TCPN 的组合可调度分析*)

李 鹏 李 勋 顾 庆 陈道蓄
(南京大学软件新技术国家重点实验室 南京 210093)

摘 要 时间约束 Petri 网(Timing Constraints Petri nets,简称 TCPNs)是一类重要的时间 Petri 网系统。针对 TCPNs 中变迁可调度原始语义的不足,本文对相关定义重新定义,丰富并完善了 TPCNs 理论。本文首先给出了新的针对单个变迁或变迁序列的可调度分析策略。如果一个特定的变迁序列是可调度的,则相应的活动序列也同样可以顺利地完成自身的执行;否则,不可调度的变迁需要调整自己的时间约束;然后提出了组合式的可调度分析策略以分析复杂变迁序列,最后提出时序一致性的概念。

关键词 时间约束 Petri 网,可调度性,组合可调度分析,时序一致 XC 性

Compositional Schedulability Analysis of Timing Constraint Petri Nets

LI Peng LI Xun GU Qing CHEN Dao-Xu
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract TCPNs (Timing constraint Petri nets) is an important kind of time-related Petri nets. For correcting the irrationality of the original concepts in TCPN, some basic concepts are redefined to enrich and perfect corresponding theory of TCPN. Firstly, this paper presents a new approach to the schedulability analysis of individual transition or transition sequences in TCPN. If a specific transition sequence is schedulable, the corresponding task sequence can complete its execution successfully; otherwise, nonschedulable transitions should be pinpointed to help adjust timing constraints. Secondly, a technique for compositional timing analysis is also proposed to deal with complex transition sequences. In addition, concept of temporal consistency is also introduced.

Keywords Timing constraints petri nets, Schedulability, Compositional schedulability analysis, Temporal consistency

1 引言

Petri 网在 1962 年被 Carl Adam Petri 作为一种过程建模和分析工具提了出来[3],随着 Petri 网理论的发展和完善,出现了各种各样的 Petri 网系统。Petri 网直观的图形表示和坚实的数学基础使其特别适合描述异步并发系统[3]。因为传统的 Petri 网缺乏对时间的描述,这样就约束了 Petri 网在一些实时系统中的应用。因此,时间概念被引入到 Petri 网中,得到了许多时间扩展的 Petri 网的模型,如 Timed Petri nets (Timed PNs)[6], Stochastic Petri nets (SPNs)[7], Time Petri nets (Time PNs)[8]。

本文主要分析的扩展时间模型为 Timing Constraint Petri nets,此模型由 Tsai[1] 提出。作为一个可视化的模型,TCPNs 具有十分丰富的时间约束信息,而且不同于 Timed PNs, SPNs 以及 Time PNs 采用强触发模式,TCPNs 采用弱触发模式。因此,与普通 Petri 网一样可以描述冲突结构。虽然 TCPNs 对时间描述有许多的优势,但是,Tsai 提出的公式 EFBT 以及 LFET 同时间约束的含义不符[2],而且由 Tsai 提出的可调度分析策略不能够充分证明单个变迁的可调度性。这些缺陷从某种意义上来说也限制了 TCPNs 的应用。

基于新提出的 EFBT 以及 LFET 求值公式,主要成果包括:1)单个变迁或变迁序列的可调度分析策略;2)复杂变迁序列的组合分析策略。本文第 2 部分主要对 TCPN 进行介绍;第 3 部分重新给出了强可调度的定义,并且提出单个变迁以及变迁序列的可调度分析策略;第 4 部分介绍组合分析策略;第 5 部分提出时序一致性的概念;最后进行总结展望。

2 时间约束 Petri 网

根据相关文献,TCPN 可以从形式上定义如下[1]:

定义 1(TCPN) TCPN 是一个六元组 $\langle P, T, F, C, D, M_0\rangle$

- $P=\{p_1, p_2, \cdots, p_m\}$ 为库所有限集;
- $T=\{t_1, t_2, \cdots, t_n\}$ 为变迁有限集;
- F 表示连接库所和变迁的有向弧集合;
- C 为关联库所和变迁的实数对 $(TC_{\min}(pt_j), TC_{\max}(pt_j))$ 的集合;
- D 为变迁的执行延迟,$[FIRE_{dur}(t)]$,t 表示一个变迁;
- M_0 表示初始标识集。

