a$ 中计算从 $\langle s, Z_0 \rangle$ 出发并满足 α_S^T 的概率来实现.

命题 4.

$$M \times A_a, \sigma' \models \alpha_S^T \text{ 当且仅当 } M \times A_a, \sigma' \models \Diamond_S^T \text{accept} \quad (4)$$

证明.

① 必要性. 基于 M 中路径 σ 与 $M \times A_a$ 中路径 σ' 之间的一一对应关系及前面的相关结论, 并由状态公式的语义规则的第一条, 可知在 $M \times A_a$ (为 atsFPM 模型, 所以遵从对应的语义解释) 中, $\langle s_n, Z_n \rangle \models \text{accept}$ 当且仅当 $\text{accept} \in L^{M \times A_a}(\langle s_n, Z_n \rangle)$, 又由 $L^{M \times A_a}$ 的定义可知 $\text{accept} \in L^{M \times A_a}(\langle s_n, Z_n \rangle)$ 成立则必须 $Z_n \cap F \neq \emptyset$ 成立, 同时, 由于 $Z_n = \delta^M(Z_0, \sigma)$, 故 $\delta^M(Z_0, \sigma) \cap F \neq \emptyset$ 成立, 因此, 由式(2)可得 $\sigma \in Path^M(\alpha)$, 又由式(1) $Path^M(\alpha) = Path^M(A_a)$ 可得 $\sigma \in Path^M(\alpha) = Path^M(A_a)$.

根据路径公式的语义, $M \times A_a, \sigma' \models \Diamond_S^T \text{accept}$ 成立相当于 $\sigma' \in Path^M(\Diamond_S^T \text{accept})$, 再由 $Path^M(\Diamond_S^T \text{accept})$ 的定义可知 $\langle s_n, Z_n \rangle \models \text{accept}$ 相当于 σ' 的最后一个状态必须满足 accept , 即 $\sigma' \in Path^M(\alpha)$. 故得证.

② 充分性. 与必要性证明过程类似. 证毕.

因此, 设 (s, Z_0) 为 $M \times A_a$ 中初始状态, 对式(4)两边取概率运算可得

$$Prob^{M \times A_a}(\langle s, Z_0 \rangle, \alpha_S^T) = Prob^{M \times A_a}(\langle s, Z_0 \rangle, \Diamond_S^T \text{accept}) \quad (4')$$

由式(1)~(4)、(4')可得结论:

$$Prob^M(s, \alpha_S^T) = Prob^{M \times A_a}(\langle s, Z_0 \rangle, \Diamond_S^T \text{accept}) \quad (5)$$

式(5)表明, M 中由 α 指定的路径范式并满足时间约束 T 和空间约束 S 下的路径集合的概率, 可通过在 $M \times A_a$ 中, 计算从初始状态 $\langle s, Z_0 \rangle$ 出发, 满足 $\Diamond_S^T \text{accept}$ 的路径集合的概率, 其中 A_a 表示由 α 生成的非确定的有限自动机.

下面考虑如何计算 $Prob^{M \times A_a}(\langle s, Z_0 \rangle, \Diamond_S^T \text{accept})$.

由于 $Prob^{M \times A_a}(\langle s, Z_0 \rangle, \Diamond_S^T \text{accept}) = Prob^{M \times A_a}(\langle s, Z_0 \rangle, \text{true } U_S^T \text{accept})$, 根据式(5)可得 $Prob^M(s, \alpha_S^T) = Prob^{M \times A_a}(\langle s, Z_0 \rangle, \text{true } U_S^T \text{accept})$. 为了描述方便, 本文中暂考虑 $T = [0, t]$, $S = [0, r]$ 的情况, 考虑计算 $Prob^{M \times A_a}(\langle s, Z_0 \rangle, \text{true } U_{[0, r]}^{\leq t} \text{accept}) = Prob^{M \times A_a}(\langle s, Z_0 \rangle, \text{true } U_{[0, r]}^{[0, t]} \text{accept})$.

