# Discovering Models of Parallel Workflow Processes from Incomplete Event Logs

Julijana Lekic[1] and Dragan Milicev[2]
[1]University of Pristina in Kosovska Mitrovica, Faculty of Technical Sciences, Kneza Milosa 7, 38220 Kosovska Mitrovica, Serbia
[2]University of Belgrade, Faculty of Electrical Engineering, Bulevar kralja Aleksandra 73, 11120 Beograd, Serbia

Keywords: Process Mining, Process Model Discovery, Parallel Business Processes, Incomplete Event Log, $\alpha$-algorithm.

Abstract: $\alpha$-algorithm is able to discover a large class of workflow (WF) nets based on the behavior recorded in event logs, with the main limiting assumption that the event log is complete. Our research has been aimed at finding ways of business process models discovering based on examples of traces, i.e., logs of workflow actions that do not meet the requirement of completeness. In this aim, we have modified the existing and introduced a new relation between activities recorded in the event log, which has led to a partial correction of the process models discovering techniques, including the $\alpha$-algorithm. We have also introduced the notions of causally and weakly complete logs, from which our modified algorithm can produce the same result as the original algorithm from complete logs. The effect of these modifications on the speed of the process model discovering is mostly evident for business processes in which many activities can be performed in parallel. Therefore, this paper presents preliminary results obtained from the investigation of opportunities to discover models of parallel processes based on incomplete event logs.

## 1 INTRODUCTION

Business process models play an important role in large enterprises. Process mining (PM) techniques help them to identify models of their actual business processes based on examples of behavior, i.e., on logs of performed activities. In process mining, an event log is used for the implementation of three techniques: *discovering*, described by Aalst van der, Weijters and Maruster (2002), Aalst van der, Weijters and Maruster (2004), Aalst van der (2011: 125-139), *conformance*, described by Rozinat and Aalst van der (2008), Aalst van der (2011: 191-213), and *enhancement* of the model, described by Aalst van der (2011: 281-305).

A technique of interest in our research is forementioned model discovering. This technique is used to construct models of business processes only on the basis of examples of behavior exhibited by the processes i.e., traces recorded in the event log. One of the basic and best known algorithms for process model discovering, based on records in a log, is the $\alpha$-algorithm introduced by Aalst van der, Weijters and Maruster (2002: 16), Aalst van der, Weijters and Maruster (2004: 1135), Aalst van der (2011: 133). From records in the event log, the $\alpha$-algorithm automatically generates a process model that belongs to a subclass of Petri nets, according to Aalst van der and Stahl (2011) known as workflow (WF) nets.

Besides a number of other limitations, one of the basic limiting assumptions of this algorithm is that the event log needs to be complete. In addition, the assumption of event log completeness requires that all activities that potentially directly follow each other, directly follow each other in some trace in the log as it is defined by Aalst van der, Weijters and Maruster (2002: 9), Aalst van der, Weijters and Maruster (2004: 1132). The assumption of the completeness of the event log requires that in the traces recorded in it, there are all direct dependency relations that may exist between the activities of the observed business process. The property of completeness of the log often requires a large number of traces in the log on which the "representative" model for the behavior seen in the log has to be constructed. Therefore, our challenge was to find logs with potentially much lower number of traces, which may not be complete, but are

Lekic J. and Milicev D..
Discovering Models of Parallel Workflow Processes from Incomplete Event Logs.
DOI: 10.5220/0005242704770482
In *Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development* (MODELSWARD-2015), pages 477-482
ISBN: 978-989-758-083-3

sufficiently valid so that, using the appropriate algorithm based on the evidences recorded in such logs, a "representative" model can be obtained.

To achieve this, we have partially modified the technique of process model discovering, and also the $\alpha$-algorithm itself by introducing the relation of indirection as another basic relation between the activities recorded in the event log. Our aim was to determine how the existence of the relation of indirection affects detection of concurrency relationships, and how it affects the efficiency of discovering the business process model. The preliminary results have shown that, by a generalization of the concurrency relation, considering the relations of indirection between the activities recorded in the event log, the concurrency relation, and therefore business process models, can be obtained from smaller, i.e., incomplete event logs, which enables faster detection of business processes models. This shows particularly good results in processes with a lot of activities that can be performed in a mutually independent order, i.e., concurrently. In order to illustrate the effects of using the proposed modified technique of discovering process models and make it more noticeable, the results of applying the modified PM technique for discovering a parallel processes model will be shown on sample processes.