不考虑相应的时间约束,P, T, F, M_0 也可以表示普通的 Petri 网,这个 Petri 网可以称为基网 $UN=\langle P, T, F, M_0\rangle$。给定一个标识 M 以及一个库所 p,$M(p)$ 表示在标识 M 下库所 p 中包含的托肯数。

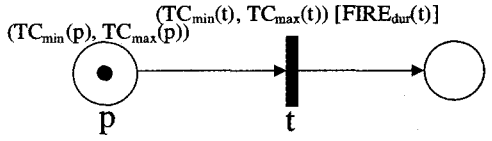

图 1 一个泛化的 TCPN 片段

*)本文得到国家 863 项目支持,项目编号:2001AA113090。李 鹏 硕士研究生,研究方向:Petri 网,工作流系统,软件过程管理;李 勋 硕士研究生,研究方向:工作流,软件过程管理;顾 庆 博士,副教授,研究方向:分布式计算;陈道蓄 博士生导师,研究方向:分布式计算与并行处理。

如图1所示，在TCPN中，有两个值附加于每个库所 p 上，$TC_{\min}(p)$ 以及 $TC_{\max}(p)$，这两个值的含义可以参考文[1]。每个变迁 t 同样附加一个时间对 $(TC_{\min}(t), TC_{\max}(t))$，$TC_{\min}(t)$ 表示相应变迁使能前所必须流逝的最小时间；而 $TC_{\max}(t)$ 表示相应变迁触发前可以经历的最长时间。这样一个变迁 t 在标识 M 下为使能状态当且仅当 $(p \in \cdot t)\ M(p) \geqslant 1$。一个变迁 t，如果在 T_0 时刻使能，则只能在时间区间 $[T_0+TC_{\min}(t), T_0+TC_{\max}(t)]$ 内触发。库所变迁时间对 $(TC_{\min}(pt_j), TC_{\max}(pt_j))$ 均表示相对时间区间。这里我们需要注意的是无法保证一个可触发的变迁 t 一定可以成功完成触发任务，因为 t 的触发需要一段时延 $FIRE_{dur}(t)$。对于没有指定具体时间约束的库所或变迁来说，库所的默认时间对为 $(0,\infty)$，变迁的默认时间对为 $(0,\infty)$，默认的变迁时延为0。

3 TCPN的可调度分析

3.1 基本概念

为了方便对后面的可调度进行分析，我们定义一些新的时间约束，$TOKEN_{arr}(p)$ 表示托肯到达库所 p 的绝对时间，$EFBT(t)/LFET(t)$ 表示变迁 t 在使能之后的最早触发开始/最晚触发结束时间，$IP(t)/OP(t)$ 表示变迁 t 的输入/输出库所集合，$IT(p)/OT(p)$ 表示库所 p 的输入/输出变迁集合。

文[2]指出，文[1]中提出的EFBT以及LFET求值公式同TCPN的时间约束含义不甚一致，因此，在这里给出符合时间约束含义的相应公式如下：

$EFBT(t_i) = \mathrm{Max}\{TOKEN_{arr}(p_j) + TC_{\min}(p_j), \mathrm{Max}\{TOKEN_{arr}(p_j): p_j \in IP(t_i)\} + TC_{\min}(t_i)\}$

$LFET(t_i) = \mathrm{Min}\{TOKEN_{arr}(p_j) + TC_{\max}(p_j), \mathrm{Max}\{TOKEN_{arr}(p_j): p_j \in IP(t_i)\} + TC_{\max}(t_i)\}$

基于新定义的 $EFBT(t_i)$ 以及 $LFET(t_i)$ 求值公式，重新给出强可触发以及强可调度定义：

定义2(强可触发) 如果一个变迁 t_i 的所有输入库所中都至少有一个托肯，若考虑 t_i 的每个输入库所中托肯的到达的时间，则称 t_i 是强可触发的当且仅当条件 $LFET(t_i)-EFBT(t_i) \geqslant 0$

定义3(强可调度) 如果一个变迁 t_i 是强触发的且它能够顺利的完成触发活动，则称 t_i 是强可调度的当且仅当 $LFET(t_i)-EFBT(t_i) \geqslant FIRE_{dur}(t_i)$

