UML模型中并发对象的组合验证*)

董 威[1,2] 王 戟[1] 齐治昌[1]

(国防科技大学计算机学院 长沙410073)[1] (武汉大学软件工程国家重点实验室 武汉430072)[2]

摘 要 本文在用层次自动机结构化表示UML Statecharts的基础上,定义了UML协同图中并发对象的同步合成,然后根据结构间的模拟关系,研究了对并发对象系统进行组合验证的方法和规则,使有可能在对UML协同图进行模型检验的过程中不必建立系统的全局状态图,以缓解状态爆炸问题。

关键词 UML,模型检验,组合验证

Compositional Verification of Concurrent Objects in UML Models

DONG Wei[1,2] WANG Ji[1] QI Zhi-Chang[1]

(School of Computer, National University of Defense Technology, Changsha410073)[1]

(State Key Laboratory of Software Engineering, Wuhan University, Wuhan430072)[2]

Abstract Based on expressing UML Statecharts with hierarchical automata structurally, the paper defines the synchronous composition of concurrent objects in UML collaboration diagram. Then, the method and rules of compositional verification for concurrent object systems are studied based on the simulation relation between structures, which makes it possible that the global state graph is not needed in model checking and state explosion problem will be reduced.

Keywords UML, Model checking, Compositional verification

1 引言

统一建模语言(Unified Modeling Language,UML)[1]在关键领域软件开发中(例如航空航天、通信、电子商务等)的应用越来越广泛,对UML模型进行正确性验证以判断设计规约是否满足目标需求也成为一个关键问题。模型检验(model checking)是一种重要的自动验证技术,主要通过显式状态搜索或隐式不动点计算来验证有穷状态并发系统的模态/命题性质,并能在系统不满足性质时提供反例路径。近来已出现了一些对UML模型进行模型检验的研究,如对UML Statecharts进行验证[2~4]以及对包含多个并发对象的UML协同图进行模型检验[2,4~6]。

在用UML为并发对象系统建模时,由于刻画单个对象行为的Statecharts本身具有并发和层次等特征,因此验证时面临空间爆炸问题;而当协同图中存在多个并发对象时,更会使整个系统的状态空间迅速增长。这些问题在大规模软件系统的设计规约经过递增和精化后更为突出(如飞行控制系统[7])。

在并发系统中,全局行为的性质经常可以分解为系统所含小部件行为的性质。于是希望只使用系统的一部分来检验每个局部性质,然后根据推理规则得到全局性质,这是组合验证的基本思想[8]。本文研究了如何针对UML行为模型的语义和特点来对并发对象进行组合验证。首先用扩展层次自动机结构化地表示UML Statecharts,并定义其操作语义的LTS。然后针对Statecharts特点定义了同步系统的UML协同图中并发对象合成方法,以得到验证所需的系统整体行为。根据这种合成关系以及结构间的模拟关系,研究了对并发对象系统进行组合验证的方法和规则,有可能在验证UML协同图的过程中不必建立全局状态图。

本文第2节简要描述如何用EHA结构化地表示UML Statecharts及其语义;第3节针对语义提出对UML协同图中并发对象进行同步合成的方法;第4节提出对多个对象进行并发组合验证的方法;最后给出了相关工作和结论。

2 从UML Statecharts到EHA

UML Statecharts刻画了对象在其生命周期中的行为和状态变迁,对其进行模型检验需要一种结构化的操作语义。层次自动机可以被看作Statecharts的抽象语法,抽取掉了语法细节,而只保留Statecharts的关键部分,并以一种结构化的方式表示出来。已有许多研究利用EHA作为UML Statecharts模型检验过程中的一种结构化表示[3,4]。

定义1(顺序自动机) 顺序自动机A是一个4元组$(\sigma_A, s_A^0, \lambda_A, \delta_A)$,$\sigma_A$是有穷的状态集合,$s_A^0$是初始状态,$\lambda_A$是有穷的迁移标记集合,$\delta_A \subseteq \sigma_A \times \lambda_A \times \sigma_A$是迁移关系。

