软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(11):3278−3294 [doi: 10.13328/j.cnki.jos.005478]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种从无“*aba*”模式的日志中挖掘 2 度循环的方法*

林雷蕾[1,3], 周 华[2], 代 飞[2,3], 朱 锐[1,3], 李 彤[1,3]

[1](云南大学 软件学院,云南 昆明 650091)

[2](西南林业大学 大数据与智能工程学院,云南 昆明 650224)

[3](云南省软件工程重点实验室(云南大学),云南 昆明 650091)

通讯作者: 代飞, E-mail: daifei@swfu.edu.cn

摘 要: 现有的过程挖掘算法依赖于“*aba*”模式来挖掘2度循环,而满足局部完备性的日志文件中不一定出现该模式.为此,扩展了经典 Alpha 算法,提出了α^{L+}算法,用于从没有“*aba*”模式的日志文件中挖掘出 2 度循环.首先建立任务间的次序向量矩阵,用于抽象 2 度循环结构的变体结构;然后从全局视角,根据事件的出现次数及位置来区分 2 度循环和并发关系;最后提出紧邻度和回路抽象,以排除并发分支上同类型循环带来的干扰.实验结果表明,与现有的挖掘算法相比,α^{L+}算法能够从具有“*aba*”模式或不具有“*aba*”模式的日志文件中挖掘出 2 度循环.此外,该算法实现且集成在开源框架 ProM 中.

关键词: 日志抽象;向量矩阵;过程挖掘;局部完备性;Petri 网

中图法分类号: TP311

中文引用格式: 林雷蕾,周华,代飞,朱锐,李彤.一种从无“*aba*”模式的日志中挖掘 2 度循环的方法.软件学报,2018,29(11):3278−3294. http://www.jos.org.cn/1000-9825/5478.htm

英文引用格式: Lin LL, Zhou H, Dai F, Zhu R, Li T. Approach to mining length-two loops from the log without “*aba*” pattern. Ruan Jian Xue Bao/Journal of Software, 2018,29(11):3278−3294 (in Chinese). http://www.jos.org.cn/1000-9825/5478.htm

Approach to Mining Length-Two Loops From the Log Without “*aba*” Pattern

LIN Lei-Lei[1,3], ZHOU Hua[2], DAI Fei[2,3], ZHU Rui[1,3], LI Tong[1,3]

[1](School of Software, Yunnan University, Kunming 650091, China)

[2](School of Big Data and Intelligence Engineering, Southwest Forestry University, Kunming 650224, China)

[3](Key Laboratory for Software Engineering of Yunnan Province, Yunnan University, Kunming 650091, China)

Abstract: The current research in mining length-two loops depends on “*aba*” pattern. However, the pattern does not necessarily appear in the logs that satisfies local completeness. This research aims at finding ways to mine length-two loops without the pattern. It results in a new algorithm (α^{L+}-algorithm) that is based on the α-algorithm. First, an order vector matrix is established by tasks in logs to abstract variant structures of length-two loops. Then, distinction between loops and concurrency structure is obtained by event’s frequency and location in traces. Finally, proximity and circuit abstraction are used to eliminate the interference caused by the concurrent branches. The

* 基金项目: 国家自然科学基金(61462095, 61702442, 61662085); 云南省自然科学基金(2016FB102); 云南省中青年学术和技术带头人后备人才培养项目(C6143002); 云南省软件工程重点实验室开放基金(2017SE201, 2016SE202); 云南省教育厅科学研究基金(2017YJS107, 2017ZZX227); 云南大学研究生创新项目(YDY17095)

Foundation item: National Natural Science Foundation of China (61462095, 61702442, 61662085); Yunnan Province Natural Science Foundation (2016FB102); Talent Project of Yunnan Province (C6143002); Open Fund Project of Key Laboratory of Software Engineering of Yunnan Province (2017SE201, 2016SE202); Yunnan Provincial Department of Education Science Research Fund (2017YJS107, 2017ZZX227); Graduate Innovation Project of Yunnan University (YDY17095)

本文由面向智能制造的业务过程管理与服务技术专题特约编辑王建民教授、刘建勋教授推荐.

收稿时间: 2017-07-20; 修改时间: 2017-09-16; 采用时间: 2017-11-14; jos 在线出版时间: 2017-12-06

CNKI 网络优先出版: 2017-12-06 15:37:29, http://kns.cnki.net/kcms/detail/11.2560.TP.20171206.1536.025.html

experimental results show that the α^{L+}-algorithm can handle length-two loops with or without “*aba*” pattern. In addition, the α^{L+}-algorithm is implemented in the ProM tool.

Key words: log abstraction; vector matrix; process mining; local completeness; Petri net

社会经济的发展,物联网、云计算等新兴技术革命的出现,使得信息系统不仅仅是围绕处理业务数据为中心,更多时候与它们所支持的运作流程越来越紧密[1].同时,业务流程的操作使得信息系统记录了数量众多的事件,如何有效地从这些日志事件中提取有价值的信息,是企业实现新型商务智能的一个重要基础.过程挖掘是业务过程自动化建模及分析的一个重要技术手段,过程挖掘的应用场景主要有过程发现(process discovery)、符合性检查(process conformance)和模型增强(process enhancement)这3个方面[1],其中,过程发现是过程挖掘最为重要的应用.目前,已经有很多过程挖掘算法被提出:Alpha 算法(α算法)[2]、启发式挖掘算法[3]、基于遗传算法挖掘[4]、基于Markov链的挖掘算法[5]、整数线性规划挖掘[6]和基于区域的挖掘[7]等.以上算法在挖掘方面各有特点,但在挖掘并发模型的效率上,van der Aalst提出的Alpha算法还是具有很大优势的,其中一个主要原因是该算法是基于局部完备性(local complete)日志,而其他算法需要日志需满足全局完备性(global complete)才能得到较好的并发模型[8,9].但对于存在大量并发及循环任务的模型,要产生满足全局完备性的日志,基本上是不可能的.

从满足局部完备性的日志文件中挖掘过程模型,现有影响最大、应用最广的过程挖掘算法是经典的Alpha算法(α算法).经典的Alpha算法能够从日志中很好地挖掘出事件间的顺序关系、选择关系、长循环(长度大于2的循环)及并发关系,但无法挖掘出短循环(长度为1的循环和长度2的循环).为此,文献[10]将经典的α算法扩展为α^{+}算法,用于挖掘出1度循环和挖掘2度循环.但是文献[10]在挖掘2度循环时,规定日志文件必须满足循环完备性(loop-complete),即日志文件中必须出现“*aba*”和“*bab*”行为特征模式,其中,*a* 和 *b* 分别代表两个不同的任务,*aba* 与 *bab* 都是任务在日志中的行为记录形式.对于不出现“*aba*”模式、但满足局部完备性的日志文件,现有算法均不能很好地挖掘出2度循环.

为解决上述问题,本文将经典α算法进行扩展,提出了α^{L+}算法,用于从满足局部完备性的日志中挖掘出2度循环.与文献[10]相比,α^{L+}算法并不规定日志文件中必须满足循环完备性.该算法的基本思想是:首先,采用次序向量对原始日志进行抽象化简,排除 2 度循环的变体结构(即循环里面包含子流程等),然后从全局视角,结合两种最简2度循环的结构特征——三角形2度循环和棱形2度循环,用以区分三角形2度循环、棱形2度循环和并发结构;其次,针对并发分支上的两种干扰结构,分别提出紧邻度和基于次序的回路抽象方法进行有效地解决;最后,对经典的Alpha算法进行扩展,以挖掘出2度循环.

本文的主要贡献如下:

(1) 通过构造任务间的基本次序向量矩阵,对原始日志进行抽象,得到不包含 2 度循环变体结构的新日志文件;

(2) 从全局视角,通过事件执行的次数以及任务间的位置,识别出三角形2度循环、棱形2度循环和并发结构;

(3) 提出紧邻度和回路抽象的方法,分别针对并发分支上两种结构存在识别混乱及识别错误的问题进行有效的解决,降低干扰因素;

(4) 扩展经典的Alpha算法,并将算法实现为ProM的插件,通过实验说明了算法的有效性.

本文第1节是例子与问题分析,通过实际例子来提出论文研究的主要内容.第2节是预备知识,主要介绍本文涉及的相关基本概念和定义.第3节具体介绍整个2度循环挖掘的过程.第4节是对案例验证及结果分析.第5节是相关工作介绍.第6节是工作总结及下一步研究展望.

1 实例分析

图1展示了网上购物过程中一部分简单的过程模型,其中,任务 *s* 表示“在线支付”,任务 *a* 表示“输入验证码”,任务 *b* 表示“重新发送”,任务 *c* 表示“输入账号密码”,任务 *d* 表示“错误提示”,任务 *e* 表示“付款”.选择在线支付后,

用户执行的任务可以是先输入绑定的银行卡密码,也可以先输入验证码,所以二者没有先后顺序,可以认为是并发执行的.在输入验证码的同时,由于随机生成的验证码存在难以辨认的问题,用户可以重复多次要求系统重新发送验证码.在输入账号密码时,系统会随时提示账号不存在或密码不合理等.如图 1 所示的过程模型中,会产生两种不同的日志 W_1=[⟨*s*,*c*,*d*,*c*,*a*,*e*⟩,⟨*s*,*a*,*b*,*a*,*c*,*e*⟩,⟨*s*,*a*,*b*,*c*,*a*,*d*,*c*,*e*⟩,⟨*s*,*c*,*d*,*a*,*c*,*b*,*a*,*e*⟩,⟨*s*,*a*,*c*,*d*,*b*,*a*,*c*,*b*,*d*,*c*,*a*,*e*⟩]和 W_2=[⟨*s*,*c*,*a*,*e*⟩,⟨*s*,*a*,*c*,*b*,*d*,*c*,*a*,*d*,*c*,*e*⟩,⟨*s*,*a*,*b*,*c*,*d*,*a*,*c*,*d*,*b*,*a*,*c*,*e*⟩].这两个日志都满足了任务间的局部完备性要求,唯一的不同是在日志 W_1 的前两条轨迹中分别存在"*cdc*"和"*aba*"模式,而日志 W_2 中所有轨迹都没有出现该类型模式.针对日志 W_1,现有的α^+算法是可以挖掘出正确的过程模型;相反,把日志 W_2 作为输入,却得不到原模型.

