公平转换系统规范及其应用*

贾国平 郑国梁

(南京大学计算机科学系 南京 210093)

摘要 本文讨论了用于并发系统规范的2种方法:时序逻辑方法和状态自动机方法.由此,本文提出了一种新的规范形式——公平转换系统规范FTSS(fair transition system specification).此规范方法集成了状态自动机方法和时序逻辑方法的优点,改进了时序逻辑方法通常较复杂、不易理解,特别是它不能用于描述并发系统的局部性质等不足.进一步对FTSS中的每一部分进行了讨论,得到结论:FTSS是机器封闭的,规范过程是相容的且是完全的.一个有丢失传输协议的例子表明作者的方法具有简单、直观、易于理解和便于使用等特点.最后给出了FTSS的一些应用.它为程序验证和并发系统的逐步求精提供了一个统一的框架,已成功地应用于程序验证中.

关键词 并发系统,规范,时序逻辑,状态自动机,公平转换系统规范,有丢失传输协议,程序验证,并发系统的逐步求精.

目前主要有2种用于说明一个程序的方法:一种是基于逻辑的方法.此方法一般是给出程序应该满足的性质集合.另一种是基于模型的方法.此方法一般是给出一个抽象模型,此模型指出程序应该如何活动.时序逻辑方法属于第一种方法.它规范程序的过程是逐个列出程序应该有的性质.而许多其它基于抽象程序或自动机的方法,如状态自动机、CCS以及Statecharts等,均属于后者.

就目前所应用的时序逻辑而言,它主要存在2个缺陷:①它并不能方便地用于表达许多并发系统的性质,其规范常常复杂且不易理解,特别当出现嵌套"Until"形式.②本质上,目前所使用的标准时序逻辑只能用于说明整个并发系统,而不能用于说明其中的单个部件,即它只能用于说明系统的全局性质,而不能说明局部性质.因为它只考虑变化的程序状态,而并不考虑引起此变化的转换(或活动).

对比时序逻辑方法,作为基于模型方法的状态自动机方法正好从某种程度上对时序逻辑方法的上述不足提供了补偿.此方法有一个类似于程序设计语言的语法,由它给出的系统规范即抽象模型是非常易于理解的.另外,状态自动机方法只用了2个基本概念,即状态和转换.这更易于为不熟悉逻辑的人所接受和理解.同时,此方法更适于对系统的局部性质的描述.

* 本文研究得到国家教委博士点基金,江苏省应用基础基金资助.作者贾国平,1968年生,博士生,主要研究领域为软件工程,形式化方法,时序逻辑.郑国梁,1937年生,教授,博士导师,主要研究领域为软件工程,软件开发环境.

本文通讯联系人:贾国平,南京 210093,南京大学计算机科学系

本文1995-09-25收到修改稿

基于上述对比讨论,我们的问题是2种方法是否可以较好地相结合?

本文基于上述目的提出了一种新的规范形式:公平转换系统规范FTSS(fair transition system specification).此规范方法集成了状态自动机方法及时序逻辑方法的优点.一个简单的例子表明此规范方法具有简单、直观、易于理解和便于使用等特点.此规范形式为并发系统的规范和验证以及逐步求精提供了一个统一的框架.我们已将它成功地用于程序验证当中.

1 公平转换系统规范

1.1 基本模型

本节提出一般的公平转换系统模型作为一种抽象的实现语言.我们给出了与此模型有关的一些基本概念.此基本模型包含了许多具有不同语法表示及通信机制的并发系统,与许多具体程序设计语言之间的对应关系可参见文献[1].为了表达公平转换系统的语法,我们用了一个基本的一阶语言,此语言中的公式称为断言.

一个公平转换系统S是一个六元组$\langle V,\Sigma,\Theta,T,WF,SF\rangle$,其中$V=\{u_1,\ldots,u_k\}$:状态变量的有穷集合;$\Sigma$:状态集合;$\Theta$:初始条件;$T$:有穷转换集合;$WF\subseteq T$:弱公平性转换集;$SF\subseteq T$:强公平性转换集.

