基于 Petri 网的工作流模型简化*)

周从华　刘志锋

(江苏大学计算机科学与通信工程学院　江苏镇江 212013)

摘　要　计算状态空间可达图是验证工作流正确性的主要方法,状态空间爆炸是这类方法的主要困难。文章对线性时态逻辑 LTL$_{-X}$描述的正确性提出了一种基于 Petri 网图形化简的验证方法,证明了所提出化简规则的完备性,并以实例说明了所提方法的有效性。

关键词　工作流网,Petri 网,正确性,线性时态逻辑

Reduction of Petri Net-based Workflow Model

ZHOU Cong-Hua　LIU Zhi-Feng

(School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013)

Abstract　The primary approach in verifying the workflow's correctness is to compute its reachable state space. The state explosion problem is the central difficulty in such verification technique. For the correctness expressed with the linear time temporal logic LTL-X a Petri net based verification approach the employs a set of graph reduction is proposed. The completeness of the proposed method is proved. And a case study is made to demonstrate our method.

Keywords　Workflow net, Petri net, Soundness, LTL

1　引言

工作流[1]是一类能够完全或者部分自动执行的业务过程,它根据一系列过程规则使得文档、信息或任务能够在不同的执行者之间传递或执行,以达到确定的业务目标。工作流技术是实现业务过程的建模、仿真分析、优化、管理与集成,从而最终实现业务过程自动化的核心技术。利用工作流技术进行建模和分析,可以规范业务过程,发现其中不合理的环节,进而对业务过程进行优化重组。然而,一个在定义过程中包含错误定义和描述的工作流,在实施过程中将会产生非常严重的后果,因此工作流的正确性对于业务处理过程是非常重要的。

Petri 网[2]是一种可用图形表示的模型,具有直观、易懂和易用的优点,对描述和分析并发现象有独到的优越之处;Petri 网又是严格定义的数学对象,借助数学开发的 Petri 网分析方法和技术,既可以用于静态分析,又可用于动态的行为分析。所以,Petri 网自然成为工作流过程建模的理想工具[3]。计算可达图是验证基于 Petri 网的工作流正确性的主要方法,然而由于工作流逻辑的复杂性,基于 Petri 网的工作流逻辑的正确性验证无论在时间上还是空间上代价都是非常高的,可达空间往往随着节点的增加呈指数级增长,因此状态空间爆炸是验证的主要困难所在。典型的基于 Petri 网的工作流逻辑具有 50～1000 个库所和变迁,如果 Petri 网是 k 界的,那么可达空间可能会达到 k^{1000} 个状态,目前存在的工具还无法处理具有这么多状态的工作流模型。因此,如何利用化简规则,在保持正确性的前提下,对模型进行规约从而降低库所和变迁的数量就显得尤为重要。

已有的方法[4~7]主要针对工作流模型的无死锁、无死任务和完整性进行化简。本文中我们针对更加一般化的正确性描述语言——线性时态逻辑 LTL$_{-X}$[8],提出一系列的化简规则,这些规则保持 LTL$_{-X}$属性不变。LTL$_{-X}$广泛应用于并发系统性质的描述,如死锁、安全性、活性、事件之间的时序性质。工作流模型包含并发机制,可以理解为一种特殊的并发系统,因此 LTL$_{-X}$也适用于工作流模型正确性的描述。本文第 2 节介绍了基于 Petri 网的工作流模型;第 3 节介绍了 LTL$_{-X}$的 Kripke 语义,并将 LTL$_{-X}$的语义扩展到工作流网,讨论了正确性需求的 LTL$_{-X}$描述;第 4 节给出了 10 条化简规则,证明了规则对 LTL$_{-X}$的保持特性,并以实例进行了说明。

2　基于 Petri 网的工作流模型

定义 2.1　一个 Petri 网是一个四元组 $PN=(P,T,F,M_0)$,其中

(1)P 表示库所节点集合,T 表示变迁节点集合;

(2)$P\cap T=\varnothing$,$P\cup T\neq\varnothing$;

(3)$F\subseteq(P\times T)\cup(T\times P)$,表示库所节点与变迁节点之间的有向弧集合;

(4)$M_0:P\to N$ 为初始标识,这里为 N 自然数集。

对 $\forall x\in P\cup T$,令 $\cdot x=\{y\mid y\in P\cup T)\wedge((y,x)\in F)\}$ 和 $x\cdot=\{y\mid(y\in P\cup T)\wedge((x,y)\in F)\}$,称 $\cdot x$ 和 $x\cdot$ 分别为 x 的前置集和后置集。称托肯在库所节点上的分布为 Petri 网的标识,即 $M:P\to N$。本文,采用 $M(p_i)$ 表示在库所节点 p_i 上的托肯数。变迁 t 在标识 M 下是使能的当且仅当 $\forall p\in\cdot t,M(p)\geqslant 1$,使能的变迁 t 是可以引发的,引发以后 Petri 网的标识发生如下变化:

$$M'(p)=\begin{cases}M(p)-1 & :p\in\cdot t-t\cdot\\ M(p)+1 & :p\in t\cdot-\cdot t\\ M(p) & :\text{otherwise}\end{cases}\qquad(1)$$

*)本文得到国家自然科学基金(No.60603041)、江苏省自然科学基金(BK2006073)的资助。周从华　博士,讲师,主要研究领域为模型检测、软件可靠性、工作流。