在 $M \times A_a$ 中, 令所有满足 accept 的状态为吸收态, 且它们的空问要求值为 0, 得到的模型为 M^* , 则 $Prob^{M \times A_a}(\langle s, Z_0 \rangle, \text{true } U_{[0, r]}^{[0, t]} \text{accept}) = Prob^{M^*}(\langle s, Z_0 \rangle, \text{true } U_{[0, r]}^{[t, t]} \text{accept})$, 具体证明参见文献[7-8]. 对于 M^* , 考虑一个二维随机过程 $((X_t, Y_t), t \geq 0)$ ^[8], 其中, X_t 的状态空间为 S , 刻画系统的动态行为, Y_t 的状态空间为 $\mathcal{R}_{\geq 0}$, 表示累计空问要求分布, 则 $Prob^{M^*}(\langle s, Z_0 \rangle, \text{true } U_{[0, r]}^{[t, t]} \text{accept})$ 表示, 在模型 M^* 中, 在时刻 t 处于 accept 状态且累计空问要求值小于 r 的概率, 即

$$Prob^{M^*}(\langle s, Z_0 \rangle, \text{true } U_{[0, r]}^{[t, t]} \text{accept}) =$$

$$Pr^{M^*}\{Y_t \leq r, X_t \in \text{Sat}(\text{accept})\} \quad (6)$$

本文采用离散化方法来计算概率, 该方法的思想来源于文献[18]. 首先选择离散步长 Δ , 实数 Δ 必须尽可能小, 以满足在长度为 Δ 的时间区间内, 模型中超过一个转移的概率趋向于 0. 通过 Δ 的离散化后, 可将讨论区间转换为 $T^* = [0, t/\Delta]$, $S^* = [0, r/\Delta]$, 其中 T^*, S^* 均为整数区间. 于是

$$Pr^M\{Y_t \leq r, X_t \in \text{Sat}(\Psi)\} \approx \sum_{s \in \text{Sat}(\Psi)} \sum_{k=1}^R F^{T'}(s, k) \cdot \Delta, \quad R = r/\Delta, T' = t/\Delta \quad (7)$$

其中, M 为系统模型, $F^{T'}(s, k)$ 表示在时刻 T' 处于状态 s 且累加空问消费值为 k 的概率, 即

$$F^1(s, k) = \begin{cases} 1/\Delta, & \text{如果 } (s, k) = (s_0, \rho(s_0)) \\ 0, & \text{其它情况} \end{cases},$$

$$F^{j+1}(s, k) = F^j(s, k - \rho(s)) \cdot (1 - E(s) \cdot \Delta) + \sum_{s' \in S} F^j(s', k - \rho(s')) \cdot R(s', s) \cdot \Delta.$$

6.3 状态空间约简

随着信息系统规模的扩大, 系统的状态数量将呈指数增加而产生状态爆炸问题^[4]. 本文提出的针对 atsFPM 的模型验证, 也是如此. 解决状态爆炸问题, 已有许多方法, 如互模拟、on_the_fly, 抽象法、偏序约简、对称法等^[4]. 我们用偏序约简法对 atsFPM 模型进行处理, 其基本思想是: 在迹理论的基础上建立动作独立关系, 利用这种独立关系和路径扫描迹等价关系建立执行动作序列的等价关系, 这样可以

对状态空间进行有选择的访问,达到状态空间约简的目的.换句话说,偏序约简技术在进行状态遍历时,并不搜索该状态的所有后继状态,仅搜索其充分动作集中动作的后继状态.

由于 atsFPM 模型的路径同时涵盖了功能以及时间和空间约束,因此要确定兼有时间和空间约束的动作独立关系的定义,在此基础上定义路径的扫描迹等价关系.并且,需证明原始模型与约简后的模型相对于待验证的性质来说是等价的,同时说明约简模型的构造与分析要比基于原始模型的模型验证效率要高.我们已经开展了这些方面的研究工作,由于本文版面关系,具体的状态约简方法将在后续论文中阐述.

6.4 模型检测算法

算法 1. atsFPM 模型验证算法.

