

个人阅读报告

郑先淇 16340305

论文题目

A Method for Repairing Process Models Containing a Choice With Concurrency Structure by Using Logic Petri Nets

论文作者及单位

Zheng, WT (Zheng, Wentai); Du, YY (Du, Yuyue); Qi, L (Qi, Liang); Wang, L (Wang, Lu)

发表期刊信息

期刊名称: IEEE ACCESS

卷: 7

页码: 13106-13120

出版年: 2019

技术问题

过程挖掘领域的三大主要研究方向: 过程发现、一致性验证和过程增强。过程发现技术以事件日志作为输入, 产生一个能够最好地符合日志中所描述行为的过程模型。就当前来说, 有很多很多的过程发现算法已经被提出, 例如, α 算法、 α^+ 算法、 α^* 算法等。然而, 普遍来讲, 当待挖掘的过程模型非常复杂的时候, 这些已被提出的过程发现算法往往不能建立正确的过程模型。这种情况下, 过程增强的概念被提出。过程增强技术通过参照事件日志, 来对过程发现过程得到的初始模型进行扩展或改进。在本文作者发表此篇论文之前, 过程增强的两个主要方式为:

- 1、找出事件日志中没有被成功建模的部分, 生成对应的子日志, 使用过程发现算法对子日志进行过程建模, 最后将初始的过程模型与新生成的子模型进行拼接, 得到能够完整符合全部事件日志的过程模型;

- 2、在初始过程模型加上一些自循环的结构和不可见的 transition, 使其能够良好地符合事件日志;

然而, 以上提到的两种过程增强技术都存在一些弊端。对于方式 1, 虽然得到的过程模型能够很好地描述事件日志中的行为, 但是子模型和初始模型进行拼接的方法可能导致很多的重复节点, 使得过程模型变得冗余和复杂; 对于方式 2, 添加自循环结构之后也不一定能够很好的描述事件日志的行为, 另外, 自添加的 transition 节点会导致过程模型不够简洁和准确度下降。

现实背景

目前来看, 面向任意复杂模型的通用过程增强技术尚未有很大的进展。因此, 本文针对于一类特定的复杂过程模型——选择结构中包含并行结构的过程模型,

提出一个基于扩展 petri 网的过程增强技术方案。此过程增强方案不向初始过程模型添加任何的 place 和 transition，避免由此带来的模型过程结构冗余和准确度下降。通过编写查询算法，可以快速地查找到初始模型与事件日志产生分歧的地方。通过向某些 transition 添加一些逻辑函数和增加一些有向边，增强模型能够很好地描述事件日志的行为。

作者思路(idea)

作者首先通过编写一个对齐(alignment)算法找到初始过程模型与事件日志产生分歧的地方，然后通过扩展 petri 网(logic petri net)来增强对过程模型中并发与选择结构的表达能力，再通过为相关的节点添加有向边和为特定的 transition 添加 logic function，来清晰地表达有关节点的逻辑关系，以此增强初始模型的业务流程表达能力，使其与事件日志行为更好地契合。

解决方案

针对于在选择结构中包涵着并发结构的复杂过程模型，本文提出了一个基础 logic Petri net 的完备增强方案，主要分为以下四个步骤：

1. 根据初始过程模型构造一棵 process tree, 遍历 process tree 找到模型中所有的 choice recognition pairs(Crp), 算法如下：

Algorithm 1 Choice Recognition Pairs
Input: The node of process tree PT denoted by $subnode$, the leaf node of PT denoted by K and $PN = (P, T; F, m)$
Output: The set of choice recognition pairs denoted by $CRPS$
1. $crp(p_i, p_j) \leftarrow \emptyset, CRPS \leftarrow \emptyset;$
2. For all the $subnode \in PT$ do
3. If ($subnode = "x"$) then
4. If ($subnode \in K$) do
5. $crp \leftarrow crp \cup \{ \bullet(LLN(K)), (RLN(K)) \bullet \};$
6. RETURN $CRPS \leftarrow CRPS \cup \{ crp = (p_i, p_j) \}.$

Crp 是一个二元组，其两个元素是由 Process tree 中的 x(选择符号)子节点作为根节点的子树的最左叶节点(LLN(k))和最右叶节点(RLN(k))。

2. 根据 logic simplest alignment 算法找到初始过程模型和事件日志的 deviation, 算法如下：

Algorithm 2 An Algorithm to Discover the Deviation

Input: An event log $L \in \mathcal{B}(A^*)$, the simplest alignment β and $PN = (P, T; F, m)$