给定 Petri 网 $PN=(P,T,F,M_0)$，在文章中会用到下面几个标记：

• $M_1 \xrightarrow{t} M_2$：变迁 t 在标识 M_1 是使能的，且在引发后标识由 M_1 变成 M_2；

• $M_1 \rightarrow M_2$：存在变迁 t 使得 $M_1 \xrightarrow{t} M_2$；

• $M_1 \xrightarrow{\sigma} M_2$：变迁序列 $\sigma=t_1,\cdots,t_{n-1}$ 使得 $M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} \cdots \xrightarrow{t_{n-1}} M_n$；

• $M_1 \xrightarrow{*} M_2$：存在变迁序列 σ 使得 $M_1 \xrightarrow{\sigma} M_2$；

• $[M_0\rangle$ 表示 PN 的可达标识集：$[M_0\rangle=\{M|M_0 \xrightarrow{*} M\}$。

我们称 Petri 网 PN 是 k 界的当且仅当 $\forall M\in[M_0\rangle, \forall p\in P(M(p)\leqslant k)$。文中我们假设所讨论的 Petri 网 PN 均是 1 界的。

定义 2.2 一个 Petri 网是强连通的当且仅当对任意两个节点 x 和 y，存在一条从 x 到 y 的路径。

由于 Petri 网具有坚实的理论基础和易于使用的图形表示，因而是一种理想的工作流建模工具。

定义 2.3 一个 Petri 网 $PN=(P,T,F,M_0)$ 是一个工作流网 WF-net[1] 当且仅当：

(1)PN 有两个特殊的库所节点：i 和 o。i 是起始库所，即：$\cdot i=\varnothing$，o 是终止库所，即：$o\cdot=\varnothing$。

(2)如果在 PN 中加入一个新的变迁 t^*，使 t^* 连接库所 i 和 o，即：$\cdot t^*=\{o\}, t^*\cdot=\{i\}$，这时所得到的 PN 是强连通的。

3 线性时态逻辑 LTL$_{-X}$

线性时态逻辑采用线性、离散且与自然数同构的时间结构，以路径(状态序列)作为命题的论断对象，线性时态逻辑公式是在状态序列上解释其真值的。

定义 3.1(Kripke 结构) 设 AP 表示一个原子命题集合，一个基于 AP 的 Kripke 结构 K 是如下的一个四元组 $K=(S,s_0,R,L)$ 且满足下面条件：

(1)S 是有限状态集合；

(2)$s_0\in S$ 是初始状态；

(3)$R\subseteq S\times S$ 是一个完全的转移关系，也就是说对于任意状态 $s\in S$，存在状态 $s'\in S$ 使得 $R(s,s')$ 成立；

(4)$L:S\rightarrow 2^{AP}$ 是一个标记函数，用于标记该状态下取真值的原子命题集。

在 Kripke 结构中，一条从初始状态 s_0 出发的的路径 π 是一个无限序列 $s_0,s_1,\cdots$，且对于每个 $i\geqslant 0, R(s_i,s_{i+1})$ 成立。我们用 π^i 来表示从状态 s_i 开始的路径，即 $\pi^i=s_i,s_{i+1},\cdots,\pi(i)$ 表示路径 π 上的第 i 个状态。线性时态逻辑 LTL$_{-X}$ 由时态算子 X,F,G,U,R、原子命题以及逻辑联结词 $\neg,\wedge,\vee$ 构成，定义如下：

• 如果 $p\in AP$，那么 p 是 LTL$_{-X}$公式；

• 如果 f 和 g 是 LTL 公式，那么 $\neg f$、$f\vee g$、$f\wedge g$ 是 LTL$_{-X}$公式；

• 如果 f 和 g 是 LTL 公式，那么 Ff、Gf、$f\cup g$、fRg 是 LTL$_{-X}$公式。

我们在 Kripke 结构上定义 LTL$_{-X}$的语义。$K,\pi\vdash f$ 表示 f 在路径 π 上成立，这里关系 $\vdash$ 定义如下(g_1、g_2 是 LTL$_{-X}$公式)：

• $K,\pi\vdash p$ 当且仅当 $p\in L(\pi(0))$；

• $K,\pi\vdash\neg g_1$ 当且仅当 $K,\pi\nvdash g_1$；

• $K,\pi\vdash g_1\vee g_2$ 当且仅当 $K,\pi\vdash g_1$ 或者 $K,\pi\vdash g_2$；

• $K,\pi\vdash g_1\wedge g_2$ 当且仅当 $K,\pi\vdash g_1$ 且 $K,\pi\vdash g_2$；

• $K,\pi\vdash Fg_1$ 当且仅当存在 $k\geqslant 0$ 使得 $K,\pi^k\vdash g_1$；

• $K,\pi\vdash Gg_1$ 当且仅当对任一 $i\geqslant 0, K,\pi^i\vdash g_1$；

• $K,\pi\vdash g_1Ug_2$ 当且仅当存在 $k\geqslant 0$ 使得 $K,\pi^k=g_2$ 且对于任一 $0\leqslant j<k, K,\pi^j\vdash g_1$；

• $K,\pi\vdash g_1Rg_2$ 当且仅当对任一 $j\geqslant 0$，如果对每个 $i<j$，$K,\pi^j\nvdash g_1$ 那么 $K,\pi^j\vdash g_2$；

我们称 LTL$_{-X}$公式 f 在 Kripke 结构 K 中是有效的当且仅当对从初始状态 s_0 开始的路径 π，均有 $K,\pi\vdash f$。Kripke 结构显式地表达了系统状态动态变化的过程，而 Petri 网则需要通过变迁的引发反映系统状态的动态变化。因此对以 Petri 网描述的系统模型，可通过变迁的引发得到描述该系统的 Kripke 结构。