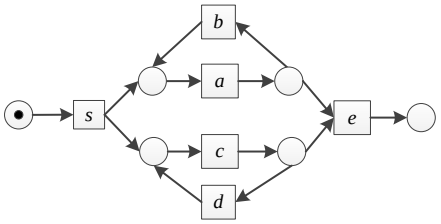


Fig.1 A shopping business process model represented as a Petri net

图 1 一个 Petri 网描述的购物过程模型

针对 W_1 和 W_2,使用经典α算法挖掘出来的模型如图 2(a)所示,与图 1 所示的过程模型不同.为此,Aalst 等人对原有算法进行扩展,提出了α^+算法:要求日志必须满足循环完备性.即在满足局部完备性的前提下,日志中必须存在模式"*aba*"和"*cdc*",以便区分 2 度循环和并发关系.针对 W_1,使用α^+挖掘出的过程模型如图 2(b)所示,与图 1 所示过程模型相同.但对于 W_2,使用α^+挖掘出的过程模型依旧不符合原模型(如图 2(a)所示).究其原因在于,W_2 虽满足局部完备性,但不存在"*aba*"和"*cdc*".因此,α^+无法从满足局部完备性、但不出现上述紧邻模式的日志文件中挖出 2 度循环.

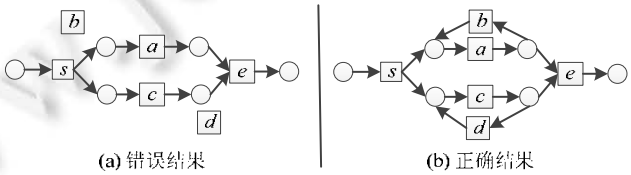


Fig.2 ProM tool mining results

图 2 采用 ProM 工具挖掘的结果

对于上述问题,本文认为日志轨迹中不存在紧邻模式"*aba*"的本质原因是:由于其他并发分支上的活动执行,打断了原有的紧邻模式.然而通过对大量实验的观察发现,虽然局部不能形成"*aba*"模式,但是全局日志行为与原有的结构是密切相关的.展开讲,对于任意两个任务 *a* 和 *b*,如果是顺序关系,则所有日志轨迹中的行为一定是任务 *a* 执行后,任务 *b* 才能执行,表现在轨迹中为紧邻执行(反之亦然);如果是选择关系,则所有日志中二者的前集任务执行完后,只能执行其中一个任务(*a* 或 *b*);如日志轨迹中存在 *a* 紧邻 *b* 执行,*b* 也紧邻 *a* 执行,则表明二者没有任何因果关系,即为并发关系;对于 2 度循环结构,局部虽然满足并发关系的行为记录,但从全局视角观察发现,其还有独特的行为记录.比如,图 1 中的模型产生每条轨迹中,*a* 记录的次数一定比任务 *b* 的多(前提是排除了 1 度循环),并且 *b* 要发生必须得 *a* 先发生.这就是 2 度循环所体现的其中一个事件特征.基于此观察,本文将对α算法进行扩展,提出一种α^{L+}算法,从满足局部完备性的日志,挖掘出 2 度循环.

需要说明的是,无论是"*aba*"还是"*cdc*",都是指两个不同任务在日志文件中紧邻出现的模式,同日志 W_1 中以⟨*a*,*b*,*a*⟩和⟨*c*,*d*,*c*⟩表示是一样的.为了表示方便及简单,后面统称"*aba*"模式.

2 预备知识

2.1 Petri网

Petri 网凭借其形式化基础、直观的图形化表示及分析技术,在业务过程管理领域常常被用于建模和分析业

务过程.因此,给出文中涉及的相关概念定义,如无特殊说明均直接引用,更多内容参见文献[11].

定义 2.1(Petri 网)[11]. Petri 网是一个四元组 $N=(P,T;F,M)$,其中,

1) $P\cup T\neq\varnothing$,习惯称 P 为库所集,T 为变迁集;
2) $P\cap T=\varnothing$;
3) $F\subseteq(P\times T)\cup(T\times P)$,称 F 为流关系;
4) 映射 $M:P\to\{0,1,2,3,\ldots\}$称为 Petri 网的一个标识.通常用 M_0 表示 Petri 网的初始标识.

通常,在 Petri 网的图形化表示中,库所使用圆圈表示,变迁使用方框表示,流关系使用有向线段表示,托肯使用实心小黑点表示.设 $N=(P,T;F,M)$是一个 Petri 网,若 $x\in P\cup T$,${}^\bullet x=\{y|(y,x)\in F\wedge y\in P\cup T\}$,则称${}^\bullet x$ 为 x 的前集;若 $x\in P\cup T$,$x^\bullet=\{y|(x,y)\in F\wedge y\in P\cup T\}$,则称 $x^\bullet$为 x 的后集.

定义 2.2(可达标识集)[11]. 设 $N=(P,T;F,M_0)$是一个 Petri 网,N 的可达标识集 $R(M_0)$满足下面两个条件的最小集合.

1) $M_0\in R(M_0)$;
2) 若 $M\in R(M_0)$,且存在 $t\in T$ 使得 $M[t\rangle M'$,则 $M'\in R(M_0)$.

定义 2.3(死锁)[11]. 设 $N=(P,T;F,M_0)$是一个 Petri 网,存在标识 M,$\forall t\in T$,在 M 下均没有发生权,则称标识 M 为死锁.

定义 2.4(安全性)[11]. 设 $N=(P,T;F,M_0)$是一个 Petri 网,$\forall p\in P$,若存在正整数 $B=1$,使得$\forall M\in R(M_0):M(p)\leqslant B$,则称 N 是安全的.

本文讨论的内容都是在满足合理性的结构化 Petri 网的基础上,即满足:(1) Petri 网模型只有一个入口库所 i,一个出口库所 o;(2) Petri 网模型是安全的,从入口标识到出口标识是可达的,且中间不残留托肯;(3) 不存在死变迁.此外,α^+算法已经很好地解决了 1 度循环(自循环).所以,本文讨论的过程模型假设不存在自循环.

2.2 事件日志

过程发现的本质就是在分析信息系统记录的日志文件基础上构造过程模型.事件日志记录了事件发生的多方面信息,但在过程挖掘的理论方面,最关心的问题是能否从事件执行的先后顺序中发掘整个系统事件的控制流模型.下面给出本文涉及相关概念的定义,更多内容参见文献[8,9].

定义 2.5(事件轨迹和事件日志)[9]. 假设 T 为所有任务集合,$\sigma\in T^*$是一条事件轨迹,$W\subseteq\mathcal{P}(T^*)$是一个事件日志.

即日志是轨迹的集合,每条轨迹由有限个任务按照执行次序排列而成(也可以说由多个事件组成).如图 1 中,$\sigma=\langle s,c,a,e\rangle$为一条轨迹,$W_2=[\langle s,c,a,e\rangle,\langle s,a,c,b,d,c,a,d,c,e\rangle,\langle s,a,b,c,d,a,c,d,b,a,c,e\rangle]$为其满足局部完备性的一个事件日志.为了方便起见,文中皆省略了案例属性及轨迹的发生次数.

定义 2.6(基本次序关系)[10]. 令 T 为任务集合,$W\subseteq\mathcal{P}(T^*)$是一个事件日志.对于$\forall a,b\in T$:

- $a>_W b$ 当且仅当存在一条轨迹$\sigma=$“$t_1t_2t_3\ldots t_{n-1}t_n$” and $i\in\{1,\ldots,n-1\}$,同时满足$\sigma\in W$ and $t_i=a\wedge t_{i+1}=b$;
- $a\to_W b$ 当且仅当 $a>_W b$ and $b\not>_W a$;
- $a\#_W b$ 当且仅当 $a\not>_W b$ and $b\not>_W a$;
- $a\|_W b$ 当且仅当 $a>_W b$ and $b>_W a$.

定义 2.7(全局完备性的事件日志)[9]. 令 T 为有限的任务集合,N 是基于 T 的过程模型,W 是基于 T 的事件日志.W 是全局完备的当且仅当所有出现在日志中的轨迹与过程模型 N 执行产生的所有可能关于 T 的序列都是等价的.

定义 2.8(局部完备性的事件日志)[8]. T 代表有限的任务集合,令 $W\subseteq\mathcal{P}(T^*)$代表事件日志,则 W 满足局部完备性的条件是:(1) 假设σ是模型 N 产生的任意一条执行轨迹,则满足$>_\sigma\subset>_W$;(2) 对于$\forall t\in T$,必存在一条轨迹$\sigma\in W$且 $t\in\sigma$.