3.2 单个变迁的可调度分析

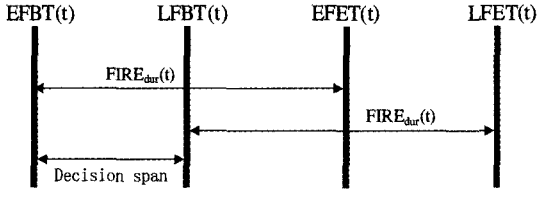

图2 时间约束关系

定理1 如果一个变迁 t 是强可调度，且在实际执行时希望能够顺利完成触发，则 $FIRE_{begin}(t) \in [EFBT(t), LFBT(t)]$。$LFBT(t)$ 表示变迁最迟触发开始时间 t，$LFBT(t) = LFET(t) - FIRE_{dur}(t)$。

证明：如果一个变迁 t 是强可调度，明显可得，$LFET(t) - EFBT(t) \geqslant FIRE_{dur}(t)$。如果在实际执行中 $FIRE_{begin}(t) > LFBT(t)$，则 $FIRE_{end}(t) = FIRE_{begin}(t) + FIRE_{dur}(t) > LFET(t)$，这样就意味 t 不可能成功完成触发。

基于时间变量 $FIRE_{begin}(t)$，$FIRE_{end}(t) \in [EFET(t), LFET(t)]$，$EFET(t)$ 表示变迁 t 的最早结束触发时间，且 $EFET(t) = EFBT(t) + FIRE_{dur}(t)$。时间区间 $[EFBT(t), LFBT(t)]$ 可以认为是一个变迁在实际运行时的决策域度时间(Decision Span)，这些时间约束的关系如图2。

定理2 对于 $\forall p_j \in OP(t_i)$，如果 t_i 顺利完成触发，则 $TOKEN_{arr}(p_j) = FIRE_{end}(t_i) \in [EFET(t_i), LFET(t_i)]$。

证明：因为 $TOKEN_{arr}(p_j) = FIRE_{end}(t_i)$，如果 t_i 能够顺利完成触发，可以直接得到本定理。

定义4 对于 $\forall p_j \in \mathrm{IP}(t_j)$，$\mathrm{Min}(TOKEN_{arr}(p_j))$ 以及 $\mathrm{Max}(TOKEN_{arr}(p_j))$ 可以用来表示 $TOKEN_{arr}(p_j)$ 的上下界。根据本文 $EFBT$ 以及 $LFET$ 求值公式，可得 $EFBT'(t_j)$，$LFET'(t_j)$，$EFBT''(t_j)$ 以及 $LFET''(t_j)$ 如下：

$EFBT'(t_j) = \mathrm{Max}\{\mathrm{Min}(TOKEN_{arr}(p_j)) + TC_{\min}(p_j), \mathrm{Max}\{\mathrm{Min}(TOKEN_{arr}(p_j)): p_j \in IP(t_j)\} + TC_{\min}(t_j)\}$

$LFET'(t_j) = \mathrm{Min}\{\mathrm{Max}(TOKEN_{arr}(p_j)) + TC_{\max}(p_j), \mathrm{Max}\{\mathrm{Max}(TOKEN_{arr}(p_j)): p_j \in IP(t_j)\} + TC_{\max}(t_j)\}$

$EFBT''(t_j) = \mathrm{Max}\{\mathrm{Max}(TOKEN_{arr}(p_j)) + TC_{\min}(p_j), \mathrm{Max}\{\mathrm{Max}(TOKEN_{arr}(p_j)): p_j \in IP(t_j)\} + TC_{\min}(t_j)\}$

$LFET''(t_j) = \mathrm{Min}\{\mathrm{Min}(TOKEN_{arr}(p_j)) + TC_{\max}(p_j), \mathrm{Max}\{\mathrm{Min}(TOKEN_{arr}(p_j)): p_j \in IP(t_j)\} + TC_{\max}(t_j)\}$

并且

$EFBT'(t_j) \leqslant EFBT(t_j) \leqslant EFBT''(t_j)$；$LFET''(t_j) \leqslant LFET(t_j) \leqslant LFET'(t_j)$