设 $PN=(P,T,F,M_0)$，$[M_0\rangle$ 为 PN 的可达空间，PN 对应的 Kripke 结构 K 定义如下：

(1)$AP\{a_1,\cdots,a_m\}$，m 为 PN 中库所节点的数目；

(2)$s=[M_0\rangle$；

(3)$s_0=M_0$；

(4)$(M,M')\in R$ 当且仅当 $M\rightarrow M'$；或者 $M=M'$ 且不存在变迁 $t\in T$ 在 M 下是使能的；

(5)$a_i\in L(M)$ 当且仅当 $M(p_i)=1$，这里 $1\leqslant i\leqslant m$。

我们称 LTL$_{-X}$公式 f 在 PN 中是有效的当且仅当 f 在 PN 对应的 Kripke 结构 K 中是有效的。

3.1 Stuttering 等价

Stuttering 等价的目的在于简化并发异步系统的状态空间，其基本思想是对于需要检测的属性异步进程的不同执行顺序是没有区分度的。这样只需考虑具有代表性的执行顺序，从而简化系统的状态空间。

定义 3.2(stuttering 等价[8]) 两条无穷的路径 $\pi=s_0,s_1,\cdots,\pi'=s'_0,s'_1,\cdots$，是 stuttering 等价的当且仅当存在两个无穷的自然数序列：$0=i_1<i_2<\cdots, 0=j_1<j_2<\cdots$，使得对 $\forall k\geqslant 0, L(\pi(i_k))=L(\pi(i_k+1))=\cdots=L(\pi(i_{k+1}-1))=L(\pi'(j_k))=L(\pi'j_k+1))=\cdots=L(\pi'(j_{k+1}-1))$。

本质上，stuttering 等价将路径分成无穷个状态序列。另外 stuttering 等价具有传递性，即 π 与 π' stuttering 等价，π 与 π''stuttering 等价，可知 π' 与 π''stuttering 等价。下面的定理表明 stuttering 等价保持 LTL$_{-X}$属性不变。

定理 1[8] 设 π,π' 是 stuttering 等价的两条路径，f 为 LTL$_{-X}$公式，则 $\pi\vdash f$ 当且仅当 $\pi'\vdash$ f。

3.2 正确性需求的 LTL$_{-X}$描述

这一小节我们探讨 LTL$_{-X}$如何精确地描述正确性需求。表 1 总结了一般正确性需求的 LTL$_{-X}$描述。

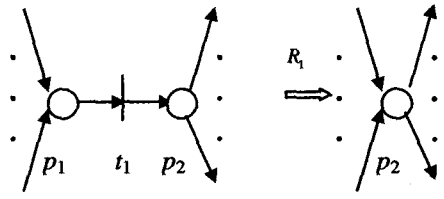
4 模型化简

基于工作流网的化简规则应当遵循 LTL$_{-X}$属性不变原则。我们给出 10 条规则 R_1-R_{10}，并且依据 LTL$_{-X}$属性保持原则，化简规则只对起始库所，终止库所之外的同时又是 LTL$_{-X}$属性没有涉及到的节点化简。对任意 LTL$_{-X}$属性 f，记号 $Atom(f)$ 表示出现在 f 中的原子命题。对每个化简规则 R_i 记号 $Atom(R_i)$ 表示 R_i 涉及到的库所节点。

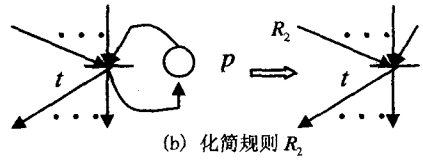
设 PN 为工作流网，化简前 $PN=(P,T,F,M_0)$，利用化简规则 R 化简后为 $PN'=(P',T',F',M'_0)$，记为 $PN \xrightarrow{R} PN'$。对化简规则 R，PN，PN' 对应的 Kripke 结构中的标记函数的定义域均为 $P\backslash Atom(R)$。

表 1　正确性需求的 LTL$_{-X}$描述

名称	LTL$_{-X}$描述	说明
不变式	$G\alpha$	刻画工作流生命周期中不变的状态
没有发生	$G(\neg \cdot t)$	刻画在工作流网中变迁 t 不会引发
最终发生	$F(\cdot t)$	刻画在工作流网中变迁 t 会被引发
Until	$\alpha U\beta$	某变迁引发使 α 为真直到另一变迁引发使 β 为真
因果	$G(\cdot t_1 \rightarrow F \cdot t_2)$	变迁 t_1 的引发会导致变迁 t_2 的引发
单因多果	$G(\cdot t_1 \rightarrow (F \cdot t_2 \wedge F \cdot t_3))$	变迁 t_1 的引发会导致变迁 t_2，t_3 的引发



(a) 化简规则 R_1



(b) 化简规则 R_2

图 1

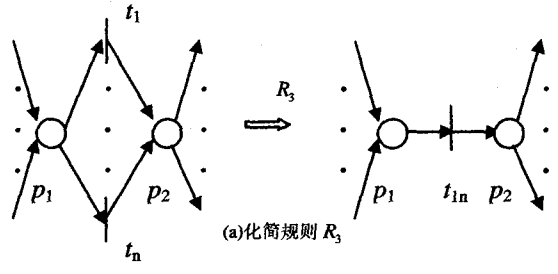
化简规则 R_1：如图 1(a)，$\exists t_1 \in T, \exists p_1, p_2 \in P(p_1 \cdot = \{t_1\} \wedge \cdot t_1 = \{p_1\} \wedge t_1 \cdot = \{p_2\} \wedge \cdot p_2 = \{t_1\}) \Rightarrow (P' = P \setminus \{p_1\}, T' = T \setminus \{t_1\}, F' = F \cup \{(t, p_2) \mid t \in \cdot p_1\} \setminus \{(p_1, t_1), (t_1, p_2)\} \setminus \{(t, p_1) \mid t \in \cdot p_1\}, \forall p \in P \setminus \{p_1\}, M'_0(p) = M_0(p))$，这里 $Atom(R_1) = \{p_1, p_2\}$。