Output: Deviation position $Dev(t_m, t_n)$

1. $Dev(p_m, p_n) \leftarrow \emptyset$;
 2. Traverse the simplest alignment β , if $(a_i \in A \text{ and } t_i = \gg)$ then
 3. Record the location of the transition when there is a place that is missing token;
 4. While $m_i \in M$ do
 5. If $\neg p$ is in a reachable marking and appears multiple times in H ;
 6. $t_m \leftarrow \bullet p, t_n \leftarrow p^\bullet$ and $t_n \neq LLN(K)$;
 7. RETURN $Dev(t_m, t_n) \leftarrow (t_m, t_n)$.
-

由于内容很多，有关于 logic simplest alignment 算法的具体步骤就不在此赘述，不过明确一点，通过 logic simplest alignment 算法，我们可以准确地定位到初始过程模型中于事件日志产生差异的地方。

3. 遍历 simplest alignment 表格，每次发生一个 log move 的时候，判断 crp 与 $dev(tm, tn)$ 的关系，当符合条件时为初始过程模型添加有向边，具体算法如下：

4. Traversing the simplest alignment β , if $(a_i \in A \text{ and } t_i = \gg)$ then
5. If $t_m^\bullet = p_i, t_i = (t_m^\bullet)^\bullet$ and $t_i \neq LLN(K)$;
6. $F' \leftarrow F' \cup \{(LLN(K))^\bullet, t_i\}$;
7. If $t_n^\bullet = p_j$ and $t_n \neq RLN(K)$;
8. $F' \leftarrow F' \cup \{RLN(K), \bullet t_n\}$;

4. 通过算法 3 对步骤三中添加的有向边附加逻辑函数，算法 3 如下：

Algorithm 3 Log-Based Logical Transition Algorithm

Input: An event log $L \in \mathcal{B}(A^*)$, $0 < m < n < 1$, k_1 and k_2

Output: Log-based logical transition relationship $lr(a, b)$

1. $lr(a, b) \leftarrow \emptyset$;
 2. For each trace $\in L'$ do;
 3. If $k_1 + k_2 < m$ then
 4. $lr(a, b) \leftarrow lr(a, b) \cup (a \otimes b)$;
 5. If $(k_1 > n \text{ and } k_2 < m)$ then
 6. $lr(a, b) \leftarrow lr(a, b) \cup \{a \rightarrow b \mid (b \leftarrow a)\}$;
 7. If $(k_1 + k_2 > n \text{ and } \min\{k_1, k_2\} > m)$ then
 8. $lr(a, b) \leftarrow lr(a, b) \cup (a \wedge b)$;
 9. Else $lr(a, b) \leftarrow lr(a, b) \cup (a \vee b)$;
 10. RETURN $lr(a, b)$.
-

综合步骤 1 到 4，完整的过程增强算法如下：

Algorithm 4 Model Repair Algorithm Based on Logic Petri Net

Input: An event log $L \in \mathcal{B}(A^*)$, $PN = (P, T; F, m)$, and the simplest alignment β

Output: The repaired logic Petri net $LPN = (P, T; F, I, O, m)$

1. $LPN \leftarrow N, t_i \leftarrow \emptyset$;
 2. Using Algorithm 1 to get the $CRPS = \{(p_i, p_j) | p_i = \bullet(LLN(K)), p_j = (RLN(K))^\bullet, \forall K \in PT\}$;
 3. Using Algorithm 2 to get $Dev(t_m, t_n)$;
 4. Traversing the simplest alignment β , if $(a_i \in A$ and $t_i = \gg)$ then
 5. If $t_m^\bullet = p_i, t_i = (t_m^\bullet)^\bullet$ and $t_i \neq LLN(K)$;
 6. $F' \leftarrow F' \cup \{(LLN(K))^\bullet, t_i\}$;
 7. If $t_n^\bullet = p_j$ and $t_n \neq RLN(K)$;
 8. $F' \leftarrow F' \cup \{RLN(K), t_n^\bullet\}$;
 9. Using Algorithm 3 to get logic transitions with logic functions;
 10. RETURN $LPN = (P, T; F, I, O, m)$.
-

创新贡献

个人认为，我阅读的这篇论文的创新贡献主要包含以下三点：

1. 提出一个 logic simplest alignment 算法，能够快速找到初始过程模型与事件日志产生分歧的地方并于过程模型中准确定位；
2. 对经典 petri 网进行扩展，设计 logic petri net 来增强经典 petri 网对于选择结构和并发结构的表达能力；
3. 设计了一个通用的算法将过程模型中选择与并发结构嵌套的复杂业务关系公式化，定义了添加有向边和 logic function 的规则，使得面对任何有关选择与并发结构嵌套的复杂业务关系时，均可简便地通过该算法来梳理出清晰的业务流程关系。

