

实时系统时段性质的模型检验

Model Checking on Duration Properties of Real-time Systems

李 勇 李宣东 郑国梁

(南京大学计算机科学与技术系 南京210093)

Abstract Model Checking is a widely used technology to verify the design's correctness automatically. Real-time systems' properties include instant properties and duration properties. It is no doubt that checking real-time systems for the latter is much more difficult than the former. This paper introduces the main work on checking some duration properties of real-time systems for decades with comparison, and discusses the future work at last.

Keywords Model checking, Real-time systems, Duration property, Timed automata

一、前言

随着计算机硬件的飞速发展,计算能力和速度大幅度提高,一方面使得计算机技术应用范围不断扩大,另一方面使得计算机系统的规模和复杂性急剧增加,导致系统出错的几率也随之增大。但任何一个错误在某些系统,如:空中运输控制系统、银行财务系统等,往往可能导致重大的经济损失,甚至人员伤亡,如1996年阿波罗5号火箭的爆炸等。显然对这种高安全性系统的正确性要求是极其严格的,在高安全性系统的开发过程中,最为关键的问题是如何在尽可能早的阶段验证设计的正确性。应用计算机系统的最大障碍就是系统的正确性无法保证,因此提高系统的正确性与可靠性变得日益迫切。目前随着因特网和嵌入式系统的广泛应用,我们的生活更加依赖于越来越多的具有计算功能的设备,发现和使用更多有效的确认方法来提高系统的正确性也就显得更重要。

目前主要使用的确认方法有:仿真(Simulation)、测试(Testing)、推导验证(Deductive Verification)和模型检验(Model Checking)等。仿真和测试都是在系统实际应用前为保证系统正确性进行试验,使用的典型方法都是通过某些选定的点输入准备的信号,然后在对应的观察点观察结果信号。用这种方法可以发现许多错误,但对于复杂的大型系统,覆盖所有的可能的情况以保证系统绝对的正确性几乎是不可能的。仿真和测试验证了系统部分可能行为的正确性,遗留下对未检测行为正确性的不可知,而对所有可能行为正确性的验证则需使用形式化方法(Formal Methods),特别是形式化验证(Formal Verification)。

形式化方法是以数学为基础用于对系统进行规格说明(Specification)和验证(Verification)的语言、技术和工具等的总称^[1]。没有系统可以被构造或证明为百分之百安全的,这是一个基本原理^[2]。形式化方法不是灵丹妙药,也不例外。因为系统并不是孤立的,而是在一定的环境中运行的,规格说明总是对运行环境做了一些假设,正确性的证明只在这些假设下有效;即使可以很完全地描述环境(这实际是不可行的),在此基础上开发的系统也不会和最初的设计相符。但形式化方法可以用来划定系统的边界(系统和环境的接口);精确地刻画系统的行为;精确地定义系统所需的性质;证明系统和它的规格说明相吻合,从而提高人们对系统安全性的信任度。规格说明是对系统及其特性进行描述的过程^[3],需强调的是任何正确性的证明和系统及其特性这两者都密切相关。规格说明

的过程是对系统进行精确描述的行为,它能获得对所描述的系统更深的理解,同时规格说明也是需求者和设计者、设计者和实现者、实现者和测试者之间极其有用的沟通工具。形式化验证的方法主要有推导验证和模型检验两种。推导验证,也称定理证明(Theorem Proving),是采用基于数学的逻辑语言,用公式的形式描述系统及其特性,以公理和推理规则为依据,推导出系统特性的一种证明过程。推导验证最初应用在严格系统的正确性证明上,但即使是验证一个简单的信号协议或电路,也需要具有相当经验的逻辑推理专家们花费大量的时间。对于复杂的大型系统,这样的资源是很难得到的,故很难在实际中应用,除非对系统的正确性要求非常高,可以不考虑人力和物力的投入。当然目前推导验证已由最初的纯手工发展到部分由计算机辅助证明(需要人机交互),且推导验证的一大优点也在于它可以应用在无限状态系统上。模型检验是一种自动检验有限状态系统的技术^[3],一般过程是对系统状态空间的穷尽搜索来判定系统是否满足某个给定的规格说明。和其他确认方法相比,模型检验具有以下两个主要优点:

- 模型检验能达到完全自动化的程度,不需要用户的交互或在数学、逻辑等方面的指导;
- 模型检验过程总会以“是”或“否”的结果中止,当以“否”的结果中止时,说明设计或系统不满足某个给定的性质。更为有价值的是,此时一个违反性质的行为反例将会被给出,此反例将对理解错误的真正原因和修正错误提供线索。