化简规则 R_2：如图 1(b)，$\exists t \in T, \exists p \in P(\cdot p = p \cdot = \{t\} \wedge \{p\} \subset \cdot t \Rightarrow (P' = P \setminus \{p\}, T' = T, F' = F \setminus \{(p, t), (t, p)\}, \forall p \in P \setminus \{p\}, M'_0(p) = M_0(p)))$，这里 $Atom(R_2) = \{p\}$。

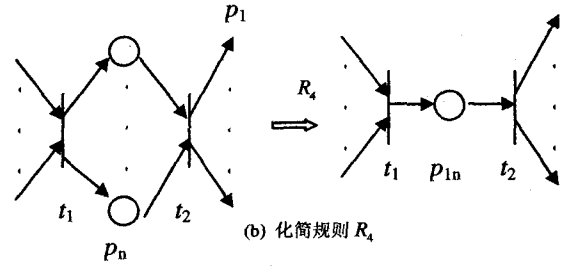
化简规则 R_3：如图 2(a)，$\exists t_1, \cdots, t_n \in T, \exists p_1, p_2 \in P(p_1 \cdot = \{t_1, \cdots, t_n\} \wedge \bigwedge_{i=1}^{n} (\cdot t_i = \{p_1\} \wedge t_i \cdot = \{p_2\}) \cdot p_2 = \{t_1, \cdots, t_n\}) \Rightarrow P' = P, T' = (T \setminus \{t_1, \cdots, t_n\}) \cup \{t_{1n}\}), F' = F \setminus (\bigcup_{i=1}^{n} \{(p_1, t_i\} \cup \bigcup_{i=1}^{n} \{(t_i, p_2)\})) \cup \{(p_1, t_{1n}), (t_{1n}, p_2)\}, M'_0 = M_0$，这里 $Atom(R_3) = \{p_1, p_2\}$。

化简规则 R_4：如图 2(b)，$\exists t_1, t_2 \in T, \exists p_1, \cdots, p_n \in P(t_1 \cdot = \{p_1, \cdots, p_n\} \wedge \cdot t_2 = \{p_1, \cdots, p_n\} \wedge \bigwedge_{i=1}^{n} (\cdot p_i = \{t_1\} \wedge p_i \cdot = \{t_2\})) \Rightarrow P'(P \setminus \{p_1, \cdots, p_n\}) \cup \{p_{1n}\}, T' = T, F' = F \setminus (\bigcup_{i=1}^{n} \{(t_1, p_i)\} \cup \bigcup_{i=1}^{n} \{(p_i, t_2)\}) \cup \{(t_1, p_{1n}), (p_{1n}, t_2)\}, \forall p \in P \setminus \{p_1, \cdots, p_n\} (M'_0(p) = M_0(p)), M'_0(p_{1n}) = 0$，这里 $Atom(R_4) = \{p_1, \cdots, p_n\}$。

化简规则 R_5：如图 3(a)，$\exists t_1, t_2 \in T, \exists p_1, p_2, p_3, p_4 \in P(p_1 \cdot = \{t_1, t_2\} \wedge p_3 \cdot = \{t_1, t_2\} \wedge \cdot t_1 = \{p_1, p_3\} \wedge \cdot t_2 = \{p_1, p_3\} \wedge t_1 \cdot = \{p_2, p_4\} \wedge t_2 = \{p_2, p_4\} \wedge \cdot p_2 = \{t_1, t_2\} \wedge \cdot p_4 = \{t_1, t_2\} \Rightarrow P = P', T' = (T \setminus \{t_1, t_2\}) \cup \{t_{12}\}, F' = (F \setminus \{(p_1, t_1), (p_1, t_2), (p_3, t_1), (p_3, t_2), (t_1, p_2), (t_1, p_4), (t_3, p_2), (t_3, p_4)\}) \cup \{(p_1, t_{12}), (p_3, t_{12}), (t_{12}, p_2), (t_{12}, p_4)\}, M'_0 = M_0$ 这里 $Atom(R_5) = \{p_1, p_2, p_3, p_4\}$。