输入: atsFPM 实例模型,验证规范 Φ

输出: 规范 Φ 的可满足集合

$Sat_set_of_ \Phi \text{ checking_atsFPM } (Model \text{ atsFPM },$
 $Specification \Phi)$

```
{
    sat =  $\emptyset$ ;
    if ( $\Phi \in AP$ )      sat =  $\{s \mid \Phi \in \mathcal{L}(s)\}$ ;
    if ( $\Phi = \Phi_1 \wedge \Phi_2$ )  sat = checking_atfFPM ( $\Phi_1$ )  $\cap$ 
                           checking_atfFPM ( $\Phi_2$ );
    if ( $\Phi = \neg \Phi_1$ )      sat =  $S \setminus \text{checking\_atsFPM } (\Phi_1)$ ;
                           // S 为 atsFPM 模型的状态集
    if ( $\Phi = S_{\approx p}(\Phi_1)$ ) { if ( $\pi^M(s, \text{checking\_atsFPM } (\Phi_1) \models p$ )
                           sat = sat  $\cup \{s\}$ ;
    if ( $\Phi = P_{\approx p}(\alpha_S^T)$ ) {
        构造  $\alpha$  的自动机  $A_\alpha$ ;
        生成模型  $M$  和  $A_\alpha$  的积 atsFPM 模型  $M \times A_\alpha$ ;
        令  $M \times A_\alpha$  中所有满足 accept 的状态为吸收态,
        且空间消耗值为 0, 得模型  $M^*$ ;
        For (all  $s \in S$ ) {
            if ( $Prob^{M^*}(\langle s, Z_0 \rangle, \text{true } U_{[0,r]}^{[t,r]} \text{accept}) \models p$ )
                sat = sat  $\cup \{s\}$ ;
        }
    }
    return sat;
}
```

7 实例分析

本节以第 2 节中的例 1 为实例,参照建立的系统模型(图 1~图 3),给出以下 4 个验证规范,从不

同的角度对系统的安全性进行讨论:

(1) 系统在运行过程中,不会出现死锁;

(2) 系统在 1h 内进入瘫痪状态的概率至多为 0.02% (即可能性很小,是个安全性性质);

(3) 正常工作情况下,系统所需带宽超过最大带宽的概率至多为 0.01;

(4) 正常工作的 2h 内系统的维护成本高于 50 个成本单位的概率至多为 0.02.

验证规范(1)、(2)、(3)分别是功能、功能和时间性能、功能和空间性能上的安全性要求,给出对应的状态或路径公式,通过已有的方法可以很方便地得到解决,下面我们讨论验证规范(4)的处理.很明显,此时的路径范式为

$$\alpha = ((\text{true}, M_{\text{fail}}) \cup (\text{true}, M_{\text{repair}}) \cup (\text{true}, P_{\text{fail}}) \cup (\text{true}, P_{\text{repair}}))^* ; (\text{off}, \checkmark).$$

验证要求为:判断 $P_{\geq 0.98}(\alpha_{[0,50]}^{[0,2]})$ 是否成立,具体做法如下:

(1) 构造 α 对应的自动机,如图 4 所示.

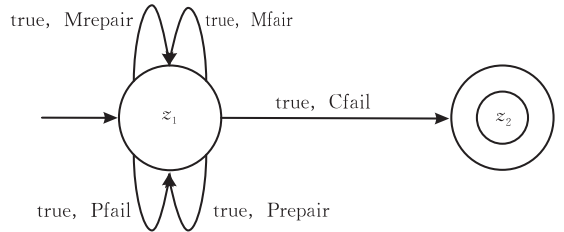


图 4 α 对应的非确定自动机 A_α

其中,初始状态集为 $Z_0 = \{z_1\}$,终止状态集为 $F = \{z_2\}$.

(2) 生成积 atsFPM 模型,如图 5 所示(部分标记省略).其中,状态标记函数为 $L^{M \times A_\alpha}(\langle s_{0,0}, \{z_2\} \rangle) = \mathcal{L}(s_{0,0}) \cup \{\text{accept}\}$, $L^{M \times A_\alpha}(\langle s_{i,j}, \{z_1\} \rangle) = \mathcal{L}(s_{i,j})$ ($1 \leq i, j \leq 3$),阴影状态为吸收态.

(3) 计算可满足集合.

在图 5 中,阴影状态没有输出边,是吸收态.令各个参数的值分别为 $m_f = 0.001$, $m_r = 0.2$, $p_f = 0.01$, $p_r = 0.02$, $s_r = 0.02$, $c_f = 0.003$.根据本文前面的描述,用离散化方法^[18]可得出