单个变迁的可调度性分析分以下三种情况：

Case 1：对于只有一个输入库所 p_j 的变迁 t_j：

$EFBT(t_j) = TOKEN_{arr}(p_j) + \mathrm{Max}\{TC_{\min}(p_j), TC_{\min}(t_j)\}$；

$LFET(t_j) = TOKEN_{arr}(p_j) + \mathrm{Min}\{TC_{\max}(p_j), TC_{\max}(t_j)\}$；

所以 $(LFET(t_j) - EFBT(t_j))$ 的结果不受 $TOKEN_{arr}(p_j)$ 的影响。

Case 2：对于有多个输入库所 $p_j\ (j=1,\cdots,k)$ 的变迁 t_j：

基于定义4，可以得到 $LFET''(t_j) - EFBT''(t_j) \leqslant LFET(t_j) - EFBT(t_j) \leqslant LFET'(t_j) - EFBT'(t_j)$

- 如果 $LFET''(t_j) - EFBT''(t_j) \geqslant FIRE_{dur}(t_j) \Rightarrow LFET(t_j) - EFBT(t_j) \geqslant FIRE_{dur}(t_j)$，则 t_j 为强可调度；
- 如果 $LFET''(t_j) - EFBT''(t_j) < FIRE_{dur}(t_j)$ 且 $LFET'(t_j) - EFBT'(t_j) \geqslant FIRE_{dur}(t_j)$，则变迁 t_j 可能强可调度也可能不是。这种情况下，t_j 处于不安全状态；
- 如果 $LFET'(t_j) - EFBT'(t_j) < FIRE_{dur}(t_j) \Rightarrow LFET(t_j) - EFBT(t_j) < FIRE_{dur}(t_j)$，则变迁 t_j 一定不可调度。

Case 3：对于处于冲突结构中的变迁 $t_j\ (j=1,2,\cdots,k)$：

TCPNs相对于其他时间相关Petri网模型的优势就是它可以像基网UN一样方便地对冲突结构进行描述，冲突结构中的变迁的处理与只有一个输入库所的变迁处理方式相同。

3.3 变迁序列的可调度分析

在基网UN中，如果存在一个序列 $\sigma = (M_0 t_1 M_1 \cdots t_i M_i \cdots t_n M_n)$，或简单变迁序列 $\delta = (t_1 \cdots t_i \cdots t_n)$ 可将 M_0 转换为 M_n，则称标识是可达的[3]。在TCPN中，如果判断一个标识是否可达，则还需要证明变迁序列 δ 在附加的时间约束基础上是可调度的。

定义5 变迁序列 $\delta = (t_1 \cdots t_i \cdots t_n)$ 为可调度当且仅当包含在序列 δ 中的所有变迁都是强可调度的。

在变迁序列的可调度分析中，每个变迁 t_i 都有一个变量 $Root\_Time$，这个变量表示 t_i 所有输入库所的托肯到达时间都可以用 $Root\_Time$ 来统一表示。序列 $\delta = (t_1 \cdots t_n)$ 的可调

度性可以按以下步骤检测：

步骤 1：决定初始标识 M_0 且假设 $TOKEN_{arr}(M_0)$ 为 T_0。

步骤 2：根据变迁序列 δ，找出变迁 $t_i(i=1,\cdots,n)$ 的输入库所，确定 t_i 的 $Root\_Time$ 以及每个输入库所的托肯到达时间，然后检查 t_i 的可调度性。如果 t_i 是强可调度的，则进行步骤 3；否则，需要调整不可调度变迁的时间约束使之强可调度。