(a)化简规则 R_3



(b) 化简规则 R_4

图 2

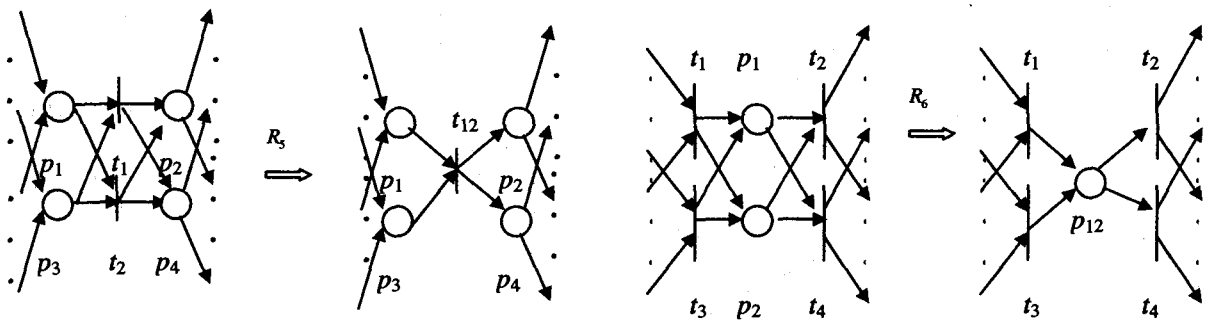
化简规则 R_6：如图 3(b)，$\exists t_1, t_2, t_3, t_4 \in T, \exists p_1, p_2 \in P(t_1 \cdot = \{p_1, p_2\} \wedge t_3 \cdot = \{p_1, p_2\} \wedge \cdot p_1 = \{t_1, t_3\} \wedge \cdot p_2 = \{t_1, t_3\} \wedge p_1 \cdot = \{t_2, t_4\} \wedge p_2 \cdot = \{t_2, t_4\} \wedge \cdot t_2 = \{p_1, p_2\} \wedge \cdot t_4 = \{p_1, p_2\}) \Rightarrow T' = T, P' = (P \setminus \{p_1, p_2\}) \cup \{p_{12}\}, F' = (F \setminus \{(t_1, p_1), (t_1, p_2), (t_3, p_1), (t_3, p_2), (p_1, t_2), (p_1, t_4), (p_2, t_2), (p_2, t_4)\}) \cup \{(t_1, p_{12}), (t_3, p_{12}), (p_{12}, t_2), (p_{12}, t_4)\}, \forall p \in P \setminus \{p_1, p_2, \} (M'_0(p) = M_0(p)), M'_0(p_{12}) = 0$，这里 $Atom(R_6) = \{p_1, p_2\}$。

化简规则 R_7：如图 4(a)，$\exists t_1, t_2, t_3 \in T, \exists p_1, p_2 \in P(\{t_1\} \subseteq p_1 \cdot \wedge \cdot t_1 = \{p_1\} \wedge t_1 \cdot = \{p_2\} \wedge \cdot p_2 = \{t_1\} \wedge \cdot p_2 = \{t_2, t_3\} \wedge t_2 \cdot = \{p_2\} \wedge \cdot t_3 = \{p_2\}) \Rightarrow P' = P / \{p_2\}, T' = T \setminus \{t_1\}, F' = (F \setminus \{(p_1, t_1), (t_1, p_2), (p_1, t_2), (p_2, t_3)\}) \cup \{(p_1, t_2), (p_1, t_3)\}, \forall p \in P \setminus \{p_2\} (M'_0(p) = M_0(p))$ 这里 $Atom(R_7) = \{p_1, p_2\}$。

化简规则 R_8：如图 4(b)，$\exists t_1, t_2, t_3 \in T, \exists p_1, p_2, p_3 \in P(t_1 \cdot = \{p_1\} \wedge t_2 \cdot = \{p_1\} \wedge \cdot p_1 = \{t_1, t_2\} \wedge p_2 \cdot = \{t_3\} \wedge \cdot t_3 = \{p_1\} \wedge t_3 \cdot \{p_2, p_3\} \wedge \cdot p_2 = \{t_3\} \wedge \cdot p_3 = \{t_3\}) \Rightarrow T' = T \setminus \{t_3\}, P' = P \setminus \{p_1\}, F \setminus \{(t_1, p_1), (t_2, p_1), (p_1, t_3), (t_3, p_2), (t_3, p_3)\}) \cup \{(t_1, p_2), (t_1, p_3), (t_2, p_2), (t_2, p_3)\}, \forall p \in P \setminus \{p_1\} (M'_0(p) = M_0(p))$，这里 $Atom(R_8) = \{p_1, p_2, p_3\}$。

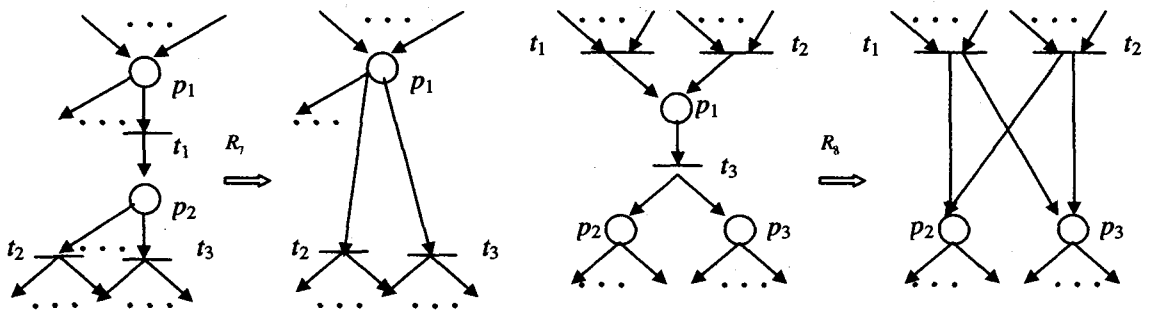
化简规则 R_9：如图 5(a)，$\exists t_1, t_2, t_3 \in T, \exists p_1, p_2, p_3 \in P(p_1 \cdot = \{t_1\} \wedge p_2 \cdot = \{t_1\} \wedge \cdot t_1 = \{p_1, p_2\} \wedge t_1 \cdot = \{p_3\} \wedge p_3 \cdot = \{t_2, t_3\} \wedge \cdot t_2 = \{p_3\} \wedge \cdot t_3 = \{p_2\}) \Rightarrow T' = T \setminus \{t_1\}, P' = P \setminus \{p_3\}, F' = (F \setminus \{(p_1, t_1), (p_2, t_1), (t_1, p_3), (p_3, t_2), (p_3, t_3)\}) \cup \{(p_1, t_2), (p_1, t_3), (p_2, t_2), (p_2, t_3)\}, \forall p \in P \setminus \{p_3\} (M'_0(p) = M_0(p))$，这里 $Atom(R_9) = \{p_1, p_2, p_3\}$。