每一状态$s\in\Sigma$是V的一个解释,它给每一变量$u\in V$赋给其对应论域中的一个值.我们用$s[u]$表示.对一个状态$s\in\Sigma$,如果它满足Θ,即$s\mid=\Theta$,则称s为初始状态.每一转换$\tau\in T$是一函数:$\Sigma\to 2^{\Sigma}$,它将每个状态$s\in\Sigma$映入(可能空)τ—后继状态集合$\tau(s)\subseteq\Sigma$.一个转换在状态s下是能行的当且仅当$\tau(s)\neq\varnothing$,否则τ在此状态下是不能行的.我们在T中加入一个转换τ_I,称为空转换,它是一个单位转换,即对每一状态$s\in\Sigma$,$\tau_I(s)=\{s\}$.每一转换τ都由一个断言来表示,记为$\rho_\tau(V,V')$,称为转换关系.它将状态$s\in\Sigma$与它的τ—后继状态$s'\in\tau(s)$通过无上撇号和有上撇号的状态变量联系起来.无上撇号的状态变量引用s中的值.有上撇号的状态变量引用s'中的值.我们说转换关系$\rho_\tau(V,V')$标识出s'是s的一个τ—后继,如果$\langle s,s'\rangle\mid=\rho_\tau(V,V')$,其中$\langle s,s'\rangle$是联合解释,它将$x\in V$解释为$s[x]$,$x'$解释为$s'[x]$.

我们称Σ中的无穷状态序列$\sigma:s_0,s_1,\cdots$是公平转换系统S的一个计算(Computation),如果σ满足:

- 初始化条件:s_0是初始状态,即$s_0\mid=\Theta$.
- 连续性:对每一$j=0,1,\cdots$,状态s_{j+1}是状态s_j的τ—后继状态,即存在转换$\tau\in T$,使$s_{j+1}\in\tau(s_j)$.
- 弱公平性:对每一转换$\tau\in WF$,不会发生τ在σ中某一位置以后一直能行,但只有有限多次被执行.
- 强公平性:对每一转换$\tau\in SF$,不会发生τ在σ中无穷多次能行,但只有有限多次被执行.

对一个公平转换系统S,我们记$Comp(S)$为系统S的所有计算的集合.

1.2 时序逻辑

本节给出时序逻辑的语法及语义,将它作为我们的规范语言.

时序逻辑是一般一阶逻辑语言的扩充.它除了包含一般逻辑算子,如布尔联接词 ~, ∧, ∨, ≡, →;等式算子 = 以及一阶量词∃, ∀ 等外,还包含 2 个时序算子:○(下一值),U(直到),其中算子○既可作用于公式,也可作用于项.其它时序算子我们作为简写,如 $\Diamond p \Leftrightarrow trueUp$, $\Box p \Leftrightarrow \sim\Diamond\sim p$, $pWq \Leftrightarrow \Box p \vee (pUq)$ 以及 $p \Rightarrow q \Leftrightarrow \Box(p \rightarrow q)$.

不包含任何时序算子(包括施用于项的下一值算子)的公式称为断言.

我们称出现在公式 p 中的所有变量集合为 p 的词汇.

为了简单起见,假设一个基本的断言语言为 L,它包含谓词演算以及用于表达某些具体论域上的标准算子及谓词的解释符.

我们首先介绍一些概念.

在词汇 V 上的一个结构是一个无穷状态序列 σ: $s_0, s_1, s_2, \cdots$,其中每一状态 s_j 是对 V 中变量的一个解释.

令 x 是 V 中一变量,状态 s' 称为状态 s 的 x—变体,如果状态 s 和状态 s' 除了 x 外对其它所有变量的解释都相同,而对 x 的解释可能不同.一个结构 σ': $s'_0, s'_1, s'_2, \cdots$ 是结构 σ: $s_0, s_1, s_2, \cdots$ 的 x—变体,如果对每一 $j \geqslant 0$, s'_j 是 s_j 的 x—变体.

给定一结构 σ,我们首先归纳给出项 t 在 σ 中位置 $j \geqslant 0$ 的值,记为 $val(\sigma, j, t)$.

· 对每一变量 $x \in V$, $val(\sigma, j, x) = s_j[x]$.

· 对 L 中每一 n—元函数 f 和项 $t_1, \cdots, t_n$, $val(\sigma, j, f(t_1, \cdots, t_n)) = f(val(\sigma, j, t_1), \cdots, val(\sigma, j, t_n))$.

· 对每一项 t, $val(\sigma, j, t^+) = val(\sigma, j+1, t)$,其中 t^+ 为项 t 的下一值.