步骤 3：需要为每一个变迁 t_i 计算 $FIRE_{end}(t_i)$。

• 对于只有一个输入库所的变迁 t_i，$FIRE_{end}(t_i)\in[EFBT(t_i)+FIRE_{dur}(t_i),LFET(t_i)]$；

• 对于有多个输入库所的变迁 t_i，$FIRE_{end}(t_i)\in[EFBT'(t_i)+FIREdur(t_i),LFET'(t_i)]$。

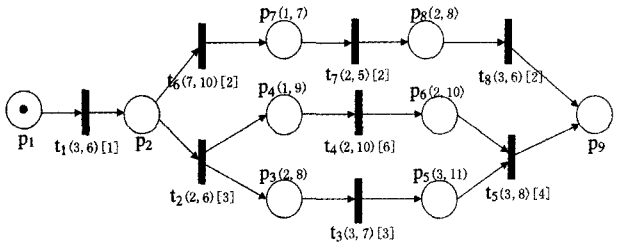

图 3　变迁序列可调度性分析

在图 3 中，变迁序列 $\delta=(t_1t_2t_3t_4t_5)$ 的可调度性分析如下：明显，在现有时间约束下，t_1，t_2，t_3 以及 t_4 是强可调度的。变迁 t_5 的 Root_Time 为 T_2，T_2 表示变迁 t_2 的触发结束时间，库所 p_5 以及 p_6 的托肯到达时间分别可以表示为 $[T_2+6,\ T_2+7]$，$[T_2+8,T_2+9]$。通过拥有多个输入库所变迁的可调度分析，可得变迁 t_5 也是强可调度的，因此基于现有时间约束，δ 是可调度的，则标识 $M_n=\{p_9\}$ 为可达。假设 T_0 为整个过程的开始时间，对于每一个变迁 t_i，可以很容易地得到以 T_0 表示的 $FIRE_{end}(t_i)$。活动序列 δ 的结束时间：$TOKEN_{arr}(p_9)$ 为 $[T_0+24,T_0+29]$。意味着如果 δ 能够顺利完成它的执行，则它不可能在 T_0+24 时刻前完成，也不可能在 T_0+29 时刻之后。

4　可调度性的组合分析

在这部分中，我们描述了如何将一个触发序列分解为一系列子序列来进行可调度分析。我们使用 $EN(M)$ 来表示在标识 M 下使能的变迁集合。根据文[2]，可以得到定义 6：

定义 6　令 $\sigma1=(M_{10}t_{11}M_{11}\cdots t_{1i}M_{1i}\cdots t_{1m}M_{1m})(m\geqslant1)$ 以及 $\sigma_2=(M_{20}t_{21}M_{21}\cdots t_{2j}M_{2j}\cdots t_{2n}M_{2n})(n\geqslant1)$ 为基网 UN 中的两个序列，M_{10} 以及 M_{20} 是从标识 M_0 可达的。σ_2 同 σ_1 是可组合的当且仅当 $M_{1m}=M_{20}$ 且 $EN(M_{1m})\cap EN(M_{1m-1})-\{t_{1m}\}=\varnothing$。$\sigma_2$ 同 σ_1 的组合，表示为 $\sigma_1+\sigma_2$，是

$(M_{10}t_{11}M_{11}\cdots t_{1i}M_{1i}\cdots t_{1m}M_{1m}t_{21}M_{21}\cdots t_{2j}M_{2j}\cdots t_{2n}M_{2n})$

举个例子，在图 3 中，$(M_2t_3M_3t_4M_4t_5M_5)$ 与 $(M_0t_1M_1t_2M_2)$ 可组合，因为 $EN(M_2)=\{t_3,t_4\}$，$EN(M_1)=\{t_2,t_6\}$，而且 $EN(M_1)\cap EN(M_2)=\varnothing$；但是 $(M_3t_4M_4t_5M_5)$ 不能和 $(M_0t_1M_1t_2M_2t_3M_3)$ 进行组合操作，因为 $EN(M_3)=\{t_4\}$，$EN(M_2)=\{t_3\ t_4\}$ 并且 $EN(M_2)\cap EN(M_3)\neq\varnothing$，所以可知 $\{t_1t_2t_3t_4t_5\}$ 不能分解为 $\{t_1t_2t_3\}$ 和 $\{t_4t_5\}$。很明显，当考虑分解一个序列的时候，不能将并发的变迁分开。

定理 3　设 σ_2 同 σ_1 可组合。$\delta_1\delta_2$ 是可调度的当且仅当 δ_1 以及 δ_2 都是可调度的。

证明：令 $\sigma_1=(M_{10}t_{11}M_{11}\cdots t_{1i}M_{1i}\cdots t_{1m}M_{1m})$ $\sigma_2=(M_{20}t_{21}M_{21}\cdots t_{2j}M_{2j}\cdots t_{2n}M_{2n})$

$\delta_1=(t_{11}\cdots t_{1i}\cdots t_{1m})$　$\delta_2=(t_{21}\cdots t_{2j}\cdots t_{2n})$