$$Prob^M(s_{3,3}, \alpha_{[0,50]}^{[0,2]}) = 0.987065,$$

$$Prob^M(s_{3,2}, \alpha_{[0,50]}^{[0,2]}) = 0.989523,$$

$$Prob^M(s_{3,1}, \alpha_{[0,50]}^{[0,2]}) = 0.989318,$$

$$Prob^M(s_{2,3}, \alpha_{[0,50]}^{[0,2]}) = 0.989901,$$

$$Prob^M(s_{2,2}, \alpha_{[0,50]}^{[0,2]}) = 0.982584,$$

$$Prob^M(s_{2,1}, \alpha_{[0,50]}^{[0,2]}) = 0.989523,$$

$$Prob^M(s_{1,3}, \alpha_{[0,50]}^{[0,2]}) = 0.981318,$$

$$Prob^M(s_{1,2}, \alpha_{[0,50]}^{[0,2]}) = 0.989701,$$

$$Prob^M(s_{1,1}, \alpha_{[0,50]}^{[0,2]}) = 0.987584,$$

$$Prob^M(s_{0,0}, \alpha_{[0,50]}^{[0,2]}) = 0.000000,$$

$$\text{故 } Sat(P_{\geq 0.98}(\alpha_{[0,50]}^{[0,2]})) = \{s_{3,3}, s_{3,2}, s_{3,1}, s_{2,3}, s_{2,2}, s_{2,1}, s_{1,3}, s_{1,2}, s_{1,1}\}.$$

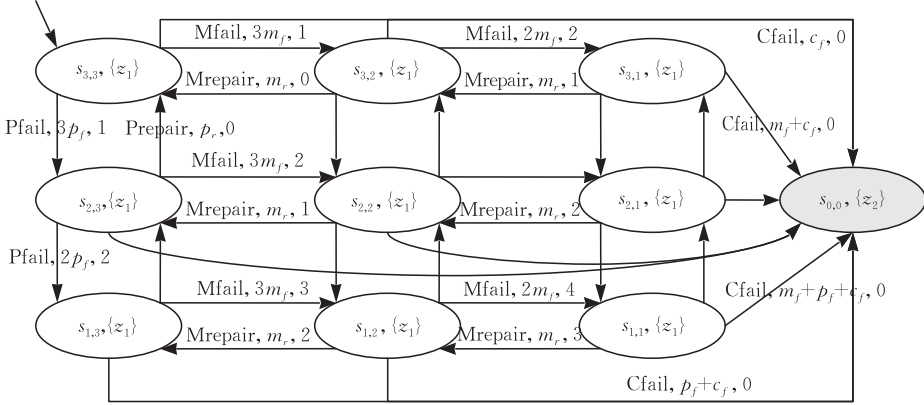


图 5 atsFPM 模型 $M \times A_g$

8 结束语

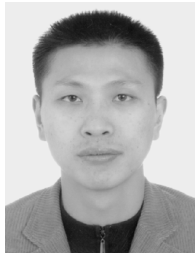
为了评估信息系统的功能正确性、性能可满足性,本文基于已有的功能、性能验证模型,通过给状态定义空间要求量,提出一种基于动作、时间和空间要求的 atsFPM 统一验证模型.在该模型中,空间要求数据表示系统驻留于某状态时,对上下文环境中空间信息资源的再分配率(单位时间内空间信息的生产或消耗量),用以表达系统基于空间资源的性能指标.将系统的功能及描述时间和空间要求的性能指标集中在一个统一的模型中进行表达,并给出模型验证方法,最后通过例子分析了 atsFPM 模型的空间信息刻画能力及组合验证方法的可用性.和以前的工作相比,本文提出的组合验证模型,具有更加丰富的表达能力,克服了传统模型中功能和性能隔离验证且无法与空间信息指标相融合的困难,能进行复杂信息系统的带有空间信息约束的系统指标的刻画和形式化验证,进一步评测信息系统的安全性、可信性.下一步,我们将给出动作上的空间信息要求,进一步完善 atsFPM 模型,并对在验证过程中有可能产生的状态爆炸问题进行深入研究.

参 考 文 献

[1] Klaus H, Natarajan S. Experiments in theorem proving and model checking for protocol verification//Proceedings of the FME'96. Oxford, UK, 1996: 662-681
[2] Basin D, Mödersheim S, Viganò L. OFMC: A symbolic