应用模型检验的最大问题是状态空间爆炸问题,尤其在系统是由多个组件构成时。近几十年来,这个问题取得了可喜的进展。因此,模型检验被工业界广泛应用在通讯协议和电路的检验上。虽然只能检验有限状态系统是其中一个主要的限制,但在大量的这类系统中仍是可应用的;而且通过与抽象方法和归纳规则应用的结合,模型检验对部分无限状态系统也是适用的。应用模型检验包含以下三个步骤:

1. 建模:建模是对一个设计进行形式化,使其能被模型检验工具接受,这是第一步;
2. 规格说明:在检验前,还需使用逻辑语言对所需检验的性质进行形式化,如所检验的性质和时间有关,则一般使用时序逻辑(Temporal Logic);
3. 检验:检验工具根据所给的形式化模型和性质自动进行检验,中止时给出结果,或是模型满足性质,或是模型不满足性质并给出反例。

二、实时系统的时段性质

2.1 实时系统

实时系统(Real-time Systems)是指那些严格要求对外部的输入需及时做出回应的系统^[4],如飞机、机器人的控制器等。显而易见,在一些实时系统中的错误是极其危险的,因此检验实时系统的正确性非常重要,但又是极不容易的。在模型中,时间可以看成离散的或是连续的。如果时间被看成离散的,则所有可能的时钟值只能为非负整数,而且所有的事件只能发生在整数时刻,这种模型非常适合同步系统,因为所有系统组件都依据一个全局时钟进行同步。多年来,这种系统被成功应用在检验同步硬件设计的正确性上。而连续的时间,则是异步系统的合适模型。近年来,许多不同的关于连续时间的模型被提出,现在 Alur 等人提出的时间自动机(Timed Automata)^[5]成为了标准。时间自动机是在有穷状态自动机的基础上扩充了一时钟变量集合 X 得到的。一个在 X 上的时钟约束 ϕ 是如下定义的:

$$\phi = x \geq a \mid x \leq a \mid \phi_1 \wedge \phi_2 \quad (x \in X, a \in \{0\} \cup \mathbb{N})$$

时间自动机是一个五元组 $\langle L, s_I, \Sigma, X, E \rangle$, 其中: L 为一有限的位置集合; $s_I \in L$ 为起始位置; Σ 为一有限的标签集合; X 为一有限的时钟集合; $E \subseteq L \times \Sigma \times \Phi(X) \times 2^X \times L$ 为一有限的转换集合, $e = \langle s, a, \phi, \lambda, s' \rangle \in E$ 为一以 a 为标签从源位置 s 到目标位置 s' 的转换, ϕ 为一在 X 上的时钟约束表明此转换何时可以发生,集合 $\lambda \subseteq X$ 则给出了哪些时钟变量在此转换执行后被重置为 0。

在此基础上,根据应用的需要,还可增加终止位置、接收位置等组件对时间自动机进行扩充。时间自动机的行为可以用带时间戳的转换序列来表示,如 $\sigma = (e_1, 5) \wedge (e_2, 3)$ 表示该系统从起始位置开始,经过 5 个时间单位后发生转换 e_1 ,进入下一个位置,又经过 3 个时间单位后发生转换 e_2 ,再进入下一个位置。 $\tau(\zeta)$ 表示某行为 ζ 总共花费的时间单位,显然 $\tau(\sigma) = 8$ 。

2.2 时序逻辑

时序逻辑(Temporal Logic)是一种特殊的模态逻辑(Modal Logic),它提供了一个形式化系统,用来定性地描述和推导断言的真伪值如何随时间的变化^[6]。典型的时序逻辑都具有有时(sometimes,表明某断言 P 在将来某个时刻为真)和总是(always,表明某断言 P 在将来所有时刻都为真)算子。时序逻辑描述了事件的时间顺序关系,但并没有明确地引入时间,它已被证明是描述并发系统,实时系统的强有力工具。目前大多数实时系统的模型检验工作都使用时序逻辑来描述需检验的性质。

2.3 时段性质

近二十年来,许多新的模型检验技术被提出用来检验实时系统(时间自动机),在此基础上一些工具被开发出来,在实践中被应用。其中大多数的方法都是检验实时系统在某一时间点的性质,我们称之为实时系统的即时性质(Instant Property)。这些方法都是基于时间自动机区域图(Regional Graph)的搜索算法,故和时间自动机的可达性问题(Reachability,即时间自动机的某一给定位置是否能被系统某一行达到)具有同样的复杂度。