下面我们归纳定义一个时序公式 p 在结构 σ 中位置 $j \geqslant 0$ 成立(记为 $(\sigma, j) \models p$)如下:

· 对 L 中每一 n—元谓词 p 和项 $t_1, \cdots, t_n$, $(\sigma, j) \models p(t_1, \cdots, t_n)$ 当且仅当 $p(val(\sigma, j, t_1), \cdots, val(\sigma, j, t_n)) = true$.

· $(\sigma, j) \models \sim p$ 当且仅当 $(\sigma, j) \models p$ 不成立.

· $(\sigma, j) \models p \vee q$ 当且仅当 $(\sigma, j) \models p$ 或 $(\sigma, j) \models q$.

· $(\sigma, j) \models Op$ 当且仅当 $(\sigma, j+1) \models p$.

· $(\sigma, j) \models pUq$ 当且仅当对某些 $k \geqslant j$, $(\sigma, k) \models q$,并且对所有 i: $j \leqslant i < k$,有 $(\sigma, i) \models p$.

· $(\sigma, j) \models \forall x.p$ 当且仅当对所有 σ 的 x—变体 σ',有 $(\sigma', j) \models p$.

· $(\sigma, j) \models \exists x.p$ 当且仅当对某些 σ 的 x—变体 σ',有 $(\sigma', j) \models p$.

当我们下面考虑一个说明转换系统 S 性质的公式 p 时,总是假设词汇 V 包含 S 的所有状态变量以及出现在 p 中的所有变量.在结构及计算中所用的状态均是对 V 中变量的解释.

V 中的变量可以分为严格变量和可变变量 2 种.严格变量在计算的不同状态上有相同的值,而可变变量在计算的不同状态上值可能不同.转换系统中的状态变量就为可变变量.严格变量主要用于规范目的,它并不出现在系统之中.它主要用于连接计算序列中不同状态中的值.

在所有状态上都成立的断言称为断言有效的(Assertionally Valid).

如果 $(\sigma, 0) \models p$,则称公式 p 在结构 σ 上成立,记为 $\sigma \models p$.一个公式 p 称为是可满足的

(Satisfiable),如果它在某些结构上成立.一个公式 p 称为是时序有效的(Temporally Valid),如果它在所有的结构上都成立.

给定一转换系统 S,我们将注意力限制在结构集 $Comp(S)$ 上,即由 S 的所有计算组成的结构集上.如果公式 p 在 S 的所有计算上均成立,则称 p 在 S 上是有效的.显然,每一时序有效公式对任何系统 S 在其上均是有效的.

1.3 公平转换系统的时序语义

对于一个给定的公平转换系统 S,本节构造出一个时序公式 $Sem(S)$,称为 S 的时序语义.我们给出此公式具有的一个性质,即 $Sem(S)$ 在结构 σ 上成立当且仅当 σ 是 S 的一个计算.这个结论最初由 Pnueli 在文献[2]中给出.

令一个公平转换系统 S 为 $\langle V,\Sigma,\Theta,T,WF,SF\rangle$.首先我们引入一些公式,它们分别表达了 S 的计算的不同性质.

• $En(\tau)$:$(\exists V')\rho_\tau(V,V')$,其含义为转换关系为 $\rho_\tau(V,V')$ 的转换 τ 是能行的.

• $taken(\tau)$:$\rho_\tau(V,V^+)$,其中 $\rho_\tau(V,V^+)$ 通过将 $\rho_\tau(V,V')$ 中每一变量 y' 由 y^+ 代替而得.假设 $taken(\tau)$ 在一个计算的 j 位置成立.则上述公式含义为如果取 $y\in V$ 为状态 s_j 中的值,即 $s_j[y]$,y' 为值 $val(\sigma,j,y^+)=val(\sigma,j+1,y)=s_{j+1}[y]$,则关系 ρ_τ 成立,即 s_{j+1} 是 s_j 的 τ—后继.

• $wf(\tau)$:$\Diamond\Box\, En(\tau)\rightarrow\Box\Diamond taken(\tau)$,此公式说明如果 τ 除了有限多个位置外均能行,则 τ 被执行无穷多次.因此,任何满足 $wf(\tau)$ 的计算序列对于转换 τ 是满足弱公平性的.

• $sf(\tau)$:$\Box\Diamond En(\tau)\rightarrow\Box\Diamond taken(\tau)$,此公式说明如果 τ 无穷多次能行,则 τ 被执行无穷多次.因此,任何满足 $sf(\tau)$ 的计算序列对于转换 τ 是满足强公平性的.