化简规则 R_{10}：如图 5(b)，$\exists t_1, t_2 \in T, \exists p_1 \in P(t_1 \cdot = \{p_1\} \wedge \cdot p_1 = \{t_1\} \wedge p_1 \cdot = \{t_2\} \wedge \cdot t_2 = \{p_1\}) \Rightarrow (P' = P \setminus \{p_1\}, T' = T \setminus \{t_1\}, F' = F \cup \{(p, t_2) \mid p \in \cdot t_1\} \mid p \in \cdot t_1\} \setminus \{(t_1, p_1), (p_1, t_2)\} \setminus \{(p, t_1) \mid p \in \cdot t_1\}, \forall p \in P \setminus \{p_1\} (M'_0(p) = M_0(p))$，这里 $Atom(R_2) = \{p_1\} \cup \cdot t_1 \cup t_2 \cdot$。



(a) 化简规则 R_5　　(b) 化简规则 R_6

图 3

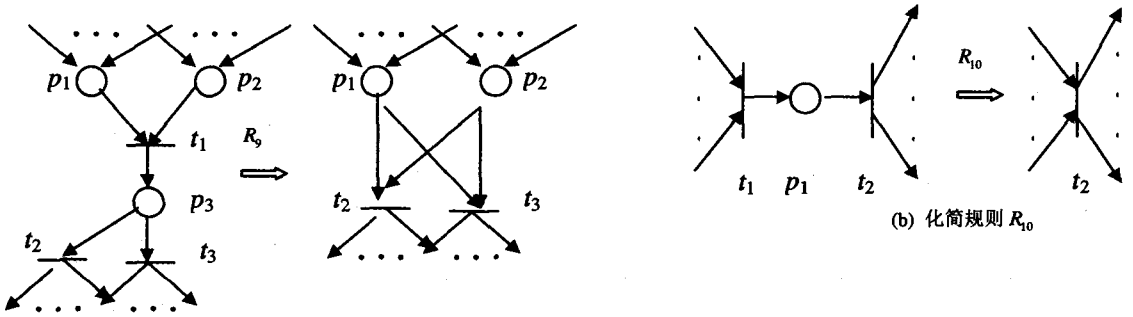


(a) 化简规则 R_7　　(b) 化简规则 R_8

图 4

定理 2(化简规则 R_1-R_9 保持 LTL$_{-X}$属性不变)　设 $PN=(P,T,F,M_0)$为一界工作流网，f 为 LTL$_{-X}$属性，$PN \xrightarrow{R_i} PN_i(1\leqslant i\leqslant 9)$。对任一化简规则 $R_i(1\leqslant i\leqslant 9)$，如果 $Atom(f)\cap Atom(R_i)=\varnothing$，则 f 在 PN 中是有效的当且仅当 f 在 PN_i 中是有效的。

证明：我们以化简规则 R_1 为例，说明规则保持 LTL$_{-X}$属性不变，其它规则的证明可以参考 R_1。设 $\pi=M_0\rightarrow M_1\rightarrow\cdots\rightarrow M_n\rightarrow M_{n+1}\rightarrow M_{n+2}\rightarrow\cdots$为 PN 中的一条路径，使用化简规则 R_1 后变为 π'。将每个标识 M_i 分为正交的二个子标识：${}_1M_i,{}_2M_i$，这里${}_1M_i$ 的定义域为 $Atom(R_1)$且 $\forall p\in Aotm(R_1)$，${}_1M_i(p)={}_1M_i(p)$，${}_2M_i$ 的定义域为 $P\setminus Atom(R_1)$且 $\forall p\in P\setminus Atom(R_1)$，${}_2M_i(p)=M_i(p)$。这样路径 π 可以分为两条路径：${}_1\pi={}_1M_0\rightarrow{}_1M_1\rightarrow\cdots\rightarrow{}_1M_n\rightarrow{}_1M_{n+1}\rightarrow{}_1M_{n+2}\rightarrow\cdots$，${}_2\pi={}_2M_0\rightarrow{}_2M_1\rightarrow\cdots\rightarrow{}_2M_n\rightarrow{}_2M_{n+1}\rightarrow{}_2M_{n+2}\rightarrow\cdots$。因为，$Atom(f)\cap Atom(R_1)=\varnothing$，由定理 1，$\pi\vdash f$ 当且仅当${}_2\pi\vdash f$，所以只需考察在应用化简规则 R_1 后，${}_2\pi$ 的变动情况。不失一般性，假设变迁 t_1 在 π 中只引发了一次，设为 $M_n\xrightarrow{t_1}M_{n+1}$。$(\cdot t_1\cup t_1\cdot)\subseteq Atom(R_1)$，所以${}_2M_n={}_2M_{n+1}$。在使用化简规则 R_1 后，${}_2\pi$ 变为${}_2\pi'={}_2M_0\rightarrow{}_2M_1\rightarrow\cdots\rightarrow{}_2M_n\rightarrow{}_2M_{n+2}\rightarrow\cdots$，因为${}_2M_n={}_2M_{n+1}$，所以${}_2\pi$ 与${}_2\pi'$ stuttering 等价，从而${}_2\pi'\vdash f$，即 $\pi'\vdash f$。