在实际应用中,我们同样必须考虑实时系统在某一时间区间上的性质,我们称之为时段性质(Duration Property)。描述时段性质,需要计算系统在不同位置上花费的累计时间,标

准的时序逻辑只具有有限的表达能力,不能描述有关时间长度的性质,故不能描述时段性质。时段演算(Duration Calculus)^[7]作为一种扩展的区间时序逻辑,非常适合描述和推导时段性质。在时段演算中,状态是从实数(代表连续的时间)到 $\{0, 1\}$ 的映射,并引入了算子“ \int ”用来计算系统在某一状态上的累计时间,而并不需考虑每次的确切时间,如 $\int S$ 代表了在 $[b, e]$ 区间上处在状态 S 的累计时间,而 $\int l$ 则代表了系统观察时间的长度,等于 $e - b$ 。在煤气炉(Gas Burner)例中,有漏气(Leak)和不漏气(Nonleak)两种状态,其中漏气状态是危险状态,设计者要求其不能过长,需要满足在任何不小于 60 秒的观察时间内,漏气状态的累计时间不能超过观察时间的二十分之一,用时段演算可将此需求表示为:

$$\text{Req: } \int l \geq 60 \Rightarrow 20 \int \text{Leak} \leq \int l$$

为满足此需求,可以设计煤气炉满足以下两个设计要求:

$$\text{① Des-1: } \Box(\lceil \text{Leak} \rceil \Rightarrow \int l \leq 1)$$

$$\text{② Des-2: } \Box(\lceil \text{Leak} \rceil \wedge \lceil \text{Nonleak} \rceil \wedge \lceil \text{Leak} \rceil \Rightarrow \int l \geq 30)$$

其中“ \Box ”是时段演算中的“总是”算子,“ $\lceil P \rceil$ ”代表状态 P 在观察时间内基本上(除了在一些离散时间点上)都为 1,“ $\lceil P \rceil \wedge \lceil Q \rceil$ ”代表在观察时间内存在一个时间点 m ,在 m 之前状态 P 基本上为 1,在 m 之后状态 Q 基本上为 1。在这里我们只直观地解释了时段演算中某些算子的含义,使读者可以理解本文中相关时段演算公式的含义,其形式化定义请参阅文[7]。故上两个设计要求实际含义为:

• Des-1: 任何时候漏气状态不能持续超过 1 秒;

• Des-2: 任何时候两次漏气之间时间间隔不能少于 30 秒。

应用时段演算,可以验证需求在此两个条件下满足^[7]。同时,我们可以使用模型检验的方法进行检验。根据此两个设计要求,我们可以构造出煤气炉的时间自动机(如图 1)。

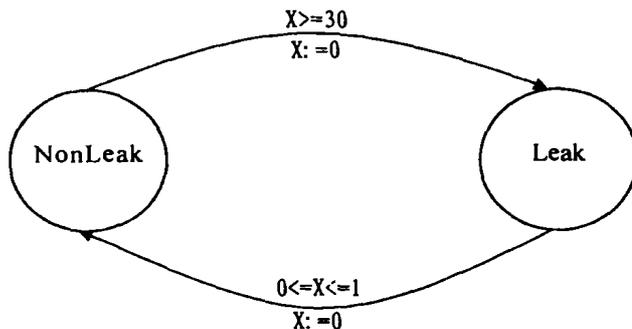


图 1 煤气炉的时间自动机模型

根据此模型,通过相应的模型检验方法可以对其需求进行检验。比较这两种方法,我们可以发现模型检验的方法不需要专家级的逻辑推导,能更有效地在实际中进行应用。

三、实时系统时段性质的模型检验方法

近十几年来,对实时系统的模型检验工作已成为一个重要的研究方向,并取得了巨大的成功。但多数的工作都是检验实时系统的即时性质,这主要是由于时段性质的检验远比即时性质的检验复杂得多,以下将对现有的这方面工作做一介绍。

3.1 有界时段可达性的检验

有界时段可达性(Duration-Bounded Reachability)^[8]是最早被考虑的时段性质,其定义为:当一给定的时间自动机从某一起始位置运行,并结束于某一终止位置时,所得到的行为其时间长度在一给定的时间区间内。在文[8]中,没有给出有界时段可达性的形式化定义,作者给出了一个基于时间自动机区域图的算法,用来检验有界时段可达性的问题,这个算法的复杂度是 PSPACE 的。