$\delta_1\delta_2=(t_{11}\cdots t_{1i}\cdots t_{1m}t_{21}\cdots t_{2j}\cdots t_{2n})$

$AD_i=[EFBT(t_i),LFET(t_i)]$

1. 假设 δ_1 同 δ_2 都是可调度的。存在两个 AD 序列用来证明 δ_1 以及 δ_2 的可调度性，为 $(AD_{11}\cdots AD_{1i}\cdots AD_{1m})$，$(AD_{21}\cdots AD_{2j}\cdots AD_{2n})$，并且对于每一个 AD_i，存在 $LFET(t_i)-EFBT(t_i)\geqslant FIRE_{dur}(t_i)$，因为 σ_2 同 σ_1 是可组合的，$M_{1m}=M_{20}$，$\delta_1\delta_2=(t_{11}\cdots t_{1i}\cdots t_{1m}t_{21}\cdots t_{2j}\cdots t_{2n})$，所以 $(AD_{11}\cdots AD_{1i}\cdots AD_{1m}AD_{21}\cdots AD_{2j}\cdots AD_{2n})$ 可以表示为检查序列 $\delta_1\delta_2$ 可调度性的 AD 序列，因此 $\delta_1\delta_2$ 是可调度的。

2. 假设 $\delta_1\delta_2$ 是可调度的。存在一个 AD 序列来检查 $\delta_1\delta_2$ 的可调度性，为，$(AD_{11}\cdots AD_{1i}\cdots AD_{1m}AD_{21}\cdots AD_{2j}\cdots AD_{2n})$。明显，$(AD_{11}\cdots AD_{1i}\cdots AD_{1m})$ 以及 $(AD_{21}\cdots AD_{2j}\cdots AD_{2n})$ 分别为 δ_1 以及 δ_2 的 AD 序列，所以 δ_1 以及 δ_2 都是可调度的。

定理 4　令 $\sigma_i(1\leqslant i\leqslant k)$ 为一系列序列，并且 $\sigma_i(2\leqslant i\leqslant k)$ 同 σ_{i-1} 可组合。$\delta_1\cdots\delta_k$ 是可调度的当且仅当 $\delta_i(1\leqslant i\leqslant k)$ 都是可调度的。

证明：当 $k=2$ 时很明显可得结论；如果 $k=3$，则 $\delta_1\delta_2\delta_3=(\delta_1\delta_2)\delta_3$。令 $\delta=\delta_1\delta_2$，$\delta_1\delta_2\delta_3=\delta\delta_3$。$\delta\delta_3$ 是可调度的 iff δ 以及 δ_3 是可调度的 iff δ_1，δ_2，以及 δ_3 是可调度的。设 $\delta=\delta_1\cdots\delta_{k-1}$ 为可调度的 iff $\delta_i(1\leqslant i\leqslant k-1)$ 均为可调度的。$\delta_1\cdots\delta_k=\delta\delta_k$。$\delta_1\cdots\delta_k$ 是可调度的 iff δ 以及 δ_k 是可调度的 iff $\delta_i(1\leqslant i\leqslant k)$ 均为可调度的。

定理 5　假设序列 σ_2 为自组合的，序列 σ_1 序列 σ_3 同 σ_2 是可组合的。令 $\delta=(\delta_2)^k=\delta_2\cdots\delta_2\cdots\delta_2$，$\delta_2$ 的个数为 $k(k>0)$。$\delta_1\delta\delta_3$ 是可调度的当且仅当 $\delta_1\delta_2\delta_3$ 都是可调度的。

证明：如果 $\sigma_2=(M_{20}t_{21}M_{21}\cdots t_{2i}M_{2i}\cdots t_{2m}M_{2m})$ 是一个自组合序列，则 $M_{20}=M_{2m}$。对于包含在 δ 中的每个变迁 t_i，随着 δ_2 执行次数的增加 t_i 的可调度性并不发生改变。因此，$(\delta_2)^k$ 的可调度性同 δ_2 的可调度性一致。根据定理 4，我们可以很容易得到此定理。