对一给定的系统 S,我们定义它的时序语义公式 $Sem(S)$ 为:

$$Sem(S):\ \Theta\wedge\Box\bigvee_{\tau\in T} taken(\tau)\wedge\bigwedge_{\tau\in WF} wf(\tau)\wedge\bigwedge_{\tau\in SF} sf(\tau) \qquad (*)$$

下面我们考虑时序语义公式 $Sem(S)$ 中的每一项:

• 合取项 Θ 确保系统 S 的计算序列中的初始状态满足初始条件 Θ.

• $\Box\bigvee_{\tau\in T} taken(\tau)$ 保证对每一 $i=0,1,\cdots$,状态 s_{i+1} 是在状态 s_i 上通过施用某一转换 $\tau\in T$ 而得到的.

• $\bigwedge_{\tau\in WF} wf(\tau)$ 保证系统 S 的计算序列满足所有的弱公平性假设.

• $\bigwedge_{\tau\in SF} sf(\tau)$ 保证系统 S 的计算序列满足所有的强公平性假设.

对比 1.1 节对公平转换系统 S 中计算的定义,我们知道公式 $Sem(S)$ 中的上述 4 项分别对应且保证定义中的 4 个要求.因此公式 $Sem(S)$ 精确地表示了系统 S 的计算.此结果由下述命题给出:

命题 1. 一个结构 σ 满足公式 $Sem(S)$ 当且仅当 σ 是公平转换系统 S 的一个计算.

证明:此结果可以简单地通过将公式 $Sem(S)$ 中的每一项与公平转换系统 S 的计算的定义中的每一项相比较而得到,证明略.

1.4 公平转换系统规范

本节首先给出一些定义,然后,给出我们的公平转换系统规范.

定义 1. 一个转换模块 M 是一个系统 $M:\langle V,U,\Sigma,\Theta,T,WF,SF\rangle$,其中 $S_M:\langle V,\Sigma,\Theta,$

$T,WF,SF\rangle$是一个公平转换系统,$U\subseteq V$是一个集合.我们称S_M是M的体,U为M的隐藏变量集.

定义 2. 一个体为S_M,隐藏变量集为U的转换模块M,它的一个执行是S_M的一个计算的任何U-变体,即与S_M的一个计算相比至多对U中变量的解释不同的任何结构.

下面给出我们的公平转换系统规范形式:

定义 3. 我们称一个时序公式具有公平转换系统规范形式,如果它有形式:$\exists U.Sem(S)$,其中S是一公平转换系统,其状态变量集为V.$Sem(S)$为S的时序语义公式.U为V的一个子集.

由此定义可知,对于某一转换模块M,其中$S:\langle V,\Sigma,\Theta,T,WF,SF\rangle$是一公平转换系统,$U\subseteq V$为其隐藏变量集合,我们的公平转换系统规范具有下述形式:

$$\exists U.[\Theta\wedge\Box\bigvee_{\tau\in T}taken(\tau)\wedge\bigwedge_{\tau\in WF}wf(\tau)\wedge\bigwedge_{\tau\in SF}sf(\tau)] \qquad (**)$$

其中Θ是一初始状态断言,它说明了转换模块M的初始状态,即说明了变量的初始值.

$\Box\bigvee_{\tau\in T}taken(\tau)$是转换模块$M$的下一状态关系,即对每一$i=0,1,\cdots$,状态$s_{i+1}$由状态$s_i$通过执行某一转换$\tau\in T$而得到.

$\bigwedge_{\tau\in WF}wf(\tau)$及$\bigwedge_{\tau\in SF}sf(\tau)$确保转换模块$M$的执行序列分别满足所有弱公平性和强公平性假设.

U是转换模块M的隐藏变量集合或称为内部变量集合.

显然,我们的公平转换系统规范(**)精确地表示了转换模块M的执行,即一个结构σ满足(**)当且仅当σ是M的一个执行.此结果由下述命题给出:

命题 2. 一个结构σ满足公平转换系统规范(**)当且仅当σ是转换模块M的一个执行.