(a) 化简规则 R_9　　(b) 化简规则 R_{10}

图 5

下面在证明化简规则 R_{10}保持 LTL$_{-X}$属性不变之前，引入一个引理。

引理 1　设 Petri 网 PN 包含图 5(b)中的结构，则对任一自然数 l，在变迁序列 $\sigma_1=t_{k_1},\cdots,t_{k_{n-1}},t_1,t_{k_n},\cdots,t_{k_{n+l}},t_2,t_{k_{n+l+1}},\cdots,\sigma_2=t_{k_1},\cdots,t_{k_{n-1}},t_1,t_2,t_{k_n},\cdots,t_{k_{n+l}},\cdots$，引发下产生的路径是 stuttering 等价的。

证明：通过对 t_1,t_2 被引发的间距进行归纳来完成证明。

(1)首先证明 t_1,t_2 被引发的间距为 1 时结论成立。设变迁序列 $t_{k_1},\cdots,t_{k_{n-1}},t_1,t_{k_n},t_2,\cdots$引发下产生的路径为 $\pi=M_{k_1}\rightarrow\cdots\rightarrow M_{k_{n-1}}\rightarrow M_1\rightarrow M_{k_n}\rightarrow M_2\rightarrow M_{k_{n+l}}\cdots$。因为 PN 是一界的，所以 $\cdot t_1\cap\cdot t_{k_n}=\varnothing$。又因为 $\cdot t_2\cap\cdot t_{k_n}=\varnothing$，所以在 M_{k_n} 既可以引发 t_{k_n}，也可以引发 t_2，且如果 $M_{k_n}\xrightarrow{t_{k_n},t_2}M'$，

$M_{k_n} \xrightarrow{t_2,t_{k_n}} M''$，则 $M'=M''$。因此在变迁序列 $t_{k_1},\cdots,t_{k_{n-l}},t_1,t_2,t_{k_n},\cdots$的引发下可产生路径 $\pi'=M_{k_1}\rightarrow\cdots\rightarrow M_{k_{n-1}}\rightarrow M_1\rightarrow M_{k_n}\rightarrow M'_{2}\rightarrow M'_{k_{n+1}}\rightarrow M_{k_{n+2}},\cdots$且 π,π' 是 stuttering 等价的。

(2)设在 t_1,t_2 被引发的间距为 l 时，结论成立。下面证明 t_1,t_2 被引发的间距为 $l+1$ 时，结论仍然成立。设在变迁序列 $t_{k_1},\cdots,t_{k_{n-1}},t_1,t_{k_n},\cdots,t_{k_{n+l}},t_{k_{n+l+1}},t_2,\cdots$引发下产生的路径为 $\pi=M_{k_1}\rightarrow\cdots\rightarrow M_{k_{n-1}}\rightarrow M_1\rightarrow M_{k_n}\rightarrow\cdots\rightarrow M_{K_{n+l+1}}\rightarrow M_2\rightarrow M_{k_{n+l+2}}\rightarrow M_{k_{n+l+3}}\cdots$。由 1 界 Petri 网的定义，$\cdot t_{k_{n+l+1}}\cap\cdot t_2=\varnothing$，所以在标识 $M_{k_{n+l+1}}$ 下变迁 $t_{k_{n+l+1}}$，t_2 都可以引发，且 $M_{k_n} \xrightarrow{t_{k_{n+l+1}},t_2} M'$，$M_{k_n} \xrightarrow{t_2,t_{k_{n+l+1}}} M''$，则 $M'=M''$。因此在变迁序列 $t_{k_1},\cdots,t_{k_{n-1}},t_1,t_{k_n},\cdots,t_{k_{n+l}},t_2,t_{k_{n+l+1}},\cdots$引发下，产生的路径为 $\pi'=M_{k_1}\rightarrow\cdots\rightarrow M_{k_{n-1}}\rightarrow M_1\rightarrow M_{k_n}\rightarrow\cdots\rightarrow M_{k_{n+l+1}}\rightarrow M'_{k_{n+l+2}}\rightarrow M''_2\rightarrow M_{k_{n+l+3}}\cdots$，且 π,π' stuttering 等价。由归纳假设得 $t_{k_1},\cdots,t_{k_{n-1}},t_1,t_2,t_{k_n},\cdots,t_{k_{n+l}},\cdots$引发下产生的路径分别为 π'' 与 π' stuttering 等价。再由 stuttering 等价的传递性，π,π'' stuttering 等价。

定理 3(化简规则 R_{10} 保持 LTL_{-X} 属性不变) 设 $PN=(P,T,F,M_0)$ 为一界工作流网，f 为 LTL_{-X} 属性，$PN \xrightarrow{R_{10}} PN_{10}$。如果 $Atom(f)\cap Atom(R_{10})=\varnothing$，则 f 在 PN 中是有效的当且仅当 f 在 PN_{10} 中是有效的。