根据定义 2.7 和定义 2.8 可知,全局完备性的日志要求是过程模型中所有可能执行的序列轨迹都要存在.这对于并发任务就是一个排列组合问题.

3 挖掘 2 度循环过程

本节将讨论从不含有“*aba*”模式的日志中,如何挖掘出 2 度循环.该挖掘过程包括 4 步,如图 3 所示.

- 第 1 步,日志抽象.对原始日志进行抽象,排除 2 度循环变体结构带来的结构干扰.
- 第 2 步,挖掘两种 2 度循环.根据不同结构体现出来的事件记录形式不同,挖掘出 2 度循环.
- 第 3 步,排除并发分支上的同类型循环引起的干扰.采用紧邻度排除并发分支上同类型 2 度循环的干扰,采用回路抽象排除并发分支上同类型的长循环结构引起的干扰.
- 第 4 步,扩展 Alpha 算法.在经典 Alpha 算法中事件间的次序关系中扩展 2 度循环关系,同时扩展原有算法步骤,挖掘出最终过程模型.

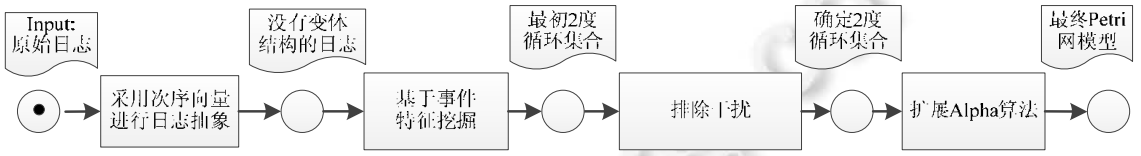


Fig.3 A process of mining length-2-loop

图 3 2 度循环挖掘的过程

3.1 日志抽象

日志循环是为了排除 2 度循环的变体.如图 4 所示,图 4(a)和图 4(b)这两种类型 Petri 网结构都是本文认为最简 2 度循环.然而,Petri 网结构在满足合理性的前提还是具备很多种变型,如图 4(c)和图 4(d)所示的两种 2 度循环的变体结构与图 4(a)和图 4(b)分别对应.这些变体结构种类繁多,比如将任务 n 细化为一个子网.限于篇幅,本文只考虑图 4(c)和图 4(d)中这两种结构,其他结构解决方式都是类似的.

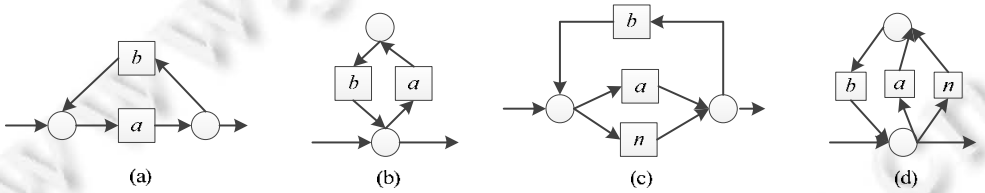


Fig.4 Length-2-Loops and its variant structure

图 4 2 度循环及其变体结构

定义 3.1(三角形 2 度循环,用△表示). 令 $N=(P,T;F,M_0)$为一个基本 Petri 网模型,任务 a,b 是 N 中的两个变迁,$a,b\in T$.a,b 构成三角形 2 度循环 $a\triangle b$ 当且仅当:(1) ${}^\bullet a={b}^\bullet,{a}^\bullet={}^\bullet b$ 且 $a\neq b$ and ${}^\bullet a\cap{a}^\bullet=\varnothing$;(2) 假设 $M_1\in R(M_0)$,使得 $M_1[a>M_2$,且不存在 $M_1[\sigma>M_2$,其中,σ有穷点火序列,则仅存在 $M_2[b>M_1$,若 M_2 非最终标识且存在 $M_2[x>M_3$,其中,$x\in T,a\neq x\neq b$(如图 4(a)所示).

定义 3.2(棱形 2 度循环,用◇表示). 令 $N=(P,T;F,M_0)$为一个基本 Petri 网模型,任务 a,b 是 N 中的两个变迁,$a,b\in T$.a,b 构成棱形 2 度循环 $a\diamond b$ 当且仅当:(1) ${}^\bullet a={b}^\bullet,{a}^\bullet={}^\bullet b$ 且 $a\neq b$ and ${}^\bullet a\cap{a}^\bullet=\varnothing$;(2) 假设 $M_1\in R(M_0)$,使得 $M_1[a>M_2$,且不存在 $M_1[\sigma>M_2$,其中,σ有穷点火序列,则仅存在 $M_2[b>M_1$,若 M_1 非最终标识且存在 $M_1[x>M_3$,其中,$x\in T,a\neq x\neq b$(如图 4(b)所示).

定义 3.1 和定义 3.2 分别从静态结构和动态执行标识来描述了两种最简 2 度循环.从静态结构来看,任务 a 和任务 b 都形成了简单的回路关系,因此采用 Petri 网的前后集来定义.从动态执行的角度,运用 Petri 网的点火标识来体现两者的执行差异.根据 2 度循环的定义,可以判定能够组成其变体结构的任务只有选择关系.

定理 1. 对于任意一个任务 n,且存在 2 度循环 $a\triangle b$,若要构成 $n\triangle b$,则 a 与 n 一定是选择关系 $a\#n$.

证明:采用反证法.假设现有两个任务 a,b 构成了最简的 2 度循环结构 $a\triangle b$,其中,$N=(P,T;F,M_0)$为一个基本 Petri 网模型,任务 a,b 是 N 中的两个变迁,$a,b\in T$.现存在任务 $n\in T,n$ 与 b 构成一个新的 2 度循环结构 $n\triangle b$:如图

4(a)所示,由于 $a\triangle b$,则满足${}^\bullet a={}^\bullet b, a^\bullet={}^\bullet b$.现如果 $n\triangle b$ 且 a 与 n 不是选择关系,则 a 与 n 则为顺序关系、并发关系中的一种.假设 a 与 n 为顺序关系,则很容易推出${}^\bullet a\neq{}^\bullet n$或者 $a^\bullet\neq{}^\bullet n$.又因为${}^\bullet a={}^\bullet b, a^\bullet={}^\bullet b$,所以${}^\bullet n\neq{}^\bullet b$或者 $n^\bullet\neq{}^\bullet b$.这与 $n\triangle b$ 的定义是互相矛盾的.因此,假设不成立.同理可证明,并发关系也是不成立的.证毕. □

以上定理 1 的证明对 2 度循环结构◇的变体也是适用的,并且同理得:若 $a\triangle n$,则 $n\#b$.因此,综上所得,2 度循环结构△和◇的变体结构中,只能出现选择关系.根据定义 2.6 可以发现,选择关系的任务与其他任务的基本次序是相等的.为此,算法 1 给出了日志抽象的过程,其主要思路是:根据最初日志记录,建立每个任务之间的相互次序向量,然后将次序向量相等的任务抽象为一个新的任务.

算法 1. *VariantAbstracting*(W) return W^+.

输入:满足局部完备性的日志 W.

输出:不存在 2 度循环变体结构的日志 W^+.

BEGIN

1. 定义二维矩阵 *order*[*n*][*n*]={0,0,…,0},定义动态链表 *list*; //其中,*n* 代表 *w* 中任务集合个数
2. 根据定义 2.6 计算出 W 中基本次序关系 *order*; //0 代表选择#,1 代表顺序→,2 代表并发||
3. 将 *order* 中每一行值转为一个一维向量 v_i,整个二维矩阵形成 $V=[v_1,v_2,\ldots,v_n]$;
4. 循环对比 V 中任意两个向量:
   4.1. IF (v_i.*equals*(v_i+1) && *order*[*i*][*i*+1]==0 && v_i.*contians*(2))
      4.1.1. *variantList.add*(t_i,t_i+1); //v_i 即任务 t_i 的次序向量
5. WHILE (*j* 小于 *variantList* 的长度) //将多个相等向量的任务放在一个集合
   5.1. 若 *variantList.get*(*j*)与 *variantList.get*(*j*+1)存在交集,合并两个集合;
6. 将 *list* 中每个集合中的任务抽象为新的任务,修改日志 W 为 W^+;

END

任务的次序向量代表当前任务与其他任务之间的基本次序关系.同时,两个任务若为选择关系(如图 4(c)中的任务 a 与 n),则体现在日志轨迹中是可以相互替换,所以构造出二者的次序向量是相等的.算法 1 中的第 4 步就是基于此思想,将选择关系找出来;然后,第 5 步将多个互为选择关系的任务放在一起;第 6 步是将这些任务抽象为一个新的任务,同时修改原有日志文件.

3.2 挖掘2度循环结构

挖掘 2 度循环结构是指从没有“*aba*”模式的的日志文件中发现三角形 2 度循环和棱形 2 度循环,其基本思想是:对于结构化的 Petri 网而言,其结构特征可以准确地反映行为,例如,任务的串行结构可以反映任务间的顺序行为等.因此,Petri 网中任务间具有串行结构、选择结构和并发结构在日志轨迹中表现出的事件间的前后位置和事件出现次数也不同.本文把事件在日志轨迹中所体现出来的前后位置和出现次数视为事件特征.首先,给出事件在轨迹中位置和次数的定义.