记$L=\bigwedge_{\tau\in WF}wf(\tau)\wedge\bigwedge_{\tau\in SF}sf(\tau)$.由前述定义及讨论,公平转换系统规范(**)有如下解释:

存在某种选取隐藏变量集U的方式,使得:(1)谓词Θ在初始状态为真.(2)系统M的每一步执行或者是某一转换$\tau\in(T\backslash\{\tau_I\})$的执行$taken(\tau)$或者是单位转换$\tau_I$的执行$taken(\tau_I)$,此时系统$M$的所有变量保持不变.(3)系统$M$的整个行为满足$L$,即满足所有弱公平性和强公平性假设.

下面我们对公平转换系统规范进行一些讨论.

在公平转换系统规范中,规范本身是一个时序公式,它精确地描述了系统的一个行为集合,其中行为是一个无穷状态序列.一个性质是一特定的行为集合.一个有穷状态序列称为有穷行为.我们称一个有穷行为满足性质F,如果它可以一直连续到F中的一个无穷行为,即有穷行为$s_0,s_1,\cdots,s_n$满足F当且仅当无穷行为$s_0,s_1,\cdots,s_n,s_n,s_n,\cdots$在$F$中.我们的规范形式说明了系统的2个基本性质:安全性(Safety)和活性(Liveness).

性质F是安全性当且仅当下述条件成立:F包含一个行为当且仅当F被此行为的每一有穷行为前缀所满足.性质L是活性当且仅当任何有穷行为都能够扩充为L中的行为.由此定义可知公平转换系统规范中,$\Theta\wedge\Box\bigvee_{\tau\in T}taken(\tau)$是安全性.对$\tau\in WF$,$wf(\tau)$以及对$\tau\in SF$,$sf(\tau)$均是活性.

安全性在所有行为集的拓扑上是一闭集.活性在所有行为集的拓扑上是一稠密集.由于在一拓扑空间中,每一集合都能表示为一个闭集和一个稠密集的交集,因此,任何性质均可写为一安全性和一活性的合取.我们的公平转换系统规范采用了安全性—活性分类范式,其主要目的是为了保证规范的完全性.然而,采用上述分类范式有一点需要特别注意,即在说明活性部分时,加入任意形式的活性是危险的,因为它可能同时又加入了不期望有的安全性.由此附带加入的安全性是时序逻辑规范中错误的根源,同时也是证明方法不完备的根源.在公平转换系统规范中,我们用公平性来说明活性.用此方法规范系统就可避免说明活性时引入额外的安全性,避免规范产生错误.为了讨论这一点,下面我们引入机器封闭的概念.它可以看成是对公平性概念的推广.

定义 4. 如果 F 是一安全性,L 是任一性质,则公式对(F,L)是机器封闭的当且仅当任何满足 F 的有穷行为都能扩充为 $F\wedge L$ 中的一无穷行为.我们称 $F\wedge L$ 为机器封闭规范.

由上述定义可知,(F,L)是机器封闭的当且仅当 F 与 L 的合取并未引入额外的安全性.因此,我们可以通过写机器封闭的规范来避免引入不期望有的安全性.

定理 1. 如果 F 是一安全性,L 为有穷或可数无穷多个形如 $wf(\tau)$和(或)$sf(\tau)$的公式的合取式,其中每一 $taken(\tau)$蕴含 F,即 $taken(\tau)\rightarrow F$,则(F,L)是机器封闭的.

证明:此定理可看成为文献[3]中命题 4 的特例,证明完全类似,此处省略.

定理 2. 如果(F,L)是机器封闭的,U 是一变量集,它在公式 L 中无自由出现,且$\exists\, U.F$)是一安全性,则$((\exists\, U.F),L)$是机器封闭的.

证明:此定理可看成为文献[3]中命题 2 的特例,证明类似(略).

考虑公平转换系统规范(* *).由定理 1 我们知道公式对$([\Theta\wedge\Box\bigvee_{\tau\in T}taken(\tau)],L)$是机器封闭的,其中 $L=\bigwedge_{\tau\in WF}wf(\tau)\wedge\bigwedge_{\tau\in SF}sf(\tau)$.又由于 U 中的隐藏变量在 L 中无自由出现,(* *)可写为$(\exists\, U.[\Theta\wedge\Box\bigvee_{\tau\in T}taken(\tau)])\wedge L$.由安全性定义,$\exists\, U.[\Theta\wedge\Box\bigvee_{\tau\in T}taken(\tau)]$是安全性.因此,由定理 2 我们知(* *)是机器封闭的.

由此,公平转换系统规范不会引入额外的安全性,此规范过程是相容的且是完全的.