证明：设 $\sigma_1=t_{k_1},\cdots,t_{k_{n-1}},t_1,t_{k_n},\cdots,t_{k_{n+l}},t_2,\cdots$ 为 PN 中的变迁序列，由引理 1 可知，σ_1 引发下的路径 π'' 与 $\sigma_2=t_{k_1},t_{k_{n-1}},t_1,t_2,t_{k_n},\cdots,t_{k_{n+l}},\cdots$引发下的路进 π stuttering 等价。因此我们在 PN 可以限制 t_1,t_2 的被引发顺序为：一旦 t_1 被引发，t_2 必须紧跟着被引发。设引发 σ_2 后得到的路径为 $\pi=M_{k_1}\rightarrow\cdots\rightarrow M_{k_n}\rightarrow M_1\rightarrow M_2\rightarrow M_{k_{n+1}}\rightarrow\cdots$，使用化简规则 R_{10} 后变为 π'。将每个标识 M_i 分为正交的二个子标识：${}_1M_{i,2}$，M_i，这里 ${}_1M_i$ 的定义域为 $Atom(R_{10})$ 且 $\forall p\in Atom(R_{10})$，${}_1M_i(p)=M_i(p)$，${}_2M_i$ 的定义域为 $P\setminus Atom(R_{10})$ 且 $\forall p\in P\setminus Atom(R_{10})$，${}_2M_i(p)=M_i(p)$。这样路径 π 可以分为两条路径：${}_1\pi={}_1M_{k_1}\rightarrow\cdots\rightarrow{}_1M_{k_n}\rightarrow{}_1M_1\rightarrow{}_1M_1\rightarrow{}_1M_2\rightarrow{}_1M_{k_{n+1}}\rightarrow\cdots$，${}_2\pi={}_2M_{k_1}\rightarrow\cdots\rightarrow{}_2M_{k_n}\rightarrow{}_2M_1\rightarrow{}_2M_2\rightarrow{}_2M_{k_{n+1}}\rightarrow\cdots$。因为，$Atom(f)\cap Atom(R_{10})=\varnothing$，由定理 1，$\pi\vdash f$ 当且仅当 ${}_2\pi\vdash f$，所以只需考察在应用化简规则 R_{10} 后，${}_2\pi$ 的变动情况。由引发的定义，${}_2\pi$ 变为，${}_2\pi'={}_2M_{k_1}\rightarrow\cdots\rightarrow{}_2M_{k_n}\rightarrow{}_2M_1\rightarrow{}_2M_{k_{n+1}}\rightarrow\cdots$，显然 ${}_2\pi$ 与 ${}_2\pi'$ stuttering 等价，从而 ${}_2\pi'\vdash f$，即 $\pi'\vdash f$。

实例分析

本小节以实例说明我们的工作流验证方法的有效性。图 6 是所验证的工作流模型，来源于文[4]。我们验证的属性为：变迁 t_2 一定在 t_1 引发以后才可以引发，用 LTL_{-X} 描述为 $f=(\neg p_2 U p_1)\vee G(\neg p_1)$。属性 f 只涉及到 t_1,t_2 的前置节点。多次应用化简规则 R_1 到 R_{10}，得到如图 7 所示的工作流模型，该模型和图 6 中的模型均满足属性 f，但是其规模远远小于图 6 中的模型。

结论 工作流模型的正确性分析是工作流技术中重要的研究课题，本文对基于 Petri 网的工作流模型提出进行图形化简的模型验证方法。提出了 10 条化简规则，并证明了这些化简规则保持 LTL_{-X} 描述的正确性不变，从而为验证大规模工作流系统的正确性提供了支持。

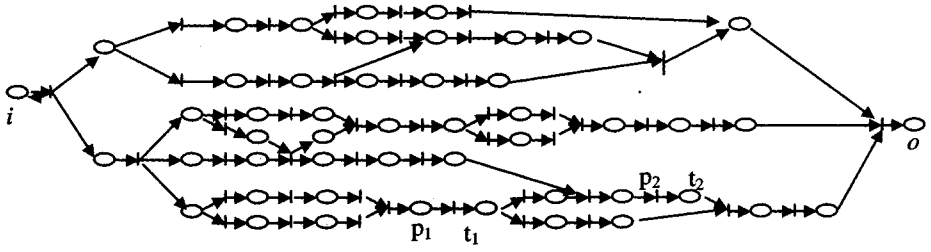


图 6 实例

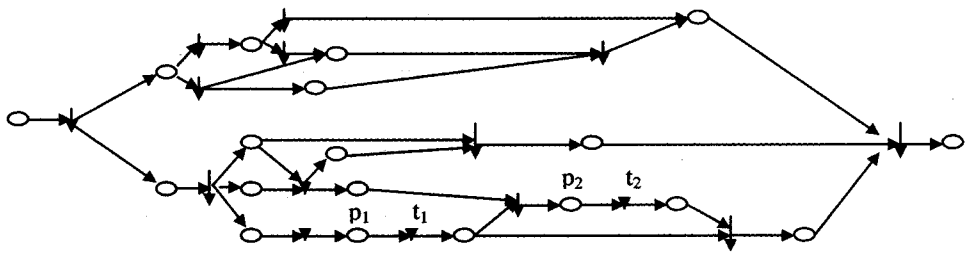


图 7 化简后的实例

参 考 文 献

1 范玉顺．工作流管理技术基础．北京：清华大学出版社，2001

2 袁崇义．Petri 网原理与应用．北京：电子工业出版社，2005

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

4 李建强，范玉顺．基于 Petri 化简方法的工作流模型验证．信息与控制，2001,30(6):492～497

5 周建涛，史美林，叶新铭．一种基于 Petri 网化简的工作流过程语义验证方法．软件学报，2005,16(7):1242～1251

6 杜淑楠，章宁，王鲁滨．工作流模型正确性验证过程中的模型化简问题

7 Sadiq W, Orlawska M E. Analyzing process models using graph reduction techniques. Information system, 2002, 25(2):117～134

8 Clarke E M, Grumberg O, Peled D A. Model Checking. MA: the MIT Press, 1999