定义 3.3(事件位置). 令$\sigma\in T^*$为日志 W 中一条轨迹,T 为任务集合.对于$\forall a\in T, L(a)=\{k_i|k$ 为正整数且 $0\leqslant k_i<\sigma.sizeand, 1\leqslant i<\sigma.size\}$表示为事件 a 在当前轨迹σ中出现的位置的集合,其中,$\forall k_i\in L(a), k_i$ 代表 a 在轨迹中第 i 次出现的位置.如$\sigma=\langle s,a,b,a,c,e\rangle$,则 $L(a)=\{1,3\}$.

定义 3.4(事件次数). 令$\sigma\in T^*$为日志 W 中一条轨迹,T 为任务集合.对于$\forall a\in T, Sum(a)$表示为事件 a 在当前轨迹σ中的出现次数,其中,$Sum(a)=N, N$ 为自然数.

轨迹中的事件位置和事件次数是任务行为的记录,反映了 Petri 网中任务之间的结构特征.这种特征本文称之为事件特征,并给出 Petri 网基本结构体现出来的不同事件特征对比,见表 1.

Table 1 Different event characteristics of different structures in Petri net

表 1 Petri 网中不同结构的事件特征

序号	Petri 网结构	基本次序	事件特征	类似轨迹
1	顺序结构	$a\rightarrow b$	$\forall\sigma\in W$,若 $a,b\in\sigma$,则必存在 $k_i\in L(a),k_j\in L(b)$,且 $k_i-k_j=1$ and 不存在 $k_j-k_i=1$	$\sigma=\langle\ldots,a,b,\ldots\rangle$
2	选择结构	$a\#b$	$\forall\sigma\in W$,若 $a,b\in\sigma$,则不存在 $k_i\in L(a),k_j\in L(b)$,且$\lvert k_i-k_j\rvert=1$	$\sigma=\langle\ldots,a,\ldots\rangle$或 $\sigma=\langle\ldots,b,\ldots\rangle$
3	并发结构	$a\lVert b$	$\exists\sigma\in W$,令 $a,b\in\sigma$,其中,1) $k_i\in L(a),k_j\in L(b)$,则有$\lvert k_i-k_j\rvert=1$; 2) $Sum(a)=Sum(b)$,且$\exists k_j\in L(b)$,找不到 $k_i\in L(a)$,使得 $k_i<k_j$	$\sigma=\langle\ldots,ab,\ldots\rangle$或 $\sigma=\langle\ldots,ba,\ldots\rangle$
4	三角形 2 度循环结构	$a\triangle b$	其中,1) $\exists\sigma\in W$,令 $a,b\in\sigma,k_i\in L(a),k_j\in L(b)$,则有$\lvert k_i-k_j\rvert=1$; 2) $\forall\sigma\in W$,令 $a,b\in\sigma,Sum(a)-Sum(b)=1$;3) 若存在 $k_{j-1},k_j\in L(b)$,则必$\exists k_i\in L(a)$,使得 $k_{j-1}<k_i<k_j$;反之,k_{i-1},k_i 亦然	$\sigma=\langle\ldots,ab,\ldots,a,\ldots\rangle$或 $\sigma=\langle\ldots,a,\ldots,ba,\ldots\rangle$或 $\sigma=\langle\ldots,a,\ldots\rangle$
5	棱形 2 度循环结构	$a\diamond b$	其中,1) $\exists\sigma\in W$,令 $a,b\in\sigma,k_i\in L(a),k_j\in L(b)$,则有$\lvert k_i-k_j\rvert=1$; 2) $\forall\sigma\in W$,令 $a,b\in\sigma,Sum(a)=Sum(b)$;3) 若存在 $k_{j-1},k_j\in L(b)$,则必$\exists k_i\in L(a)$使得 $k_{j-1}<k_i<k_j$;反之,k_{i-1},k_i 亦然	$\sigma=\langle\ldots,ab,\ldots,a,\ldots,b,\ldots\rangle$或 $\sigma=\langle\ldots,a,\ldots,ba,\ldots,b,\ldots\rangle$或 $\sigma=\langle\ldots,a,b,\ldots\rangle$

由表 1 可知,不同结构反应在日志轨迹中的事件特征也是不同的.顺序结构、选择结构的事件特征实质上就是定义 2.6 所表示的紧邻关系,因此可以用经典 Alpha 算法找到.但是并发结构、三角形 2 度循环和棱形 2 度循环这 3 种结构仅依靠事件特征中第 1 个条件(实质是紧邻关系)是无法区分,必须结合后两个条件.因此,本文给出了算法 2 用于挖掘三角形 2 度循环.

算法 2. *MiningLoop*(W^+) return *triangleList*.

输入:满足局部完备性的日志 W^+.

输出:挖掘三角形 2 度循环.

BEGIN

1. 令 *iterator* 为迭代变量,Q 为日志 W^+的轨迹集合;
2. WHILE (*iterator* 小于 *Q.size*())
   2.1. 计算每条轨迹中出现的任务 t_i 的位置 $L(t_i)$和次数 $Sum(t_i)$;
3. 若 *order*[*i*][*j*]为 2,则 *mayloop.add*(t_i,t_j); //满足事件特征第 1 个条件
4. WHILE (*iterator* 小于 *mayloop.size*())
   4.1. WHILE (*iterator* 小于 *Q.size)*
      4.1.1. 判定$\langle t_i,t_j\rangle$是否满足的事件特征 2、特征 3 的条件;
   4.2. 若所有轨迹 Q 都满足,则动态链表添加三角形 2 度循环任务对 *triangleList.add*(t_i,t_j);

END

挖掘棱形 2 度循环的算法基本和算法 2 步骤一样,仅是将步骤 4.1.1 中的判定条件改为棱形 2 度循环的事件特征第 2 个条件即可,其挖掘结果放入动态链表 *prismList* 中,所以这里不给出其挖掘算法内容.

3.3 排除干扰

本节主要讨论算法 2 在挖掘 2 度循环时存在的两个干扰问题:识别混乱和识别错误,见表 2.

Table 2 Interference of loop structure in concurrent branch

表 2 并发分支上循环结构的干扰

序号	名称	Petri 网结构	产生的日志文件	算法 2 挖掘结果	结果分析
1	识别混乱	图 1 所示的结构	$W_3=[\sigma_1,\sigma_2]$,其中, $\sigma1=\langle s,a,c,b,d,c,a,d,b,c,d,a,b,c,a,e\rangle$, $\sigma2=\langle s,c,a,d,b,a,c,e\rangle$	$a\triangle b$, $c\triangle b$, $c\triangle d$	无法识别任务 c 到底是与任务 b 构成并发,还是与任务 d 构成三角形 2 度循环
2	识别错误	图 5(a) 所示的结构	$W_4=[\sigma_1,\sigma_2,\sigma_3,\sigma_4]$,其中, $\sigma_1=\langle t_0,t_4,t_1,t_2,t_5,t_6,t_3,t_4,t_1,t_2,t_3,t_1,t_7\rangle$, $\sigma_2=\langle t_0,t_1,t_2,t_4,t_3,t_1,t_5,t_2,t_6,t_4,t_3,t_1,t_7\rangle$, $\sigma_3=\langle t_0,t_1,t_4,t_2,t_3,t_5,t_1,t_2,t_3,t_6,t_1,t_4,t_7\rangle$, $\sigma_4=\langle t_0,t_1,t_2,t_4,t_5,t_3,t_1,t_6,t_2,t_4,t_3,t_1,t_7\rangle$	$t_2\triangle t_5$	任务 t_2 和 t_5, 本应该是并发关系,被错误地识别为循环关系

- 问题1:识别混乱.

本质在于并发分支上存在同类型2度循环结构,导致两个2度循环的事件特征在日志中交织在一起,形成识别混乱.针对识别混乱问题,启发于文献[12]采用了相关性计算公式 $ECC(a,b)$从全局的轨迹角度计算了两个事件类相关程度的思想,本文提出紧邻度的概念用于解决并发结构中多个同类型2度循环间的相互干扰问题.

定义 3.5(紧邻度). 令 W 是关于任务 T 的一个局部完备性日志,$\{\sigma_1,\sigma_2,\ldots,\sigma_n\}$是 W 中轨迹的集合,任务 $a,b\in T$,a,b 的紧邻度表示为 *Follow Degree*(a,b),简称 $FD(a,b)$.其中,$FD(a,b)$的计算公式如下:

$$FD(a,b)=\sum_{k=1}^{n}\sum_{r=1}^{m}\beta^{t-1},$$

其中,变量 m 是在每条轨迹中 b 出现的次数;n 是轨迹的数量;t 是任务 b 与任务 a 的距离;β是稀释因子,默认β=0.5.

紧邻度的核心思想是,正确的2度循环在全局日志中其紧邻程度比假的2度循环强(假的2度循环指二者应该为并发关系).算法3给出了具体的计算过程:采用算法3中步骤1.1的紧邻度计算 W_3 的结果是$\{FD(a,b)=2.5,FD(c,b)=1.75,FD(c,d)=2.5\}$,再结合步骤2和步骤3,最终确定2度循环的正确结果是 *TloopList*$=\{\langle a,b\rangle,\langle c,d\rangle\}$.注意,算法3换做棱形循环结构也可以解决,输出集合为 *PloopList*.

算法 3. *DivideLoop*(*triangleList*) return *TloopList*.

输入:满足结构特征的任务对.

输出:最后确定的循环对.