对比前节的时序语义公式(*)及我们的公平转换系统规范(* *),可以看到它们本质的不同在于作用于隐藏变量上的存在量词.对每一时序逻辑,在可变变量上的存在量词是表达大多数复杂规范所必需的.同时,它也是减小规范复杂性以及增强时序逻辑语言表达能力的有效方法.内部变量如何实现并不重要,它们对于实现者是完全自由的.存在量词作用于这些内部变量就保证了它们是完全自由于实现的.这正如程序设计语言中隐藏(Hiding)的概念.对存在量词的精确含义的理解是理解公平转换系统规范的关键.在下面例子中,我们将对此进一步讨论.

2　例子:有丢失传输协议

我们给出一个有丢失传输协议(Lossy—Transmission Protocol)的例子来说明我们的方法.协议通过 2 个异步"线路对"与它们的环境进行通信.每一线路对包含一个值线路:val,它保存消息;一个布尔字节线路:bit.在一线路对上消息 m 的发送是通过将 m 赋给 val 线路,而对 bit 线路取补码而完成的.接收者通过观察 bit 线路是否改变了其结果来检测一

个消息是否是新的.传递消息的输入是在线路对(ival,ibit)上进行的,输出是在线路对(oval,obit)上进行的(见图 1).

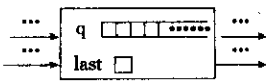

图1 有丢失传输协议

由于并没有确认协议,因此如果输入速度比传输所能处理的速度快,则输入是可以丢失的.此传输协议所要保证的性质是输出消息序列为输入消息序列的一个子序列.

此协议规范在文献[3,4]中分别用转换公理方法和基于 TLA 方法进行了描述.此处作为比较和说明,我们给出它的公平转换系统规范形式.

有丢失传输协议的公平转换系统规范是一个时序公式.它只用到 4 个变量:*ival*,*ibit*,*oval* 和 *obit* 以及 2 个内部变量:*q* 表示已接收到但还未输出的消息序列;*last* 等于前一次所接收到消息的 *ibit* 值.引入它的主要目的是避免同一消息被接收 2 次.这 6 个变量是可变变量.为了规范的目的,我们可以引入严格变量 *Mes*,它表示所有可能的消息集合.

下面引入几个记号:表达式( )表示空消息序列;(m)表示只有 m 为其元素的单元素序列;$\cdot$ 表示合并操作;x' 表示变量 x 的下一值;$Head(\sigma)$表示消息序列 σ 的第 1 个元素;$Tail(\sigma)$表示通过移去 σ 中第 1 个元素而得到的序列.

我们可以定义一个转换模块 $M_{FTSS}:\langle V,U,\Sigma,\Theta,T,WF,SF\rangle$,其中:

$V:\{ival,ibit,oval,obit,q,last\}$, $U:\{q,last\}$, $\Theta:q=(\ )\wedge(ival,oval\in Mes)$,

$T:\{\tau_I,\tau_1,\tau_2,\tau_3\}$,其转换关系如下:

$\rho_{\tau_I}:(ival',ibit',oval',obit',q',last')=(ival,ibit,oval,obit,q,last)$,即 τ_I 为单位转换.

$\rho_{\tau_1}:(ibit'=\sim ibit)\wedge(ival'\in Mes)\wedge(obit',oval',q',last')=(obit,oval,q,last)$.

$\rho_{\tau_2}:(last\neq ibit)\wedge q'=q\cdot(ival)\wedge last'=ibit\wedge(ibit',obit',ival',oval')=(ibit,obit,ival,oval)$.

$\rho_{\tau_3}:q\neq(\ )\wedge oval'=Head(q)\wedge q'=Tail(q)\wedge obit'=\sim obit\wedge(ibit',ival',last')=(ibit,ival,last)$.

$WF:\{\tau_3\}$, $SF:\{\tau_2\}$.