实际上,有界时段可达性问题是传统的可达性问题的扩充,即不仅要求某一终止位置可达,并且要求其在规定的时间内可达。它是一类较简单的时段性质,它的观察时间是系统一次完整的有限行为,且要求计算的仅仅是在所有位置上的累计时间(即各位置的加权系数都为1)。

3.2 线性时段不变式的检验

线性时段不变式(Linear Duration Invariants)^[9]是一种时段演算的公式,其形式如下:

$t \leq \int l \leq T \Rightarrow \sum_{s \in L} c_s \int s \leq M$ (其中 $t, T, M, c_s (s \in L)$ 为实数, T 可为 ∞ , s 为状态)

线性时段不变式的含义为,在任何一个时间长度在区间 $[t, T]$ 内的观察期内,系统在各个位置所停留的累计时间总是满足某一线性不等式。在煤气炉例中,需求 Req 就为线性时段不变式,因为 $\int l = \int Leak + \int Nonleak$, 故 Req 为 $\int l \geq 60 \Rightarrow 19 \int Leak - \int Nonleak \leq 0$ 。而且煤气炉所对应的时间自动机是一类特殊的时间自动机,在每次转换完成时总是重置所有的时钟变量,此类时间自动机被称之为实时自动机(Real-time Automata)^[9]。在文[9]中,周巢尘等人提出了一个基于线性规划(Linear Programming)的算法,对实时自动机的线性时段不变式进行了检验。由于线性规划是个 P 问题,依据此算法,可以十分有效地解决实时自动机的线性时段不变式的检验问题。在文[10]中,Kesten 和 Pnueli 等人应用积分图(Integration Graph)的方法检验了时间自动机的线性时段不变式。这个算法同时说明检验时间自动机的线性时段不变式是个 NP 问题,从此处也可以看出检验实时系统的时段性质的复杂度非常高。这个算法虽然解决了对时间自动机全集的检验,但由于算法复杂度无法在实际中进行应用。此后对线性时段不变式的检验工作集中在寻找可有效检验的时间自动机子集上。在文[11]中,李宣东等人应用线性规划的方法,对可以用时间正则表达式(Timed Regular Expression)表示其行为的时间自动机的线性时段不变式进行了检验。时间正则表达式 R 和其表示的时间自动机行为 $L(R)$ 的定义如下:

- ϵ 是时间正则表达式,且 $L(\epsilon) = \{\epsilon\}$;
- 如果 e 是一个转换,则 e 是时间正则表达式,且 $L(e) = \{(e, t) | t \geq 0\}$ 且可为 ∞ ;
- 如果 R_1 和 R_2 是时间正则表达式,则 $R_1 \wedge R_2$ 是时间正则表达式,且 $L(R_1 \wedge R_2) = \{\sigma_1 \wedge \sigma_2 | \sigma_1 \in L(R_1) \wedge \sigma_2 \in L(R_2)\}$;
- 如果 R_1 和 R_2 是时间正则表达式,则 $R_1 \oplus R_2$ 是时间正则表达式,且 $L(R_1 \oplus R_2) = \{\sigma | \sigma \in L(R_1) \vee \sigma \in L(R_2)\}$;
- 如果 R 是时间正则表达式,则 R^* 是时间正则表达式,且 $L(R^*) = \{\sigma_1 \wedge \sigma_2 \wedge \dots \wedge \sigma_m | m \geq 0 \wedge \sigma_i \in L(R) (0 \leq i \leq m)\}$ (当 $m=0$ 时, $\sigma_1 \wedge \sigma_2 \wedge \dots \wedge \sigma_m = \epsilon$);
- 如果 R 是时间正则表达式,则 $(R, [a, b])$ 是时间正则表达式,且 $L((R, [a, b])) = \{\sigma | \sigma \in L(R) \wedge a \leq r(\sigma) \leq b\} (a \geq 0, b$

≥ 0 且可为 ∞)。

稍后,这个方法又被推广到混成自动机(Hybrid System)中的一个子集^[12]。最后在文[13]中,我们应用此方法检验了正闭合环路时间自动机(Positive Loop-closed Timed Automata)的线性时段不变式,这是应用线性规划可检验线性时段不变式的时间自动机的最大子集。