BEGIN

1. WHILE (*iterator* 小于 *triangleList* 的长度)

   1.1. $FD(a,b)$; //定义 3.5

2. 循环比较 *triangleList* 每个集合,若$\{a,b\}\in$*triangleList*,且所有跟 b 构成循环结构中 $FD(a,b)$最大,则 *TloopList*.*add*(a,b);

3. 确认 *TloopList* 中集合间不存在交集,返回最终结果;

END

- 问题2:识别错误.

本质是并发分支上存在长循环,导致了不同长循环之间的两个任务可以产生任意的事件特征(除了入口任务外,即 t_1 和 t_4).针对识别错误问题,本文提出在日志层面,通过定义2.6建立基本次序,然后通过次序向量来找到任务的回路(特指长循环结构),将整个回路抽象为一个新任务,再基于事件特征挖掘及紧邻度判定.需要说明的是,之所以没有放在第3.1节是为了避免内容混乱,保证章节内容的紧凑性.

定理 2. 构成并发的长循环结构,一定能抽象为不存在长循环的简单并发结构.

如图5所示,首先说明的是,图5(b)中的4种情况都会导致Petri网模型不合理,证明可参见文献[2].所以,证明过程是为了阐述图5(a)在日志层面可以化简为图5(c)的模型.

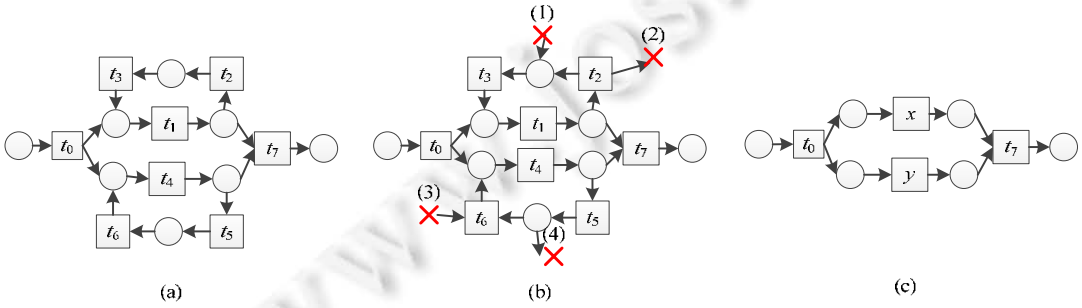


Fig.5 Illustrations of Theorem 2

图5 定理2证明图示

证明:为了证明定理 2,需要考虑两方面.

1) 长循环并发结构中,一定存在两个任务是真并发关系,不是 2 度循环;

2) 从该任务出发,基于次序向量,有限次数寻找后继,一定能找到长循环结构.

- 首先证明 1).

由于模型对应的日志满足局部完备性,所以图 5(a)中一定存在 $t_0>t_1,t_0>t_4,t_1>t_4,t_4>t_1$.又由于 Petri 网的点火规则,日志中一定存在轨迹为 $\sigma_1=\langle t_0,t_1,t_4,\ldots,t_7\rangle,\sigma_2=\langle t_0,t_4,t_1,\ldots,t_7\rangle$.任务 t_4 和 t_1 虽然可以任意次数出现,且满足并发关系,但是在轨迹中为了保证 t_0 的局部完备性,一定会出现 t_1 的前面没有 t_4,t_4 的前面也可以没有 t_1.这就破坏了二者形成 2 度循的事件特征.因此,两者为真并发,不会挖出二者满足 2 度循环.

- 接着证明 2).

令并发分支上存在长循环,分两种情况:长循环包含任务 t_1 和不包含 t_1 的.

- 对于第 1 种情况,现从任务 t_1 出发,运用广度优先搜索在每个任务的次序向量中找其后继任务,由于日志是满足局部完备性的,所以肯定能找到从 t_1 为出发点又回到 t_1 的循环路径.
- 第 3 种情况是长循环不包含 t_1,但是证明过程类似,依然从 t_1 出发,总是能找到某个任务通过自身的后继完后找到,又回到本身.所以,基于次序向量通过有限次寻找后继,一定能将长循环任务抽象.

综上可得,并发分支上的长循环结构,一定能够通过满足局部完备性的日志找到,并抽象.证毕. □

算法 4. *CircuitAbstracting*(W) return W^+.

输入:满足局部完备性的日志 W.

输出:不存在并发分支上的长循环结构的日志 W^+.

BEGIN

1. 定义二维矩阵 *order*[*n*][*n*]={0,0,…,0},定义动态链表 *list*;
2. 根据定义 2.6 计算出 W 中基本次序关系 *order*;
3. 将 *order* 中每一行值转为一个一维向量 v_i,整个二维矩阵形成 $V=[v_1,v_2,\ldots,v_n]$;
4. 循环对比 V 中任意两个向量 v_i 和 v_j
   4.1. IF(*order*[*i*][*j*]==2 && v_i 与 v_j 存在共同前继)
      4.1.1. 采用广度优先搜索分别从 v_i 与 v_j 出发,寻找当前任务的后继,建立动态链表;
      4.1.2. 如果动态链表形成回路,则回路中所有任务添加到 *circuitList* 中,并记下入口出口;
5. WHILE (*index* 小于 *circuitList* 的长度)
   5.1. 将 *circuitList.get*(*index*)中的任务全部替换为一个新的任务,修改日志;

END

算法 4 中,为了保持内容完整性,前 3 步与算法 1 是一致的.核心步骤为第 4 步,其中满足并发关系且存在共同前继的两个任务肯定是真并发.再采用广度循环搜索算法,寻找当前任务的后继(在次序向量中表现为含有 1 所对应的列),一旦形成回路的链表,就表明存在长循环结构,并抽象.

3.4 扩展Alpha算法

扩展 Alpha 算法是在前面分析的基础上重新定义基本次序,进而扩展经典的 Alpha 算法内容,使其能从不具备“*aba*”模式的日志中挖掘出含有 2 度循环的过程模型.

定义 3.6(挖掘循环的次序关系). 令 $N=(P,T,F;M_0)$为合理的 WF-net,W 为 N 的局部完备事件日志,对于 $\forall a,b\in T$,

- $a\triangle_W b$ 当且仅当$\langle a,b\rangle\in TloopList$,
- $a\diamond_W b$ 当且仅当$\langle a,b\rangle\in PloopList$,
- $a>_W b$ 当且仅当存在一条轨迹σ=“$t_1t_2\ldots t_{n-1}t_n$” and $i\in\{1,\ldots,n-1\}$,同时满足$\sigma\in W$ and $t_i=a$, $t_{i+1}=b$,
- $a\rightarrow_W b$ 当且仅当 $a>_W b$ and ($b\not>_W a$ or $a\triangle_W b$ or $a\diamond_W b$),
- $a\#_W b$ 当且仅当 $a\not>_W b$ and $b\not>_W a$,

- $a\|_W b$ 当且仅当 $a>_W b$ and $b>_W a$ and $\neg(a\triangle_W b)$ and $\neg(a\diamond_W b)$.

定义 3.6 主要增加了三角形 2 度循环和棱形 2 度循环两种次序,再结合紧邻关系$>_W$,用于定义顺序关系$\rightarrow_W$和并发关系$\|_W$.

定义 3.7(α^{L^+}算法). 令 W 表示基于任务 T 的一个局部完备日志,那么$\alpha^{L^+}(W)$的定义如下:

(1) $T_w=\{t\in T|\exists\sigma\in w:t\in\sigma\}$;

(2) $T_I=\{t\in T|\exists\sigma\in w:t=first(\sigma)\}$;

(3) $T_O=\{t\in T|\exists\sigma\in w:t=last(\sigma)\}$;

(4) $X_W=\{(A,B)|A\subseteq T_w\wedge A\neq\varnothing\wedge|B\subseteq T_w\wedge B\neq\varnothing\wedge\forall a\in A,\forall b\in B,a\rightarrow_W b\wedge\forall a_1,a_2\in A,a_1\#_W a_2\wedge\forall b_1,b_2\in B,b_1\#_W b_2\}$;

(5) $Y_W=\{(A,B)\in X_W|\forall(A',B')\in X_W A\subseteq A'\wedge B\subseteq B'\Rightarrow(A,B)=(A',B')\}$;

(6) For each $(A,B)\in Y_W$ do:

(a) For each $a\in A$ do:

If (a is avariant task) then restoring the original tasks;

(b) For each $b\in B$ do:

If (b is avariant task) then restoring the original tasks;

(7) $P_W=\{P_{(A,B)}|(A,B)\in Y_W\}\cup\{I_w,O_w\}$;

(8) $F_W=\{(a,P_{(A,B)})|(A,B)\in Y_W\wedge a\in A\}\cup\{(P_{(A,B)},b)|(A,B)\in Y_W\wedge b\in B\}\cup\{(I_w,t)|t\in T_I\}\cup\{(t,O_w)|t\in T_o\}$;

(9) $\alpha^{L^+}(W)=\{P_W,T_w,F_W\}$;

算法α^{L^+}的定义内容主要体现在扩展了第 6 步:由于针对原始日志中的 2 度循环变体结构进行了抽象,则在算法第 5 步消除冗余关系后,紧接着第 6 步是将抽象出来的新任务删除掉,还原原始的多个选择任务.然后再添加任务之间的库所和流关系,得到最终模型.注意,长循环的抽象在这里没有扩展进来.原因在执行α^{L^+}算法前,对输入的日志 W 就已经做了还原处理.这里为了保证与原有 Alpha 算法的对比性,所以没有体现出来.

3.5 更多讨论

本文提出在不具备“aba”模式的局部完备性日志中挖掘 2 度循环结构,是现有挖掘算法的一个扩展补充.但是,该算法目前还无法解决结构极其复杂、非结构化的 2 度循环问题.下面重点讨论 3 个问题.