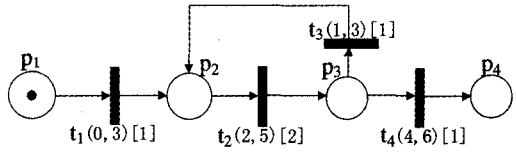

图 4　含有循环结构变迁序列

定理 4 以及定理 5 能够简化包含循环结构的序列的可调度分析。如图 4 所示，存在触发序列 $\sigma=(M_0t_1M_1t_2M_2)(\sigma_1)^k(M_2t_4M_3)$ 能够将初始标识 $M_0=\{p_1\}$ 转化为最终标识 $M_3=\{p_4\}$，$\sigma_1=(M_2t_3M_1t_2M_2)$，$M_1=\{p_2\}$，$M_2=\{p_3\}$，且 $k\in N$。对应于触发序列 σ 的变迁序列为 $\delta=t_1t_2(t_3t_2)^kt_4$。根据定理 5，$t_1t_2(t_3t_2)^{100}t_4$ 可调度当且仅当 $t_1t_2(t_3t_2)t_4$ 是可调度的。这里，基于附加的时间约束，变迁序列 $t_1t_2(t_3t_2)t_4$ 是可调度的。为决定标识 M_3 的可达性，我们只需要判断基本序列的可调度性即可，例如，$\delta_1=(t_1t_2t_4)(k=0)$ 以及 $\delta_2=t_1t_2(t_3t_2)t_4(k=1)$。因此，标识 M_3 是可达的，因为 δ_1 以及 δ_2 是可调度的。

定理 4 以及定理 5 可以简化包含循环的序列的可调度分析。需要注意的是定理 3～5 只有当序列是可以分解的情况下才是有用的。

5　时序一致性验证

为了确保模型的顺利执行，需要对模型中的时间约束进行必要的验证，检验其是否符合时序的满足性，并在模型开始

执行之前找出可能的问题，减少不必要的损失。一个活动的时序一致性直接依赖于活动的可调度性，一个 TCPNs 模型的时序一致性验证则依赖于模型的可调度性。这样时序一致性验证主要包含两方面的范畴：

对一个独立的活动 t_i 来讲，确保 $LFET(t_i)-EFBT(t_i)\geqslant FIRE_{dur}(t_i)$，则 t_i 是强可调度的。

·对整个 TCPN 模型来说，模型执行满足所有预设时间约束的前提条件且该模型中所有变迁在时间约束下都可调度。

定义 7 一个 TCPNs 模型中所有变迁基于附加时间约束都强可调度，则 TCPNs 模型一定能够顺利执行完成。

定理 6 一个 TCPN 模型是时序一致的当且仅当这个模型中所有活动基于现有时间约束都是可调度。

证明：在一个建模好的 TCPN 模型中，因为所有变迁基于附加时间约束都强可调度，因此对于每一个变迁 t_i，都有 $LFET(t_i)-EFBT(t_i)\geqslant FIRE_{dur}(t_i)$，这样每一个变迁也都存在决策域度。在事先期望的时间约束条件下，只要每个变迁在相应的可调度决策范围内开始执行，则整个 TCPN 模型一定能够顺利地完成执行工作。

总结与展望 本文的工作主要包括：

1. 扩展 TCPNs 的原始定义，对其变迁的强可调度性重新进行了定义，并给出单个变迁以及变迁序列的强可调度判定定理；

2. 变迁序列的组合分析策略以分析含有循环结构的复杂变迁序列；

3. 基于 TCPNs 可调度分析策略，提出时序一致性概念。

如何扩展组合分析策略来分析含有不规则结构的序列将是我们今后需要继续的工作。

致谢 在此，我们向对本文提出宝贵意见的评审专家表示感谢。

参 考 文 献

1 Tsai J J P, Yang S J, Chang Y H. Timing Constraint Petri Nets and Their Application to Schedulability Analysis of Real-Time System Specifications. IEEE Trans. Software Eng., 1995, 21(1): 32～49

2 Xu Dianxiang, He Xudong, Deng Yi. Compositional Schedulability Analysis of Real-Time Systems Using Time Petri Nets. IEEE Trans. Software Eng., 2002, 28(10): 984～995

3 Murata T. Petri nets: properties, analysis and application. Proc. IEEE, 1989, 77(8): 541～580

4 Van der Aalst W M P. The Application of Petri Nets to Workflow Management. Journal of Circuits, Systems, and Computers, 1998, 8(1): 21～66