The paper is organized as follows. The next section describes: the proposed modification of the PM technique for business processes model discovering, the concept of a parallel business process and the proposed partial modification of the $\alpha$-algorithm ($\alpha^{\|}$-algorithm). Section 3 presents the application of the proposed modified PM techniques to discovering parallel business process models based on the so called *causally complete* and *weakly complete* logs. Section 4 brings conclusions and guidelines for a future work.

## 2 MODIFIED TECHNIQUE FOR DISCOVERING PROCESS MODELS

The subject of our research, presented in this paper, is the problem of completeness of the event log, as it was presented by Aalst van der (2011: 147). In the basic $\alpha$-algorithm, it is assumed that an event log $L$ is complete in terms of the relation $>_L$, which was basic assumption established by Aalst van der, Weijters and Maruster (2002: 9), Aalst van der, Weijters and Maruster (2004: 1132). This assumption about the completeness of the log $L$ requires that if a process model allows an activity $b$ to be executed immediately after an activity $a$, then the log must contains at least one trace where $b$ is really executed immediately after $a$, i.e. $a >_L b$ must hold. In this case, $b$ is said to be *directly* (or *immediately*) following $a$ in $L$, as it was defined by Aalst van der, Weijters and Maruster (2002: 8), Aalst van der (2011: 130).

Considering the fact that the existence of the relation of direct following in the event log affects the definition of the relation of parallelism ($a \|_L b$ iff $a >_L b$ and $b >_L a$), according to the definition introduced by Aalst van der, Weijters and Maruster (2002: 8), Aalst van der, Weijters and Maruster (2004: 1132), Aalst van der (2011: 130), our aim was to examine how the existence of the relation of *indirect* following between activity affects the inference of the relation of parallelism, and how it reflects on the efficiency of discovering process models. Due to this reason, we have introduced a new basic relation $a >>_L b$, which indicates that $b$ *indirectly follows* $a$. The log-based relations that are used to indicate the relevant patterns in the log in our modified technique of detecting process models are defined by Definition 1 below.

According to Aalst van der (2011: 129) the event log $L$ is observed as a multiset of traces over a set of actions $\mathcal{A}$, i.e., $L \in \mathbb{B}(\mathcal{A}^*)$. Each trace corresponds to a single executed process case (or scenario). The elements of the set $\mathcal{A}$ are the activities that correspond to transitions in the resulting Petri net, and are denoted by lowercase letters (i.e., $a$, $b$, $c$, ... $\in \mathcal{A}$), while the sets of activities are denoted by uppercase letters (i.e., $A$, $B$, $C$, ... $\subseteq \mathcal{A}$ ).

**Definition 1.** (Log-based ordering relations, in the modified PM technique of discovering process models). Let $L$ be an event log over $\mathcal{A}$, i.e., $L \in (\mathcal{A}^*)$. Let $a, b \in \mathcal{A}$ be two activities. Then, by definition:

- $a >_L b$ if and only if there is a trace $\sigma = \langle t_1, t_2, t_3, \ldots, t_n \rangle$ and $i \in \{1, \ldots, n-1\}$ such that $\sigma \in L$ and $t_i = a$ and $t_{i+1} = b$
- $a >>_L b$ if and only if there is a trace $\sigma = \langle t_1, t_2, t_3, \ldots, t_n \rangle$ and there are $i, j \in \{1, \ldots, n\}$ such that $i + 2 \leq j$, where $\sigma \in L$ and $t_i = a$, $t_j = b$, and it is not that $a >_L b$
- $a \rightarrow_L b$ if and only if $a >_L b$, and it is not $b >_L a$, and it is not $b >>_L a$
- $a \Rightarrow_L b$ if and only if $a >>_L b$, and it is not $b >_L a$, and it is not $b >>_L a$
- $a \#_L b$ if and only if it is not $a >_L b$, and it is not $b >_L a$, and it is not $a >>_L b$, and it is not $b >>_L a$

- $a \|_L b$ if and only if $a >_L b$ and $b >_L a$, or $a >_L b$ and $b >>_L a$, or $a >>_L b$ and $b >_L a$, or $a >>_L b$ and $b >>_L a$.

These relations can be represented with a matrix, which is the footprint of the event log, as it will be shown in the later examples.

## 2.1 a$^{\|}$-algoritam

The introduction of the relation of indirection, as an additional basic relation between activities recorded in the log, has led to a significantly faster discovering of the concurrency relations between the activities. The basic idea is the following: in order to conclude that two activities $a$ and $b$ are mutually independent, i.e., concurrent, it is enough to have a trace in the event log in which $a$ directly or *indirectly* precedes $b$, and another trace where $b$ directly or *indirectly* precedes $a$. Discovering concurrency relations from traces recorded in the event log is one of the key elements to discovering process models. Since concurrency is very important in real business processes, in which many things may occur simultaneously, our research has primarily been focused on the effects of introducing the relation of indirection to the speed of obtaining business process models in which many activities are performed in parallel. This is why we have defined a particular subclass of business processes of interest, *parallel processes*.