定义2(扩展层次自动机,EHA) EHA H是一个5元组(F, E, ρ, A_0, V),F是一个有穷的顺序自动机集合,$\forall A_1, A_2 \in F, \sigma_{A1} \cap \sigma_{A2} = \varnothing$;$E$是有穷的事件集合;$V$是变量集合;$\rho: \bigcup_{A\in F}\sigma_A \to 2^F$是精化函数,给出$F$的一个树型结构满足:1)存在一个唯一的根自动机$A_0 \in F$,使得不存在$s \in \bigcup_{A\in F}\sigma_A, A_0 \in \rho(s)$;2)每个非根自动机恰有一个父状态:对于$\forall A \in F \setminus \{A_0\}, \exists\, {}_1s \in \bigcup_{A'\in F\setminus\{A\}}\sigma_{A'}, A \in \rho(s)$;3)不存在环路:$\forall S \subseteq \bigcup_{A\in F}\sigma_A, \exists\, s \in S, S \cap (\bigcup_{A\in\rho(s)}\sigma_A) = \varnothing$。

例如,图1中关于一个TV控制器的UML Statecharts,其对应的EHA如图2所示。层次自动机的全局状态由格局来表示,它由包含的顺序自动机的某些局部状态组成。

*)本文受到国家自然科学基金(No. 60303013)和武汉大学软件工程国家重点实验室开放基金(No. SKLSE03-08)资助。董 威 博士,讲师,主要研究方向为软件测试与验证;王 戟 博士,教授,主要研究方向为高可信软件技术;齐治昌 教授,主要研究方向为软件工程。

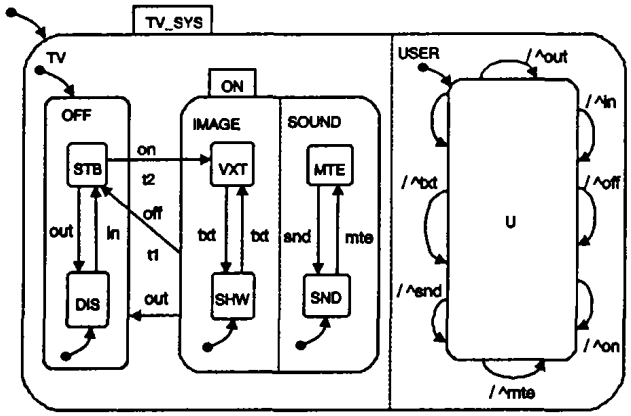

图1 TV 控制器的 Statecharts 模型

定义3(格局) H 的一个格局是一个集合 $Conf \subseteq S(A_0)$ 使得:1)$\exists\ _1 s \in \sigma_{A_0}$ 满足 $s \in Conf$;2)$\forall\ s, A$,如果 $s \in Conf$ 且 $A \in \rho(s)$,那么 $\exists\ _1 s' \in A, s' \in C$。

EHA 的操作语义可以用一个 LTS 定义,它是通过迁移相关联的一组状态。在 Statecharts 中,操作语义中的状态被称为状况(status),每个状况由格局和当前的环境组成[4]。

定义4(EHA 的操作语义) EHA H 的操作语义是一个 LTS $Ts=(S, S_0, L, R)$,其中 S 是状况集合,S_0 是初始状况集,$L: S \to 2^{AP}$ 为每个状况标注一组原子命题,$R \subseteq S \times S$ 是迁移关系。

文[4]中给出把 UML Statecharts 转换为 EHA 的具体方法,并基于开放系统给出关系 R 的一组演绎规则,指出它能正确描述 UML Statecharts 的执行语义。

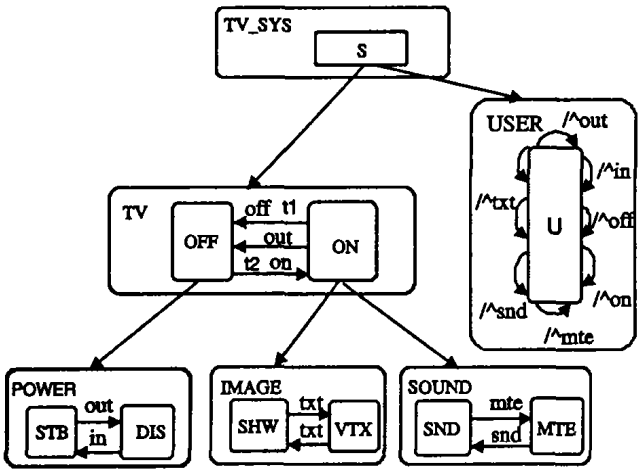

图2 TV 控制器对应的 EHA

3 UML 协同图中并发对象的合成

UML 协同图描述相互协作的并发对象间的动态交互关系和联接关系[1],对象之间通过发送事件进行通信和同步。协同图只能从整体上反映整个系统和各对象之间的关系,而不能以一种严格的方式从对象的行为推导出整个系统的行为,而这是模型检验所需要的。同步概念已被广泛应用在实时系统、电路和嵌入式系统的开发中,我们下面以一种同步的方式对各个对象 Statecharts 描述的行为进行合成,从而得到整个系统的行为。