5 Adam NR, Atluri V, Huang WK. Modeling and Analysis of Workflow Using Petri Nets. Journal of Intelligent Information Systems, 1998, 10(2): 131～158

6 Ramamoorthy C V, Ho G S. Performance evaluation of asynchronous concurrent systems using Petri nets. IEEE Trans. Software Eng., 1980, SE(6): 440～449

7 Ciardo G, German R, Lindemann C. A characterization of the stochastic process underlying a stochastic Petri net. IEEE Trans. Software Eng., 1994, 20(7): 506～515

8 Merlin P M, Farber D J. Recoverability of communication protocols implications of a theoretical study. IEEE Trans. Comm., 1976, 24(4): 1036～1043

（上接第 289 页）

则便无法正确解析要传染的可执行程序，从而也就无法实施传染操作。也就是说，掌握解密密钥是计算机病毒实施传播的前提，据此有如下结论：

定理 2 $k_c=(k_e,k_d)$ 是代码保护密钥，若能确保代码解密密钥 k_d 不被窃取或非法使用，则计算机病毒不可能在系统中传播。

如果计算机病毒掌握了解密密钥或可以非法使用解密密钥，那么它就能正确解密受保护的代码并把病毒代码传染给它。即便如此，如果它不掌握加密密钥或不能非法使用加密密钥，就无法把被传染后的文件重新加密。这样，根据规则 3，并由定理 1 知计算机病毒代码仍然不可能被正确植入系统。这个结论由如下定理描述：

定理 3 $k_c=(k_e,k_d)$ 是代码保护密钥，若能确保代码加密密钥 k_e 不被窃取或非法使用，则在规则 3 下，计算机病毒不可能在系统中传播。

逻辑隔离的主要脆弱性是一旦引用验证机制被旁路，那么它将彻底或局部失去恶意代码防御能力。下面的定理指出了基于密码隔离的恶意代码防御策略较之逻辑隔离的优越性。

定理 4 $k_c=(k_e,k_d)$ 是代码保护密钥，假设在初始时刻系统中不存在恶意代码，那么如果能够确保代码加、解密密钥 k_e、k_d 不被窃取或非法使用，且规则 3 总是起作用，那么系统将具有如下恶意代码防御能力：

(1)在任意时刻可执行程序都不会被恶意代码传染；

(2)在任意时刻主体都不会触发恶意代码；

(3)在任意时刻恶意代码无法在系统中传播。

由前面所述的各个定理和推论知定理 4 的结论是显然的。定理 4 结论成立，要求规则 3 总是起作用，即要求实现规则 3 的安全机制不能被旁路。实现这一点较之确保逻辑隔离的有效性要容易得多，因为只要在操作系统的程序装载器中增加一个解密模块即可实现。

结论 目前安全操作系统所实现的恶意代码防御策略从本质上说都是基于访问控制策略的逻辑隔离，这类防御策略具有脆弱性：首先，TCB 的引用验证机制存在被旁路的可能；其次，所实施的访问控制策略自身可能存在局限性，从而限制了其恶意代码防御能力。本文基于密码隔离机制建立了一个恶意代码免疫模型，它可以克服逻辑隔离的脆弱性。该模型定义了代码植入规则、代码保护规则和代码执行规则，证明了在系统初态安全且代码加、解密密钥安全的条件下，任何时刻系统中都不会有恶意代码的执行和传播，从而达到对恶意代码免疫的防御效果。

参 考 文 献

1 Cohen F. Computer Viruses: Theory and Experiments. Computers and Security, 1987, 6(1): 22～35

2 Biba K J. Integrity Considerations for Secure Computer Systems. NTIS AD-A039324, Electronic Systems Division, Air Force Systems Command, April 1977

3 赵庆松. 安全操作系统的恶意代码防御技术的研究和实施:[学位论文]. 北京:中国科学院软件研究所, 2002

4 杨涛. SUNIX 安全操作系统:[学位论文]. 长沙:国防科技大学, 1993

5 Department of Defense of USA. Trusted Computer System Evaluation Criteria. DoD 5200.28 - STD, Aug 1983

6 Schneier B. 应用密码学 - 协议，算法与 C 源程序. 北京:机械工业出版社, 2000