(1) 过程模型限定问题:为什么要限定挖掘的过程模型是合理的结构化 Petri 网?

首先,合理的 Petri 网模型是现有过程管理的基础,一个不合理的过程模型一定存在错误,则其产生的日志轨迹也失去了实际意义;其次,结构化的 Petri 网其行为记录(日志文件)能充分反映原有模型的结构特征,而非结构化 Petri 网,其行为记录无法准确刻画任务间的结构.如图 6 所示,该模型产生的日志 $W=[\langle a,b\rangle,\langle a,c,d,b\rangle,\langle e,d,c,f\rangle,\langle e,f\rangle]$,日志中既不出现模式“aba”,又满足了局部完备性.当前,挖掘算法都无法准确挖掘到该模型中的 c 与 d 构成的 2 度循环,究其原因是,非结构化的模型产生的日志无法刻画任务间的真正关系.现有方法是在模型层面将非结构化的模型转化为结构化的模型[13,14].此外,文献[14]明确指出了何为结构化模型:一个过程模型是结构化的,则对于任意一个具有多条输出弧的节点(分叉点),都存在唯一一个具有多条输入弧的节点(汇聚点)与之对应,反之亦然;同时,由分叉和汇聚点之间的节点和边构成的模型片段称之为一个单入单出的过程组件;如果,整个过程模型能分解为一个个过程组件,则该过程模型是结构化的,否则,为非结构化的.

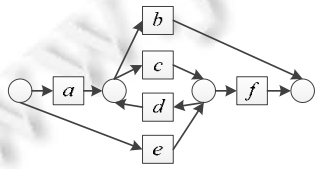


Fig.6 An unstructured process model[10]

图 6 非结构化的过程模型[10]

(2) 2 度循环结构复杂变体问题:本文仅讨论由单个任务组成的 2 度循环变体结构,对于子网组合的变体如何解决?

本文提供了基于次序向量的方法来解决 2 度循环结构的变体,如图 4(c)所示.如果将图 4(c)中的任务 a 和任务 n 分别替换为两个子网,按照本文的方式,也可以通过抽象来完成,只是需要更多的约束条件来保证抽象的选择关系的准确性.但是如果子网里面还有 2 度循环,相当于 2 度循环里面有子网,子网里面又有 2 循环,这就极其复杂,就当前的解决方式,还不能较好地解决,需要进一步进行讨论.实际当中,能够做到两层不同结构相互嵌套,模型已经很复杂了,极少出现出现 3 层及以上不同结构相互嵌套.以上嵌套结构,特指循环以及更加复杂的子网嵌套,对于简单的结构嵌套,基本不影响本文算法.

(3) 回路抽象问题:针对识别错误问题,本文采用了回路抽象方法,能够准确抽象吗?效果又如何呢?

定理 2 已证明,从日志中一定能够长循环找到,并抽象为一个新的任务.现有一些研究也充分体现了这一思想,如团队成员已在文献[15]中准确地找出了单触发序列中的循环结构,并成功地划分日志轨迹以达到更好的拟合度.文献[16]采用了增量式的挖掘方式来解决循环结构问题,本质上也是将长循环抽象为一个任务.而本文中所采用的抽象方式是基于次序向量来寻找,与已有的工作相比,该算法效率更优.唯一不足的是,如果处于并发分支上的两个长循环里面又嵌套 2 度循环,该抽象会导致挖掘内部 2 度循环无法被挖掘出来.当然,现有的抽象方法也都会碰到这个难题,而且这种嵌套已经形成了上述问题 2 中讨论的 3 层嵌套结构,存在的概率极小.针对该问题,本文暂时不作讨论,将在后续工作中展开研究.

4 案例测试及算法评估

本节主要从两个方面来验证上述理论内容:(1) 通过人工案例来测试α^{L+}算法的有效性;(2) 通过人工案例及实际案例相结合方式来对现有循环挖掘算法进行测试对比,体现本文算法的优势.

4.1 案例测试

算法α^{L+}内容已经成功地实现了插件,并集成到开源平台 ProM 中[17].文献[17]实现插件的步骤包括:

1) 配置需要运行的环境.如安装 eclipse 集成环境及配置 java 运行的环境(此部分内容读者可在网上轻松获取到相关资料).
2) 下载 ProM 的开源代码.首先要求读者安装一个版本控制器,如 TortoiseSVN;其次,创造一个新的文件夹“new file”,打开文件后,右键,选择“SVN Checkout”.在 URL of repository 中输入下载源码的地址:https://svn.win.tue.nl/repos/prom/Framework/trunk/.点击“OK”下载就行.
3) 将源码导入至 eclipse 环境中,所有的插件代码在项目中的“src-Plugins”文件下面添加即可.
4) 运行 ProM 工具.网址 https://svn.win.tue.nl/trac/prom/wiki/setup/RunningProM 很详细地说明了如何选择哪个类进行运行.需要说明的是,如果是第 1 次运行 ProM 工具,请确保计算机连着网络,耐心等待它自动安装完所需插件.

为了验证算法的有效性,首先通过一个较为复杂的人工案例来进行测试.图 7 为人工案例模型,表 3 为该模型产生的日志文件.

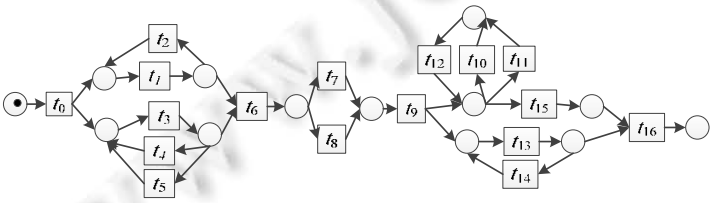


Fig.7 An artificial model

图 7 人工模型

Table 3 An event log from the artificial model

表 3 人工模型产生的事件日志

日志名称	日志轨迹内容	日志分析
W_5	$W_5=[\sigma_1,\sigma_2,\sigma_3,\sigma_4]$,其中, $\sigma_1=\langle t_0,t_1,t_3,t_6,t_7,t_9,t_{10},t_{13},t_{12},t_{11},t_{14},t_{12},t_{10},t_{12},t_{13},t_{15},t_{14},t1_3,t_{16}\rangle$, $\sigma_2=\langle t_0,t_1,t_2,t_3,t_4,t_1,t_3,t_5,t_2,t_1,t_3,t_2,t_5,t_3,t_1,t_6,t_8,t_9,t_{13},t_{11},t_{12},t_{14},t_{10},t_{13},t_{14},t_{12},t_{13},t_{10},t_{12},t_{15},t_{16}\rangle$, $\sigma_3=\langle t_0,t_3,t_5,t_1,t_3,t_2,t_4,t_3,t_1,t_4,t_2,t_3,t_1,t_5,t_3,t_6,t_8,t_9,t_{11},t_{13},t_{12},t_{14},t_{13},t_{10},t_{14},t_{12},t_{13},t_{14},t_{11},t_{12},t_{15},t_{13},t_{16}\rangle$, $\sigma_4=\langle t_0,t_3,t_1,t_6,t_7,t_9,t_{15},t_{13},t_{16}\rangle$; $\sigma_5=\langle t_0,t_1,t_3,t_6,t_7,t_9,t_{13},t_{14},t_{15},t_{13},t_{16}\rangle$	1) 满足局部完备性; 2) 不存在“*aba*”模式; 3) 存在2度循环变体结构

如表 3 所示,该日志内容运用现有循环挖掘算法得不到图 7 中的原模型,根本原因在于所有轨迹中都不存在“*aba*”模式.

针对日志 W_5,采用本文算法却可以完整地挖掘出原模型.图 8 所示为α^{L+}算法的插件选择界面,图 9 为α^{L+}算法对 W_5 挖掘的最终结果(与原模型一致).此外,从过程挖掘官网(http://www.promtools.org/prom5/downloads/Process%20Log%20examples.zip)中下载的 8 个实例,算法都能挖掘出原有的过程模型.这也表明了算法在扩展经典 Alpha 算法后,不仅没有破坏原有算法挖掘顺序、选择和并发结构的能力,反而增强了挖掘2度循环的能力,对原有算法是一种很好的扩展.

Fig.8 Interface of α^{L+}-algorithm plug-in

图 8 α^{L+}算法插件界面

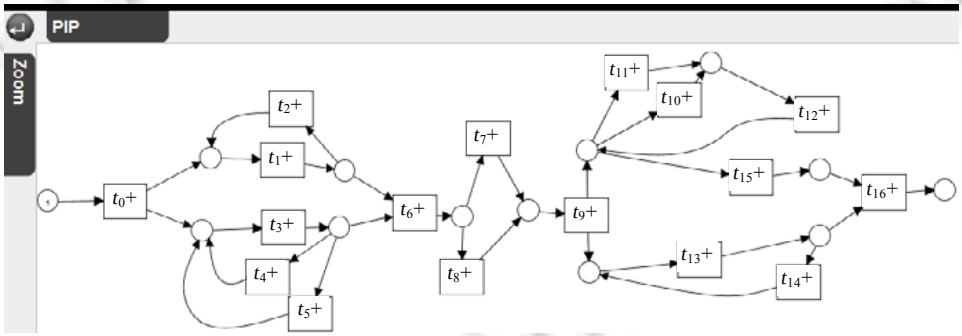


Fig.9 Mining result from the log W_5

图 9 日志 W_5 挖掘结果

4.2 算法评估

算法的评估内容包括 3 个方面.

a) 从时间复杂度上分析扩展后α^{L+}算法,其时间复杂度还是多项式时间.

b) 结合实际例子分析,通过多种循环挖掘算法对比测试,表明本文算法针对现实日志能够更好地挖掘出2度循环.

c) 在上述两者的基础上,深入探讨α^{L+}算法的核心内容,为下一步工作奠定基础.