对一个 EHA H,给其操作语义的定义中增加一个原子命题集 AP 和一个接收集 F,即其操作语义定义为 $K(H)=(S, S_0, AP, L, R, F)$。在同步系统模型中,要求在每个格局时,对任何输入都应存在一个 RTC 步可以执行。例如图3是4阶段握手协议的一个简单电路所对应的 UML Statecharts,图4给出该电路的一个可能的环境[9]。4阶段握手协议电路通过设置 r 的值来向环境发出请求,环境通过事件 a 来响应请求。该系统的协同图如图5所示。

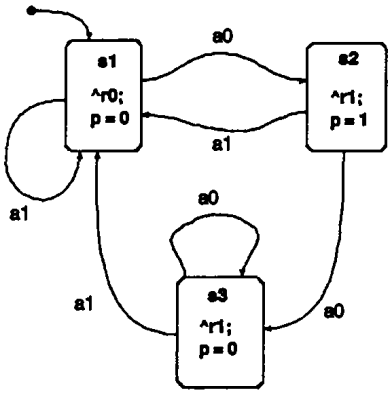

图3 4阶段握手协议电路的状态图

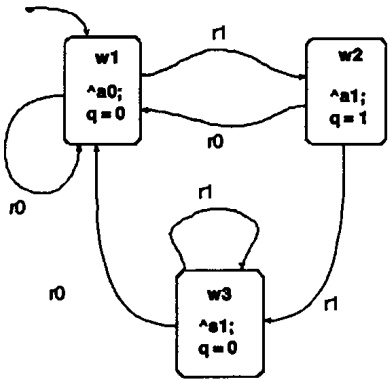

图4 一个环境电路的状态图

令 $s.E$ 表示状况 s 的有效事件集合。由于不考虑延迟事件,因此到达某个状况过程中动作产生的事件只能在该状况中有效。为了方便定义同步合成,把关系 R 进行以下改动得到迁移关系 R':对每一个迁移 $(s,t) \in R$,若该迁移对应的是原 EHA 中受外部事件 e 激发产生的 RTC 步,则 $(s,\{e\},t) \in R'$;若该迁移是受该对象内部事件激发的,则 $(s,\varnothing,t) \in R'$。下文 EHA 语义中的关系 R 均指经过上述修改后的关系。

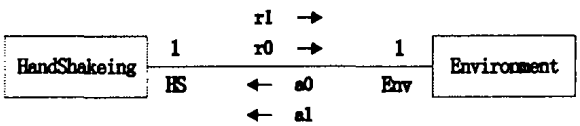

图5 4阶段握手协议电路系统的协同图

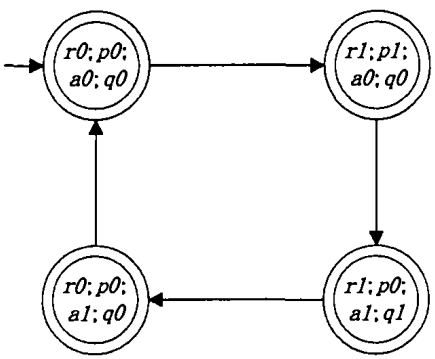

图6 同步合成后的系统状态图

定义5 同步合成(Synchronous Composition) $M=(S,S_0,AP,L,R,F')$和$M'=(S',S'_0,AP',L',R',F')$是两个EHA的LTS.$M$和$M'$的同步合成,记为$M''=M\|M'$,如下定义的一个结构$(S'',S''_0,AP'',L'',R'',F'')$:

1)$S''=\{(s,s')\mid L(s)\cap AP'=L'(s')\cap AP\}$;

2)$S''_0=(S_0\times S'_0)\cap S''$;

3)$AP''=AP\cup AP'$,$(s,s').E=s.E\cup s'.E$;

4)$L''((s,s'))=L(s)\cup L'(s')$;