3.3 线性时段性质的检验

检验线性时段不变式的复杂度很高的一个主要原因是由于它具有一个对观察时间长度约束的前提条件,如果把这个前提条件去掉,就得到了线性时段性质(Linear Duration Property)^[14]:

$$\sum_{s \in L} c_s \int s \leq a \text{ (其中 } a, c_s (s \in L) \text{ 为实数, } s \text{ 为状态)}$$

在文[14]中,赵建华等人首先证明了线性时段性质是一个可离散化(Discretisable)的性质,即时间自动机的所有行为满足该性质当且仅当时间自动机的所有整数行为满足该性质,这就大大缩小了所需搜索的空间,然后利用整数空间搜索方法和多项优化手段对时间自动机的线性时段性质进行检验。线性时段性质虽然舍弃了线性时段不变式的前提条件,但仍具有很强的表达能力,如在煤气炉例中需检验的需求 Req 就可以通过对煤气炉所对应的时间自动机进行修改后转换成线性时段性质,从而进行检验。

3.4 时序时段性质的检验

我们在前两节工作的基础上,开发完成了相应的模型检验工具,并成功检验了一些实时系统的时段性质。但在实际中,仍发现我们的工具对某些系统的检验无法得到满意的结果。仔细分析了以上算法后,我们发现线性规划可能包含数目巨大的变量和状态空间爆炸是主要问题。使用线性规划可以减少空间搜索的代价,但必须限制线性规划的变量数。同时我们考察了煤气炉例子中的设计条件,发现了一种新的时段性质——时序时段性质(Temporal Duration Property)^[15]:

$$\square (\lceil S_1 \rceil \wedge \lceil S_2 \rceil \wedge \dots \wedge \lceil S_k \rceil \Rightarrow \sum_{s \in L} c_s \int s \leq Q) \text{ (其中 } Q, c_s (s \in L) \text{ 为实数, } s (1 \leq i \leq k) \text{ 为状态)}$$

时序时段性质的含义是,如果系统运行中经过的位置路径和前提条件所吻合,则系统在各个位置所停留的累计时间总是满足某一线性不等式。在实际应用中,对某个实时系统而言,某些路径可能是“危险”的,系统在每个位置上的累积停留时间需满足给定的线性不等式。因为 $\int l = \int Leak + \int Nonleak$, 所以煤气炉例中的设计要求 $Des-1$ 和 $Des-2$ 就是时序时段性质,即 $\square (\lceil Leak \rceil \Rightarrow \int Leak \leq 1)$ 和 $\square (\lceil Leak \rceil \wedge \lceil Nonleak \rceil \wedge \lceil Leak \rceil \Rightarrow - \int Leak - \int Nonleak \leq -30)$ 。

我们首先在文[15]中,对时间正则表达式的时序时段性质应用线性规划技术进行了检验,得到了满意的结果。在文[16]中,我们又对时间自动机的时序时段性质进行了检验。在这个方法中,我们首先证明了时序时段性质也是一个可离散化的性质,然后基于整数的区域图应用线性规划技术对该性质进行了检验,并分析了复杂度。由于该算法所得到的线性规划的变量数和前提条件中的位置数相同,为一定值,故其复杂度和时间自动机的可达性问题具有相同的复杂度,而且采用整数化技术使得需搜索的状态空间大大减少。我们考察了目前成功使用的模型检验工具^[17,18],发现大多都是检验时间自动机的即时性质,而且以可达性问题为基础,说明时序时段性质是一个在实际中可以高效检验的性质,目前我们正在完成

(下转封四)

(上接第 167 页)

该检验工具的开发,将其应用到实际中。

时序时段性质的表达力虽然不如线性时段不变式,但此类性质常出现在实时系统的设计阶段,而线性时段不变式常出现实时系统的规格说明阶段,它们互补地进行使用。

结束语 对实时系统的模型检验工作近十年来已成为一个研究的热点,其中对时段性质进行检验是十分重要的,但有时非常复杂的。和即时性质相比,我们必须承认和接受时段性质检验工作的复杂度。为了有效地进行检验,可以采用以下两个方法:①限制需检验的模型,如只检验时间自动机的某个子集;②限制需检验的时段性质,如对线性时段不变式的前提条件进行修改。

所采用的方法应尽量和时间自动机的可达性问题具有同样的复杂度,这样实现的工具才能如目前广泛应用的可达性的检验工具一样具有高效率。另外,如何发现更多的实例进行时段性质的检验也是一个重要的研究课题。