我们的公平转换系统规范为:

$$\exists\, q,last:\left\{\begin{array}{l} q=()\wedge(ival,oval\in Mes)\\ \wedge\Box\left[\begin{array}{l}[(ival',ibit',oval',obit',q',last')=(ival,ibit,oval,obit,q,last)]\\ \vee[(ibit'=\sim ibit)\wedge(ival'\in Mes)\wedge(obit',oval',q',last')=(obit,oval,q,last)]\\ \vee[last\neq ibit\wedge(q'=q\cdot(ival))\wedge last'=ibit\wedge(ibit',obit',ival',oval')=(ibit,obit,ival,oval)]\\ \vee[q\neq()\wedge oval'=Head(q)\wedge q'=Tail(q)\wedge obit'=\sim obit\wedge(ibit',ival',last')=(ibit,ival,last)]\end{array}\right]\\ \wedge wf(\tau_3)\\ \wedge sf(\tau_2)\end{array}\right.$$

其中,性质 $wf(\tau_3)$说明如果转换 τ_3 在某一位置以后永远能行,则 τ_3 被执行无穷多次,即 $taken(\tau_3)$发生无穷多次.此性质隐含每一到达 q 的消息最终一定被输出.性质 $sf(\tau_2)$说明如果转换 τ_2 无穷多次能行,则 τ_2 被执行无穷多次,即 $taken(\tau_2)$发生无穷多次.此性质隐含如果有无穷多个消息被发送,则 q 一定接收到无穷多个消息.因此,$wf(\tau_3)\wedge sf(\tau_2)$隐含活性:无限多个输入产生无限多个输出.它也就保证协议满足输出消息序列是输入消息序列的一个

子序列的要求.

值得指出的是,虽然我们的规范用了一个表值变量 q 来表达期望的性质,但这并不意味着在系统实现时必须采用相同的数据结构.这仅仅是一种使规范更易于表示和理解的措施.保证上述规范是抽象的且其实现是完全自由的,关键就在于在可变变量 q 及 *last* 上的存在量词.它非常类似于程序设计语言中隐藏(Hiding)的概念.

3 结束语

Lamport 在文献[4]中提出了一种转换公理方法.此方法也结合了时序逻辑与状态自动机方法各自的优点.其中引入的"允许变化"(Allowed Changes)算子直观上较易理解,但其语义却似乎非常复杂.文献[5]基于转换公理方法,将时序逻辑与活动逻辑(Logic of Actions)相结合得到 TLA,采用 TLA 得到 TLA 规范形式,此规范简单、较易理解.我们的公平转换系统规范方法基于 Manna—Pnueli 的时序逻辑框架[1],其时序语义清楚明了.另外,由于公平转换系统规范既可看成是一转换模块,同时规范本身又是一时序公式,它精确描述了系统的允许行为集.当选用时序逻辑作为规范语言时,这种规范形式一方面为程序验证提供了一个统一框架,另一方面它也为并发系统自顶向下逐步求精提供了基础.我们可以充分利用现有的许多用于证明并发系统时序性质的完全证明系统.我们的方法已成功地用于程序验证.这些应用表明此方法是可行的.

参考文献

1 Manna Z, Pnueli A. The temporal logic of reactive and concurrent system: specification. New York: Spinger—Verlag, 1991.

2 Pnueli A. The temporal semantics of concurrent programs. Theor. Comp. Sci., 1981, **13**: 1～20.

3 Abadi M, Lamport L. An old—fashioned recipe for real time. ACM Trans. on Prog. Lang. and Sys., 1994, **16**(5): 1543～1571.

4 Lamport L. Specifying concurrent program modules. ACM Trans. on Prog. Lang. and Sys., 1983, **5**: 190～222.

5 Lamport L. The temporal logic of actions. ACM Trans. on Prog. Lang. and Sys., 1994, **16(3)**: 872～923.

FAIR TRANSITION SYSTEM SPECIFICATION AND ITS APPLICATIONS

Jia Guoping　Zheng Guoliang

(*Department of Computer Science　Nanjing University　Nanjing*　210093)

Abstract　This paper discusses two approaches to the specification of concurrent systems: temporal logic approach and state machine approach. As a result of this discussion, the authors propose a new kind of formalisms of specification: fair transition system specification (FTSS). This specification approach combines the best features of temporal logic

and state machine methods and revises these drawbacks that temporal logic approach usually is complicated and not easy to understand, especially, it fails to be used for "local" properties of concurrent systems. The authors further consider each component of FTSS and get conclusions that FTSS is machine closed and the specification process is consistent and complete. An example of a lossy—transmission protocol shows that the presented approach is simple and easy to understand and to use. At the end of the paper, some applications of FTSS are given. The approach provides a unified framework for program verification and step—wise refinement of concurrent systems. It has been successfully applied to program verification.

Key words Concurrent system, specification, temporal logic, state machine, fair transition system specification, lossy — transmission protocol, program verification, step—wise refinement of concurrent systems.