5)$((s,s'),v'',(t,t')\in R''$当且仅当$(s,v,t)\in R$且$(s',v',t')\in R'$,$v''=v\cup v'$,$(t,t').E=t.E\cup t'.E$,并且满足:i)对任意$e\in v$,如果$e$是$M'$的输出事件,则$e\in s'.E$;ii)对任意$e'\in v'$,如果$e'$是$M$的输出事件,则$e'\in s.E$;

6)$F''=S''$。

对于两个EHA H和H',我们简单用$H\|H'$表示$K(H)\|K(H')$对应的结构。合成后的结构反映了H和H'刻画的对象组成的同步系统的整体行为,使得对系统进行验证成为可行。图3和图4各自对应的LTS进行同步合成后的结构如图6所示(其中p和q取值进行简写,如$q0$表示$q=0$)。

对于多个对象组成的系统,可以依次合成得到整个系统的结构,下面定理保证这种合成的正确性。由于本文定理和推论的证明过程所占篇幅过多,因此均被略去。

定理1 同步合成对于结构同构(isomorphic)是可交换和可结合的,即$H\|H'$和$H'\|H$同构,$(H\|H')\|H''$和$H\|(H'\|H'')$同构。

4 并发组合验证

在并发系统中,全局行为的性质经常可以分解为系统所含小部件行为的性质,于是希望只使用系统的一部分来检验每个局部性质。当这种基本形式因为组件之间相互依赖而不可行时,就需要在验证一个组件的性质时必须假设其他组件的行为,这些假设在后来其它组件的正确性被建立后必须抛弃掉。该方法典型的推理过程如下[8]:

$$\frac{\langle\rangle M\langle\varphi\rangle \quad \langle\varphi\rangle M'\langle\psi\rangle}{\langle\rangle M\|M'\langle\psi\rangle}$$

该推断过程表示:M满足ψ,且如果M'的环境满足φ,则M'满足ψ,那么M和M'的合成满足ψ。以这种方式进行验证的好处在于:不用检验$M\|M'$的组合状态空间,而只用M验证φ,并只用M'和假设φ(可能很简单)来验证ψ。为了避免错误的结论,基本假设必须是不用任何假设来证明的,就像$\langle\rangle M\langle\varphi\rangle$。

我们在此考虑对LTL公式描述的并发系统性质进行验证。描述UML statecharts性质的LTL公式应包含三种形式的原子命题[4]:(1) $x\ rop\ c$,其中rop是关系符(例如$>$、$<$、$\neq$等),当变量x和常量c满足rop时为真;(2) $@s_i$,其中s_i是状态名,当UML statecharts的当前活跃格局包括s_i时为真;(3)$\hat{}\ e$在当前状况的事件队列中包含事件e时为真。

上面组合推理的关键问题是建立$\langle\varphi\rangle M'\langle\psi\rangle$的正确性,一个比较好的方式是对有穷状态系统提供一种前序$\prec$来体现"更多行为"的概念,并使用语义与前序相关的逻辑[10]。该前序应保持逻辑公式的可满足性,也就是一个公式对一个模型为真时,对前序中更小的模型也应该为真:

$$(P\prec Q)\wedge Q\models\varphi\rightarrow P\models\varphi$$

下面我们先定义EHA操作语义结构中的路径,然后借鉴文[11]中的思想定义同态模拟作为前序关系并寻求其性质。这些性质将使UML模型的并发组合推理成为可能。

定义6(路径,Path) M中的路径是一个无穷的状态序列$\pi=s_0s_1s_2\cdots$,使得对所有$i\geqslant 0$存在事件集合E_i,有$(s_i,E_i,s_{i+1})\in R$。

定义7(同态模拟,Homomorphism Simulation) AP和AP'是两个结构M和M'分别对应的命题集,$AP\supseteq AP'$。关系$\mathscr{R}\subseteq S\times S'$是一个$M$到$M'$的同态关系,当且仅当对所有满足$\mathscr{R}(s,s')$的$s$和$s'$有以下条件成立:

a) $L(s)\cap AP'=L'(s')$,并且

b)对于M中每条从$s=s_0$开始的路径$\pi=s_0s_1\cdots$,在M'中存在一条从$s'=s'_0$开始的路径$\pi'=s'_0s'_1\cdots$,对每个$i\geqslant 0$有$\mathscr{R}(s_i,s'_i)$。