1) 讨论算法的时间复杂度

由于现有的文献[2,10]已经很好说明了 Alpha 算法可以很好地在多项式时间内容挖掘出过程模型,本文是在此基础上进行了扩展,所以只给出扩展部分算法的时间复杂度分析.令 W 为满足局部完备性的日志,其中轨迹数量为 t,总任务数量为 n,日志事件总数量为 m,则日志抽象中两个主要步骤建立基本次序矩阵和循环比较次序向量,其时间复杂分别为 $O(m)$和 $O(n^2)$;事件特征挖掘中由于两种类型循环可以并行执行,所以仅考虑其中三角形 2 度循环结构的时间复杂度为 $O((n/2)\times m)$;并发分支结构干扰中采用紧邻度来解决的识别混乱问题时间复杂度为 $O((m/2)\times t)$;基于次序向量,采用广度优先搜索算法进行长循环抽象及还原,时间复杂度为 $O(n)$;扩展经典的 Alpha 算法中,还原变体结构的时间复杂度为 $O(n)$.由于α^{L+}算法是顺序执行上面步骤内容,所以整体复杂度为 $O(m+n^2+(n+t)\times m/2+n)$.更为重要的一点是:相对人工建模的业务过程时间(几周~月),采用过程挖掘算法从日志中得到的模型时间(几分钟~几十分钟)可以忽略不计.如图 9 的 W_5 日志在个人 PC 上的挖掘时间为 1ms,而针对较为复杂的实际日志(Log of Volvo IT incident management system,包含 7 554 条轨迹,65 533 个事件,来源 http://data.4tu.nl/repository/uuid:500573e6-accc-4b0c-9576-aa5468b10cee)在个人 PC 上的挖掘时间也只有十几分钟.因此,时间一般不是挖掘算法的瓶颈问题.

2) 讨论现有循环挖掘算法与本文算法的对比

首先给出过程模型中具有 2 度循环的挖掘准确率对比柱状图(图 10 所示),充分展示了本文算法相对原有算法的优势;其次,将现有 2 度循环挖掘的算法集成为一种,统称为集成算法,再与本文算法进行对比(图 11 所示),表明本文算法依然具有优势.

图 10 中,横坐标轴代表的是过程模型中任务的数量,并且长循环及 2 度循环的数量占任务数量的 10%;纵轴代表不同算法对应过程模型随机产生的局部完备性日志挖掘出循环的准确率.每种类型的模型,随机产生 200 个日志文件,然后依次采用现有算法进行挖掘.增量式挖掘算法针对 2 度循环挖掘能力最弱,基本上只能解决长循环结构,对于部分没有处于并发分支上的 2 度循环也能处理;Alpha+挖掘算法强制规定了日志轨迹必须同时存在“*aba*”与“*bab*”模式来满足循环完备性,以此挖掘 2 度循环,而模型复杂度增加会导致同时出现的概率大幅度下降;启发式是目前较为认可的方式来挖掘 2 度循环,即通过“*aba*”或“*bab*”出现的频率,设定一个阀值,满足阀值要求即可.其思想核心也是因为 2 度循环结构体现在日志中的事件特征是“*aba*”模式;本文算法α^{L+}在此基础上更进一步,考虑了事件特征不仅体现在紧邻模式上,更表现为事件间的位置及次数.正常分析,本文算法应该是随着模型规模增长,其准确率与其他算法会逐渐拉开差距,但是在规模为 100 个任务时,本文算法与启发算法基本上是一样的准确率.其主要原因是,紧邻度还不能完全保证百分百地排除识别混乱的问题;其次,模型产生的日志不含有“*aba*”模式的概率虽然是随着模型复杂度增加而降低,但并不是完全成正比关系.不过,从整体上分析,本文算法相对现有算法还是能够较好地解决 2 度循环挖掘问题.

3) 进一步讨论本文算法的内容

本文α^{L+}算法可以从含有或不含有“*aba*”模式的日志轨迹中挖掘出 2 度循环,但是还是不能做到百分之百的准确率.为此,针对现有的实验数据,逐步执行算法的每部分内容,分析当前算法每部分的贡献度及不足.贡献度是为了刻画α^{L+}算法在挖掘 2 度循环过程中,每部分核心内容对挖掘出最终 2 度循环的重要性.贡献度的计算为

$$CT=\frac{SW'}{SW}.$$

借鉴文献[18]用并行度来衡量挖掘后模型的效率问题,本文使用贡献度来衡量算法核心内容的重要性.其中,CT 为贡献度,分子 SW'表示算法核心内容单独或者与其他内容联合执行,能够挖掘出来 2 度循环的日志数量.而分母 SW 表示所有内容联合起来,挖掘出 2 度循环的日志的数量.

如图 11(a)所示,随着模型规模的增加,事件特征内容呈递减下降,而其他 3 部分内容呈缓慢递增上升.其反映了本文算法是以事件特征为核心,其他内容为辅的挖掘方法.在模型规模较少时,紧靠事件特征基本上就可以挖掘出 2 度循环;但随着模型的复杂度增加,变体结构、嵌套结构的出现,需要对日志进行预处理,才能保证日志轨迹也能很好地体现出 2 度循环的事件特征.图 11(b)针对 300 规模模型产生的数据,具体给出了算法中不同内容

的贡献度柱状图:除了 3 层以上的嵌套结构不进行探索以外,目前主要核心内容的不足体现在紧邻度.虽然紧邻度能够较好地解决识别混乱问题,但是其核心依据是真正2度循环的任务间紧邻度要比其他假的2度循环结的任务强.这种解决方式只是根据实际实验判断而形成的,存在一定的小概率事件.即存在识别混乱问题,依靠紧邻度还是不能解决.此外,多重嵌套结构也会造成变体抽象和回路抽象出现错误,从而导致挖掘的2度循环缺失.这部分内容已经在上述讨论中给出,这里不再赘述.综上所述,后续研究要具体针对这些不足进行深入探讨.

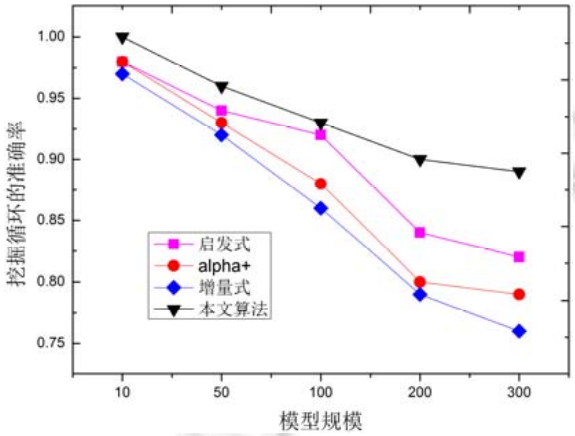
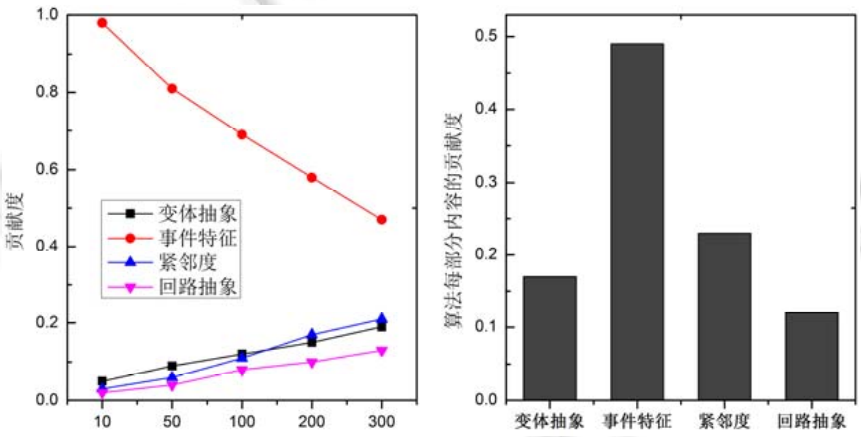


Fig.10 Comparison of precision among different algorithms

图 10 不同算法准确率对比



(a) 模型规模 (b) α^{L+}算法内容

Fig.11 Contribution of each part in α^{L+} algorithm

图 11 α^{L+}算法每部分的贡献度

5 相关工作

过程挖掘已经成为现代组织用于管理复杂运作流程的一种重要手段,其中,准确地挖掘出过程模型是该领域极其重要的研究内容.现有的算法在挖掘模型中的顺序关系、选择关系、并发关系及长循环关系中都获得了较好的进展.最大的难题是如何从日志文件中挖掘出长度为 2 的短循环结构.该内容的讨论及研究也从未停止过,表 4 给出了在挖掘循环结构上取得一定成果的文献内容(按时间降序).