The first assumption is that, in parallel processes, all the activities perform either sequentially or in parallel, but not in alternative (optionally). The second assumption is that during the execution of a single instance of the process, each activity can be performed at most once, which eliminates the possibility of iterations in parallel process models. For the presentation of the model, we will use a variant of the classical Petri nets called *sound* WF-nets, which belong to the set of WF-nets, denoted with W, in accordance with Aalst van der, Weijters and Maruster (2002: 7). Figure 1 shows our running example of a parallel process.

The assumptions defined above led to a lack of the relation $\#_L$ defined by Aalst van der, Weijters and Maruster (2002: 8), Aalst van der, Weijters and Maruster (2004: 1132), Aalst van der (2011: 130), in a part which referred to relations between different activities of parallel processes. Since the $\alpha$-algorithm is based on the relations $\#_L$ and $\rightarrow_L$ (as it was cited previously), the loss of the relation $\#_L$ between different activities will lead to changes in the $\alpha$-algorithm and its application to the discovery of a parallel process model.

That modified $\alpha$-algorithm will be denoted with $\alpha^{\|}(L)$, where „$^{\|}$“ reflects the fact that the algorithm targets parallel business processes, and is defined as follows:

**Definition 2.** ($\alpha^{\|}$-algorithm). Let $L$ be an event log over $T \subseteq \mathcal{A}$. $\alpha^{\|}(L)$ is defined as follows:

(1) $T_L = \{t \in T \mid (\exists \sigma \in L)\ t \in \sigma\}$

(2) $T_I = \{t \in T \mid (\exists \sigma \in L)\ t = first(\sigma)\}$

(3) $T_O = \{t \in T \mid (\exists \sigma \in L)\ t = last(\sigma)\}$

(4) $X_L = \{(A, B) \mid A \subseteq T_L \wedge A \neq \emptyset \wedge B \subseteq T_L \wedge B \neq \emptyset \wedge (\forall a \in A)\ (\forall b \in B)\ (a \rightarrow_L b)\}$

(5) $P_L = \{p_{(A, B)} \mid (A, B) \in X_L\} \cup \{i_L, o_L\}$

(6) $F_L = \{(a, p_{(A, B)}) \mid (A, B) \in X_L \wedge a \in A\} \cup \{(p_{(A, B)}, b) \mid (A, B) \in X_L \wedge b \in B\} \cup \{(i_L, t) \mid t \in T_I\} \cup \{(t, o_L) \mid t \in T_O\}$

(7) $\alpha^{\|}(L) = (P_L, T_L, F_L)$

# 3 APPLYING THE MODIFIED TECHNIQUE FOR DISCOVERING PARALLEL PROCESS MODELS FROM INCOMPLETE EVENT LOGS

In PM techniques for model discovering (as it was cited previously), a necessary condition for discovering the original network by the $\alpha$-algorithm is that the log on which the algorithm is implemented needs to be complete, where the condition of completeness is based on the relation $>_L$. The condition of completeness in our modified PM technique for model discovering is related to the causality relation $\rightarrow_L$, and accordingly new types of completeness are defined: causal completeness and weak completeness, which will be defined in this section.

## 3.1 Discovering Original Networks from Causally Complete Logs

For a particular process model to be discovered, there may be a large (in general, an unlimited) number of different complete logs. However, all these complete logs have the same footprint, i.e., the same causality relation. We call this relation the *basic causality relation*. On the other hand, there may exist other logs for the same process model that are not complete, but which have the same footprint, i.e., the same causality relation obtained from those logs. We are focused on investigating such logs, which we refer to as *causally complete logs* ($L_c$). Obviously, the idea is to find causally complete logs

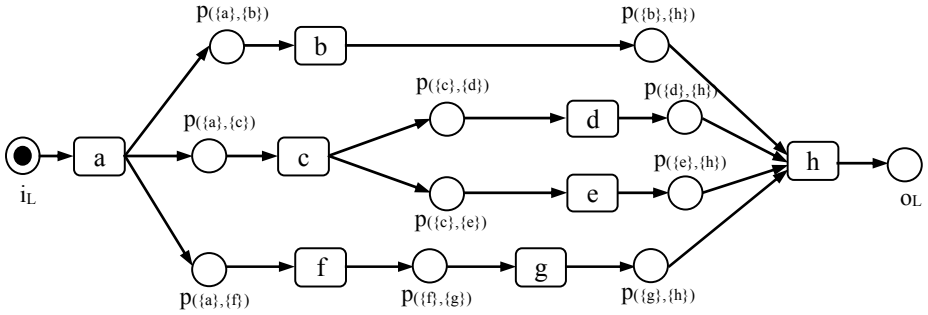


Figure 1: Example of a parallel process model.

that may be, in general, much smaller than fully complete logs (in the terminology of the original $\alpha$-algorithm).