model checker for security protocols. International Journal of Information Security, 2005, 4(3): 181-208
[3] Meyer J F. Performability evaluation: Where it is and what lies ahead//Proceedings of the International Computer Performance and Dependability Symposium 1995. Erlangen, Germany, 1995: 334-343
[4] Baier C, Katoen J-P. Principles of Model Checking. Massachusetts: The MIT Press, 2008
[5] Baier C, Haverkort B, Hermanns H, Katoen J-P. Model-checking algorithms for continuous time Markov chains. IEEE Transactions on Software Engineering, 2003, 29(6): 524-541
[6] Baier C, Haverkort B, Hermanns H, Katoen J-P. On the logical characterization of performability properties//Proceedings of the ICALP 2000: Automata, Languages and Programming. Geneva, Switzerland, 2000: 780-792
[7] Cloth L, Katoen J-P, Khattri M, Pulunqan R. Model checking Markov reward models with impulse rewards//Proceedings of the DSN 2005. Yokohama, Japan, 2005: 722-731
[8] Haverkort B, Cloth L, Hermanns H, Katoen J-P, Baier C. Model checking performability properties//Proceedings of the DSN 2002. Washington, USA, 2002: 103-112
[9] Al-Jaar R Y, Desrochers A A. Performance evaluation of automated manufacturing systems using generalized stochastic Petri nets. IEEE Transactions on Robotics and Automation, 1990, 6(6): 621-639
[10] Lin Chuang, Wei Ya-Ya. Stochastic process algebras and stochastic petri nets. Journal of Software, 2002, 13(2): 203-213(in Chinese)
(林闯, 魏丫丫. 随机进程代数与随机 Petri 网. 软件学报, 2002, 13(2): 203-213)
[11] Hermanns H, Katoen J-P, Kayser J M, Siegle M. Towards model checking stochastic process algebra//Proceedings of the IFM 2000. Dagstuhl Castle, Germany, 2000: 420-439

- [12] Hermanns H, Herzog U, Katoen J-P. Process algebra for performance evaluation. *Theoretical Computer Science*, 2002, 274(1-2): 43-87
- [13] Hermanns H, Herzog U, Mersiotakis V. Stochastic process algebras-between lotus AND Markov chains. *Computer Networks and ISDN Systems*, 1998, 30(9-10): 901-924
- [14] Kuntz M, Siegle M. A stochastic extension of the logic PDL//*Proceedings of the PMCCS-6*. Illinois, USA, 2003: 58-61
- [15] Meyer-Kayser J. Automatische verifikation stochastischer system [Ph. D. dissertation]. Institut für Informatik, Universität Erlangen-Nürnberg, Germany, 2004
- [16] Baier C, Cloth L, Haverkort B, Kuntz M, Siegle M. Model checking Markov chains with actions and state labels. *IEEE Transactions on Software Engineering*, 2007, 33(4): 209-224
- [17] Cloth L, Haverkort B, Hermanns H, Katoen J-P, Baier C. Model checking pathCSL//*Proceedings of the PMCCS-6*, Illinois, USA, 2003: 19-22
- [18] Tijms H C, Veldman R. A fast algorithm for the transient reward distribution in continuous-time Markov chains. *Operations Research Letters*, 2000, 26(4): 155-158



NIU Jun, born in 1976, Ph. D. candidate, lecturer. His main research interests include trusted software and software verification.

ZENG Guo-Sun, born in 1964, Ph. D., professor, Ph. D. supervisor. His research interests include software verification and information security.

CHEN Bo, born in 1963, Ph. D. candidate, associate professor. His main research interests include trusted software and software verification.

Background

This work is supported by the National Natural Science Foundation of China under grant Nos.90718015 and 60673157 and the National High Technology Research and Development Program (863 Program) of China under grant No. 2007AA01Z425, and the National Basic Research Program (973 Program) of China under grant No. 2007CB316502.

An important object of these projects is to investigate how to analyze and verify the complicated information system such as internet software or embedded system by model checking approaches automatically, and come to a conclusion whether the system is secure, dependable and trustworthy. Our main task is focused on analysing information system's function and performance by some formal models such as labeled transition systems or continuous time Markov chains and some model checking algorithms. The function model is based on labeled transition systems and performance is based on continuous time Markov chains or continuous reward model.

The traditional work on the security analysis of complicated information system's is mainly focused on functional

analysis by LTL, CTL or CTL* model checking approaches. Some approaches by which we can pursuing performance analysis on time or space have been existed. The time properties are described by adding time parameters on transition labels and the spatial properties are described by adding the information which express the constraints of spatial resource such as time, memory and cost. The research group proposed several verification approaches about the web service's security by the analysis of their behavior chains patterns.

With lack of a combinatorial model and verification approaches on function and performance at one time, this paper first introduces a integrated verification model atsFPM, which integrates the continuous time Markov chains and continuous reward model and includes functional, time and spatial constraints, and then proposes a model checking algorithm by extends the continuous stochastic logic and continuous stochastic reward logic. The results shows that the atsFPM and the model checking approaches would be helpful to somebody while evaluating security.