Table 4 Researches about mining loop

表 4 循环挖掘的相关研究

文献	解决问题	技术方法	方式类别
文献[19]	短循环及长距离依赖	运用启发式定义循环因子	“*aba*”模式
文献[20]	不可见任务、循环和非自由网	遗传算法+启发式算法	“*aba*”模式
文献[21]	长循环及任务簇之间依赖关系	定义循环四元组,采用组合案例簇	无
文献[16]	长循环及部分简单短循环结构	增量式挖掘	无
文献[22]	短循环及噪音干扰问题	根据阀值,设定采用启发式挖掘算法	“*aba*”模式
文献[23]	长循环及部分并发干扰结构	定义四元组,采用启发式扩展α算法	无
文献[3]	短循环	启发式挖掘算法	“*aba*”模式
文献[10]	短循环	定义循环完备性的日志	“*aba*”模式

表 4 展示了循环结构挖掘的一些代表性成果,其中,短循环特指本文研究的 2 度循环结构,方式为模式代表其挖掘循环的内容用到了本文前面所提及的“*aba*”模式.从表 4 可以得出,现有的挖掘算法能够很好地解决长循环的挖掘问题[16,21,23].相比之下,短循环的结构挖掘是一个亟待解决的问题.为此,文献[10,19,20]分别提出了启发式因子、遗传+启发、扩展α算法及定义循环完备等方式来解决 2 度循环的挖掘难题.上述解决 2 度循环挖掘问题的本质在于采用了“*aba*”模式,但现实情况是,当日志满足局部完备性时,其日志轨迹中可以不出现“*aba*”模式.因此,本文在前人的工作基础上提出了α^{L^+}算法,用于解决从满足局部完备性且不含“*aba*”模式的日志轨迹中挖掘出 2 度循环.

当前,业务过程随着经济发展而变得相当庞大复杂,要想保证信息系统记录的日志能够完全反映任务间的关系(全局完备性),几乎是不可能的.已有较多的研究工作者提出了在更弱的完备性日志中挖掘出过程模型,如,文献[24,25]分别提出了因果完备性(causally complete)和弱完备性(weakly complete),但此类挖掘算法往往不能得到完整的模型(如并发结构和循环结构不能准确挖掘).因此,本文算法的优势在于不仅放宽日志的约束条件,又能保证准确地挖掘出完整的过程模型.

6 结束语

现有的过程挖掘算法在挖掘 2 度循环时,规定了日志中必须存在“*aba*”模式.但是,满足局部完备性的日志文件可以不存在该模式.为此,本文扩展了 Alpha 算法,从不具备“*aba*”模式的日志中挖掘出 2 度循环结构.本文开始给出了最简 2 度循环结构的定义,并通过次序向量对原始日志进行抽象,排除 2 度循环的变体结构;然后,通过全局视角,采用事件特征来挖掘两种不同的 2 度循环结构及并发结构;接着,针对并发分支上同类型循环结构带来的识别混乱和识别错误问题,提出了紧邻度和回路抽象的概念,较好地解决了以上问题;最后,借鉴前人工作的基础,提出了扩展算法α^{L^+}用于从不具备“*aba*”模式的日志中挖掘包含 2 度循环结构的完整过程模型.此外,运用人工案例及实际案例验证了α^{L^+}算法的可实现性,并通过实验的对比,探讨了目前算法的一些局限性.因此,后续研究也将在本文的基础上,对算法的不足进行更进一步的研究分析.另外,如何在不满足局部完备性的日志中挖掘出合理模型,以进一步提高算法在实际应用中的鲁棒性,也是未来工作的一个重点.

References:

[1] van der Aalst W. Process Mining: Discovery, Conformance and Enhancement of Business Proceses. Springer Publishing Company, Incorporated, 2014.

[2] van der Aalst W, Weijters T, Maruster L. Workflow mining: Discovering process models from event logs. IEEE Trans. on Knowledge & Data Engineering, 2004,16(9):1128−1142.

[3] Weijters AJMM, Aalst WMP, Medeiros AKA. Process mining with the heuristics miner algorithm. Eindhoven University of Technology, 2006,166:1−34.

[4] Medeiros AKAD, Weijters AJMM, Aalst WMPVD. Genetic process mining: An experimental evaluation. Data Mining & Knowledge Discovery, 2007,14(2):245−304.

[5] Sarno R, Sungkono KR. Hidden Markov model for process mining of parallel business processes. Int'l Review on Computers & Software, 2016,11(4):290−306.

[6] van der Werf JMEM, Dongen BFV, Hurkens CAJ, *et al*. Process discovery using integer linear programming. Fundamenta Informaticae, 2008,94(3):368−387.

[7] Bergenthum R, Desel J, Lorenz R, *et al*. Process mining based on regions of languages. In: Proc. of the Int'l Conf. on Business Process Management. Springer-Verlag, 2007. 375−383.

[8] Yang HD, Wen LJ, Wang JM. An approach to evaluate the local completeness of an event log. In: Proc. of the 12th IEEE Int'l Conf. on Data Mining (ICDM 2012). 2012. 1164−1169.

[9] Hofstede AHMT. Estimating completeness of event logs. Technical Report, No.04, BPM Center, 2012.

[10] Medeiros AKAD, Dongen BFV, Weijters AJMM. Process mining: Extending the α-algorithm to mine short loops. Eindhoven University of Technology, 2004,133:145−180.

[11] Yuan CY. Petri Net Application. Beijing: Science Press, 2103 (in Chinese).

[12] Günther CW. Activity mining by global trace segmentation. In: Proc. of the Int'l Workshops on Business Process Management Workshops (BPM 2009). Ulm: Revised Papers, 2009. 128−139.

[13] Polyvyanyy A, García-Bañuelos L, Dumas M. Structuring acyclic process models. Information Systems, 2010,37(6):518−538.

[14] Polyvyanyy A, García-Bañuelos L, Fahland D, Weske M. Maximal structuring of acyclic process models. The Computer Journal, 2014,57(1):12−35.

[15] Zhu R, Li T, Mo Q, Dai F, Gao TL, He Y, Sun X. Heuristic parallelized mining single firing sequence. Computer Integrated Manufacturing Systems, 2016,22(2):330−342 (in Chinese with English abstract).

[16] Ma H, Tang Y, Wu LK. Incremental mining of processes with loops. Int'l Journal on Artificial Intelligence Tools, 2011,20(01): 221−235.

[17] Van Dongen BF, De Medeiros AKA, Verbeek HMW, *et al*. The ProM framework: A new era in process mining tool support. In: Proc. of the Int'l Conf. on Applications and Theory of Petri Nets. Springer-Verlag, 2005. 444−454.

[18] Jin T, Wang J, Yang Y, Wen L, Li K. Refactor business process models with maximized parallelism. IEEE Trans. on Services Computing, 2016,9(3):456−468.

[19] Lu FM, Zeng QT, Duan H, Cheng JJ, Bao YX. College of information science and engineering parallelized heuristic process mining algorithm. Ruan Jian Xue Bao/Journal of Software, 2015,26(3):533−549 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4769.htm [doi: 10.13328/j.cnki.jos.004769]

[20] Vázquez-Barreiros B, Mucientes M, Lama M. ProDiGen: Mining complete, precise and minimal structure process models with a genetic algorithm. Information Sciences, 2015,294:315−333.

[21] Lu FM, Zeng QT, Bao YX, Duan H, Zhang H. Mining algorithm of task dependencies based on process case clusters. Computer Integrated Manufacturing Systems, 2013,19(8):1771−1783 (in Chinese with English abstract).

[22] Wang HY. A process mining algorithm for cycle tasks [MS. Thesis]. Tianjin: Hebei University of Technology, 2011 (in Chinese with English abstract).

[23] Wu S. An extended alpha mining algorithm for complex loop structures [MS. Thesis]. Harbin: Harbin Engineering University, 2011 (in Chinese with English abstract).

[24] Lekić J, Milićev D. Discovering block–structured parallel process models from causally complete event logs. Journal of Electrical Engineering, 2016,67(2):111−123.

[25] Lekic J, Milicev D. Discovering models of parallel workflow processes from incomplete event logs. In: Proc. of the Int'l Conf. on Model-Driven Engineering and Software Development. IEEE, 2015. 477−482.

附中文参考文献:

[11] 袁崇义.Petri 网应用.北京:科学出版社,2013.

[15] 朱锐,李彤,莫启,代飞,高提雷,何云,孙雪.启发式并行化单触发序列挖掘算法.计算机集成制造系统,2016,22(2):330−342.

[19] 鲁法明,曾庆田,段华,程久军,包云霞.一种并行化的启发式流程挖掘算法.软件学报,2015,26(3):533−549. http://www.jos.org.cn/1000-9825/4769.htm [doi: 10.13328/j.cnki.jos.004769]

[21] 鲁法明,曾庆田,包云霞,段华,张昊.基于流程案例簇的任务关系挖掘算法.计算机集成制造系统,2013,19(8):1771−1783.

[22] 王海燕.面向循环任务的过程挖掘算法研究[硕士学位论文].天津:河北工业大学,2011.

[23] 吴苏.一种可发现复杂循环结构的扩展α过程挖掘算法[硕士学位论文].哈尔滨:哈尔滨工程大学,2011.

林雷蕾(1989－),男,海南万宁人,硕士,主要研究领域为软件工程,流程管理,系统分析与集成.

周华(1963－),男,博士,研究员,博士生导师,主要研究领域为软件工程,系统分析与集成.

代飞(1982－),男,博士,副教授,CCF 专业会员,主要研究领域为软件工程,业务过程管理.

朱锐(1987－),男,博士,讲师,CCF 专业会员,主要研究领域为软件过程,过程挖掘.

李彤(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程.