当存在一个同态$\mathscr{R}$使得对每个$s_0\in S_0$,存在$s'_0\in S'_0$满足$\mathscr{R}(s_0,s'_0)$时,称M'同态模拟M。对于上面b)中的两条路径π和π',可以记作$\mathscr{R}(\pi,\pi')$。当M'同态模拟M时,记为$M\prec_s M'$。

在同态模拟中,第二个结构可以看作第一个结构的规约,而第一个结构可以看作第二个结构的实现。由于规约可能隐藏的一些细节,因此可能有更小的原子命题集合。对于两个EHA H和H',定义$H\prec_s H'$当且仅当$K(H)\prec_s K(H')$。

定理2 $\prec_s$满足下列性质:

1)$\prec_s$是一个前序;

2)对于所有的H和H',$H\|H'\prec_s H$;

3)对所有的H、H'和H'',如果$H\prec_s H'$,那么$H\|H''\prec_s H'\|H''$;

4)对于所有H,$H\prec_s H\|H$。

$\prec_s$是前序意味着它有传递性,这使得可能用$\prec_s$作链式的推理。性质2)使得可以用系统中的一个模块作为系统的一个高层规范。性质3)表明可以用一个实现模块的规范代替它来进行合成,并能得到系统的一个高层规范。

定理3 如果$H\prec_s H'$,则对每个由AP'中原子命题构成LTL公式φ,$H'\models\varphi$则$H\models\varphi$。

LTL的一个重要理论基础是公式f可以转换为一个表示相同无穷状态序列(模型)的ω自动机(如Büchi自动机)[12]。假定公式φ所对应的Büchi自动机为$T(\varphi)$。

定理4 对于一个LTL公式φ,如果$H\models\varphi$,则有$K(H)\prec_s T(\varphi)$。

于是对于UML模型的组合推理可以和标准的模型检验问题结合起来。推论1是根据上述各定理得到针对EHA的两种典型组合推理方法。这样,验证$H\|H'$的性质有可能避免直接生成其组合空间。由于$T(\varphi)$通常比$K(H)$小得多,因此$T(\varphi)\|K(H')$的状态空间就可能比$H\|H'$要小。

推论1 对于两个EHA H和H',以及LTL公式φ、ψ、χ,下面两个推理过程成立:

$$\text{(a)}\quad \frac{H\models\varphi \quad T(\varphi)\|K(H')\models\psi}{H\|H'\models\psi} \qquad \text{(b)}\quad \frac{H\models\varphi \quad T(\psi)\|K(H)\models\chi}{H\|H'\models\chi}$$

对于对象之间只有事件交互的系统,所假设的性质经常和事件的产生有关。对于上节中的例子,现要验证系统是否满足性质$\psi:G(p=0\vee q=0)$,即$H\|H'\models\psi$是否成立。利用组合验证方法,我们首先可以验证H'满足性质$\varphi:G(\hat{}\ r0\rightarrow X$

(下转封四)

(上接第 233 页)

$(q=0))$,$H'|=\varphi$。然后可以验证 $T(\varphi)\|K(H)|=\psi$,并根据(a)可以推出 $H\|H'|=\psi$。性质 φ 对应的 Büchi 自动机如图 7 所示,比起 H' 对应的 LTS 包含 6 个状态和 12 个迁移,状态和迁移数目均减少。$T(\varphi)\|H$ 的结果如图 8 所示,该例子中状态和迁移数目与 $H\|H'$ 一样。但在做合成过程中所需的时空开销得到缩减,而且乘积自动机中每个状态的原子命题数目减少,在验证过程中也能节省开销。在很多情况下,$T(\varphi)\|K(H)$的结构要比 $H\|H'$ 小。