参考文献

- Clarke E M, Wing J M. Formal Methods: State of the Art and Future Directions
- Wing J M. A Symbiotic Relationship Between Formal Methods and Security
- Clarke E M, O Grumberg Jr., Peled D A. Model Checking. The MIT Press, 1999
- Krishna C M, Shin K G. Real-time Systems. The McGraw-Hill Companies, Inc., 1997
- Alur R, Dill D L. A Theory of Timed Automata. Theoretical Computer Science, 1994. 183~235
- Emerson E A. Temporal and Modal Logic. Handbook of Theoretical Computer Science, Elsevier Science Publishers B. V., 1990. 997~1071
- ZHOU Chaochen, Hoare C A R, Ravn A P. A calculus of durations. Information Processing Letters, 1991, 40(5): 269~276
- Alur R, Courcoubetis C, Henzinger T A. Computing Accumulated Delays in Real-time Systems. In: Proc. of the Fifth Intl. Conf. on Computer-Aided Verification (CAV 93), Lecture on Computer Science 818, Springer-Verlag, 1993. 181~193
- ZHOU Chaochen, ZHANG Jingzhong, YANG Lu, LI Xiaoshan. Linear Duration Invariants. [Research Report 11]. P. O. Box 3058, Macau, July 1993. Published in Formal Techniques in Real-Time and Fault-Tolerant systems, LNCS 863, 1994
- Kesten Y, Pnueli A, Sifakis J, Yovine S. Integration Graphs: A Class of Decidable Hybrid Systems. Hybrid Systems, volume 736 of Lecture Notes in Computer Science, Springer-Verlag, 1994. 179~208
- LI Xuandong, Dang Van Hung. Checking Linear Duration Invariants by Linear Programming. [Research Report 70]. UNU/IIST, P. O. Box 3058, Macau, May 1996. Published in Joxan Jaffar and Roland H. C. Yap (Eds.), Concurrency and Parallelism, Programming, Networking, and Security LNCS 1179, Springer-Verlag, Dec. 1996. 321~331
- LI Xuandong, Dang Van Hung, Zheng Tao. Checking Hybrid Automata for Linear Duration Invariants. [Research Report 109]. UNU/IIST, P. O. Box 3058, Macau, June 1997. Published in R. K. Shamasundar, K. Ueda (Eds.), Advances in Computing Science, Lecture Notes in Computer Science 1345, Springer-Verlag, 1997. 166~180
- LI Xuandong, et al. Efficient Verification of a Class of Linear Hybrid Automata Using Linear Programming. In: Tiziana Margaria, Tom Melham, Eds. Correct Hardware Design and Verification Methods, Lecture Notes in Computer Science 2144, Springer, 2001. 465~479
- Zhao Jianhua, Dang Van Hung. Checking Timed Automata for Some Discretisable Duration Properties. [Technical Report 145]. UNU/IIST, P. O. Box 3058, Macau, Aug. 1998. Journal of Computer Science and Technology, 2000, 15(5): 423~429
- LI Yong, Dang Van Hung. Checking History Properties of Real-time Systems. [Technical Report 214]. UNU/IIST, P. O. Box 3058, Macau, Oct. 2000
- LI Yong, Dang Van Hung. Checking Temporal Duration Properties of Timed Automata. [Technical Report 247]. UNU/IIST, P. O. Box 3058, Macau, Nov. 2001. Accepted to be published by Journal of Computer Science and Technology
- Henzinger T A, Ho P H, Toi H W. A Users Guide to HyTech. [Technical report]. Department of Computer Science, Cornell University, 1995
- Larsen K G, Huttel H. UPPAAL-An Automatic Tool for Verification of Real Time and Hybrid Systems. In: Proc. of the 16th IEEE Real-Time Systems Symposium, Dec. 1997. 76~87

计算机科学

(1974年1月创刊)

第29卷第11期(月刊)

2002年11月25日出版

中国标准刊号: ISSN 1002-137X
CN50-1075/TP

定价: 18.00元 国外定价: 5美元

邮发代号: 78-68

发行范围: 国内外公开

主管单位: 国家科学技术部

主办单位: 国家科技部西南信息中心

编辑出版: 《计算机科学》杂志社

重庆市渝中区胜利路132号 邮政编码: 400013

电话: (023) 63500828 E-mail: jsjcx@swic.ac.cn

社长: 牟炳林

主编: 朱宗元

印刷者: 重庆科情印务有限公司

总发行处: 重庆市邮政局

订购处: 全国各地邮政局

国外总发行: 中国国际图书贸易总公司(北京399信箱)

国外代号: 6210-MO