效果评价

效果： 我们使用衡量过程模型的四个标准 (*precision*、*fitness*、*simplicity*、*generalization*) 来衡量该解决方案的效果，以 *precision* 和 *simplicity* 为例，本文提到的解决方案与其他几种常用的解决方案的对比如下：

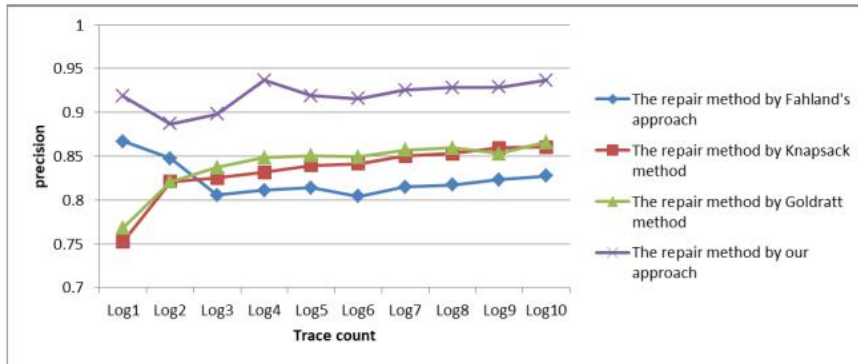


FIGURE 21. The precision value between our method and other methods.

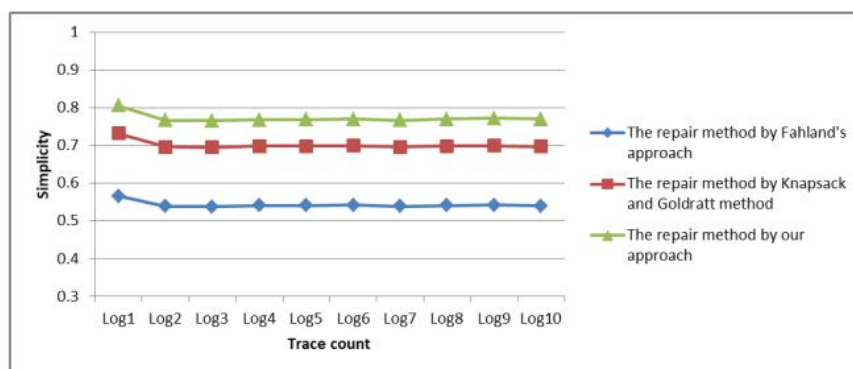


FIGURE 22. The simplicity value between our method and other methods.

由以上结果可看到本文提出的解决方案在 precision、simlicity 方面均有明显的性能提升。

优点：基于 logic petri net 的解决方案在 precision、simlicity 等方面均有明显的性能提升。

缺点：泛化能力比较一般，只在解决选择结构与并发结构相互嵌套的情况下能够取得比较好的效果。

约束条件：只能用于选择与并发结构嵌套的情况，不适合其他复杂的业务流程情形。

个人观点

这篇论文讲述的是过程挖掘领域中与过程增强技术相关的问题。文章以扩展 Petri 网为核心，针对复杂过程模型中的选择结构与并发结构相嵌套的这个点，提出了 Logic simplest alignment 算法和添加有向边和 Logic 函数的算法，从而构造出一个完备的性能良好的解决方案。个人觉得，论文中提出的利用 alignment 查找 deviation 的思想是很清晰且易懂的，同时也取得了很好的效果，这是我觉得这篇论文里很出彩的地方，即我认为此篇论文很好地解决了差异定位的问题。但是，这篇论文没有讲述通过算法 4 添加有向边的原理，让人很难去判断该通过该算法添加有向边是否具有合理性。另外，对于如何定义有向边对应的 logic function，这篇文章也讲述地很模糊，对于有些变量、有些参数没有事先说明就使用，对于算法的输入输出都没有做比较仔细の説明，有一点遇到了困难没有克服它反而是水过去的嫌疑。不过，退一步来说，作者通过仅仅增加一些有向边和逻辑函数来进行模型增强的思想在我看来的确是很好的，因为这样能够最大程度上保持初始模型的准确度。最后，对于论文提出的这个解决方案，我觉得它还有很大的发展空间。作者虽然对经典 petri 网做了一定程度上的扩展，但是并没有修改经典 Petri 网的规则，所以这个扩展的 petri 网是向下兼容的，所以很多很多有关于经典 petri 网的研究成果依旧适用于这个扩展的 petri 网，如果作者能够在 logic function 和有向边的算法改进多下功夫的话，我相信肯定能提出一个更加优秀的解决方案。