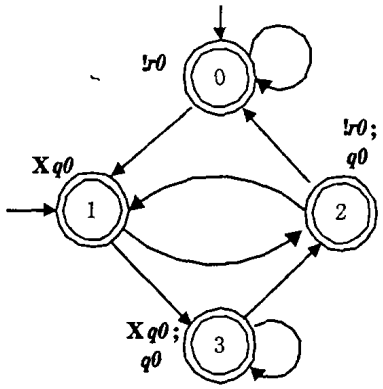

图 7 $G(\hat{}\ r0\rightarrow X(q0))$的 Büchi 自动机

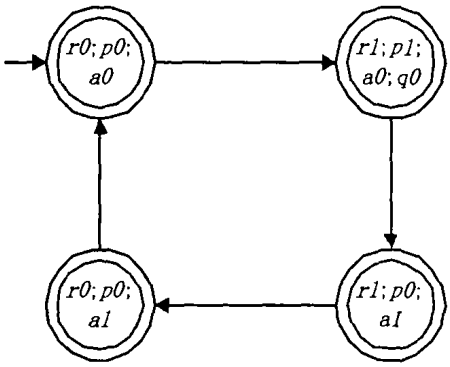

图 8 $T(\varphi)\|H$ 的系统状态图

相关工作和结论 文[11]基于 Moore 机描述一个对有穷状态进程进行组合验证的框架,对∀ CTL * 公式进行验证。它给出了一个结构的前序来得到模块和包含模块的系统之间的关系,并给出在该框架下进行 assume-guarantee 推理的方式。文[12]使用组合最小化技术,对两个通过信道连接的进程 P 和 Q,对 Q 产生一个缩减的版本 Q',Q' 只体现 Q 对 P 可见的行为,所以称为"接口进程"。但是由于系统的非封闭性以及模型的复杂性,该方法也表现出许多缺点。文[13]给出了反应模块的循环推理方法,可以处理以循环方式表达的环境假设。

本文用扩展层次自动机结构化地表示 UML Statecharts 并保持语义的正确性,该语义可由一个 LTS 表示。通过结构合成可以得到包含多个并发对象的同步系统的整体行为,据此可以对系统进行组合验证,有可能不用为多个并发对象建立全局状态迁移图来得到整个系统的正确性。我们的下一步工作包括针对一个较大的实例进行应用,以及如何对异步方式系统进行组合验证等。

参 考 文 献

1 UML1.4 specification, OMG document, 2001
2 Lilius J, Paltor I P. vUML: a Tool for Verfying UML Models. TUCS Centre for Computer Science: [TUCS Technical Report No 272]. 1999
3 Gnesi S, Latella D. Model Checking UML Statecharts Diagrams using JACK. In: Proc. of the 4th IEEE Intl. Symposium on High-Assurance Systems Engineering, 1999
4 Dong W, Wang J, Qi X, Qi Z C. Model Checking UML Statecharts. In: Proc. of the Eighth Asia-Pacific Software Engineering Conference (APSEC 2001). IEEE Computer Society Press, Dec. 2001
5 Scaffer T, Knapp A, Merz S. Model Checking UML State Machines and Collaborations. Electronic Notes in Theoretical Computer Science, 2001, 55(3)
6 Knapp A, Merz S, Rauh C. Model Checking Timed UML State Machines and Collaborations
7 Heimdahl M P E, Whalen M W. Reduction and Slicing of Hierarchical State Machines. In: Proc. of the Fifth ACM SIGSOFT Symposium on the Foundations of Software Engineering, Sep. 1997
8 Pnueli A. In Transition for Global to Modular Temporal Reasoning About Programs. In: K. R. Apt, ed. Logics and Models of Concurrent Systems, NATO ASI 13. Springer, 1984
9 Long D E. Model Checking, Abstraction, and Compositional Verification: [Thesis of PH. D]. 1993
10 Peng H, Tahar S. A Survey on Compositional Verification: [Technical Report]. Dept. of Electrical & Computer Engineering, Concordia University. Nov. 1998
11 Grumberg O, Long D E. Model checking and modular verification. ACM Transitions on Programming Languages and Systems, 1994, 16(3): 843～871
12 Clarke E M, Long D E, McMillan K L. Compositional model checking. In proc. 4th Annual Symposium on Logic in Computer Science, Pacific Grove, CA, June1989. 353～362
13 Henzinger T A, Qadeer S, Rajamani S K. You assume, we guarantee: Methodology and case studies. In: Proc. 10th Intl. Conf. on Computer Aided Verification. Springer-Verlag, 1998

计算机科学

(1974 年 1 月创刊)
第 32 卷第 7 期 (月刊)
2005 年 7 月 25 日出版

ISSN 1002－137X
CN50－1075/TP

定价: 25.00 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: jsjkx@swic.ac.cn
网址: www.jsjkx.com
社 长: 牟炳林
主 编: 彭 丹
副 主 编: 朱宗元
主编助理: 徐书令
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