**Definition 3.** (The basic causality relation) Let $N = (P, T, F)$ be a sound WF-net, i.e., $N \in \mathcal{W}$, and let $L$ be a complete workflow log of $N$. $\rightarrow^{B}_{N}$ is the *basic causality relation* of network $N$ iff $\rightarrow^{B}_{N} = \rightarrow_{L}$.

Considering the so defined basic causality relation, a causally complete log is defined as follows.

**Definition 4.** (Causally complete log) Let $N = (P, T, F)$ be a sound WF-net, i.e., $N \in \mathcal{W}$, and let $\rightarrow^{B}_{N}$ be the basic causality relation of $N$. $L_c$ is a *causally complete workflow log* of $N$ iff:

1) $\rightarrow_{Lc} = \rightarrow^{B}_{N}$, and

2) for any $t \in T$ there is $\sigma \in L_c$ so that $t \in \sigma$.

From Definition 2 it can be seen that the $\alpha^{\|}$-algorithm is based on the causality relation. Preliminary results have shown that by the modified PM technique and the $\alpha^{\|}$-algorithm, the network can be rediscovered from any incomplete log in which $\rightarrow_{L} = \rightarrow^{B}_{N}$, i.e., from any causally complete log, as the example that follows will show.

Let us observe a parallel process model shown in Figure 1, and a log $L_1$ obtained after several executions of the process.

$L_1 = [< a, b, c, d, e, f, g, h >^4, < a, f, g, b, c, e, d, h >^2, < a, c, d, e, f, g, b, h >, < a, b, f, g, c, d, e, h >]$

The basic causality relation for this example is:

$\rightarrow^{B}_{N} = \{(a, b), (a, c), (a, f), (b, h), (c, d), (c, e), (d, h), (e, h), (f, g), (g, h)\}$

The footprint of the event log $L_1$ is given in Table 1.

From the footprint shown in Table 1, it can be seen that the causality relation of $L_1$ is:

$\rightarrow_{L1} = \{(a, b), (a, c), (a, f), (b, h), (c, d), (c, e), (d, h), (e, h), (f, g), (g, h)\}$

It can be observed that the causality relation of $L_1$ is equal to the basic causality relation, i.e., $\rightarrow_{L1} = \rightarrow^{B}_{N}$, which makes the log $L_1$ causally complete.

By applying the $\alpha^{\|}$-algorithm to the given log $L_1$, we obtain the following:

(1) $T_{L1} = \{a, b, c, d, e, f, g, h\}$

(2) $i_{L1} = a$

(3) $o_{L1} = h$

(4) $X_{L1} = \{(\{a\},\{b\}), (\{a\},\{c\}), (\{a\},\{f\}), (\{b\},\{h\}), (\{c\},\{d\}), (\{c\},\{e\}), (\{d\},\{h\}), (\{e\},\{h\}), (\{f\},\{g\}), (\{g\},\{h\})\}$

(5) $P_{L1} = \{p_{(\{a\},\{b\})}, p_{(\{a\},\{c\})}, p_{(\{a\},\{f\})}, p_{(\{b\},\{h\})}, p_{(\{c\},\{d\})}, p_{(\{c\},\{e\})}, p_{(\{d\},\{h\})}, p_{(\{e\},\{h\})}, p_{(\{f\},\{g\})}, p_{(\{g\},\{h\})}, i_{L1}, o_{L1}\}$

(6) $F_{L1} = \{(a, p_{(\{a\},\{b\})}), (p_{(\{a\},\{b\})}, b), (a, p_{(\{a\},\{c\})}), (p_{(\{a\},\{c\})}, c), (a, p_{(\{a\},\{f\})}), (p_{(\{a\},\{f\})}, f), (b, p_{(\{b\},\{h\})}), (p_{(\{b\},\{h\})}, h), (c, p_{(\{c\},\{d\})}), (p_{(\{c\},\{d\})}, d), (c, p_{(\{c\},\{e\})}), (p_{(\{c\},\{e\})}, e), (d, p_{(\{d\},\{h\})}), (p_{(\{d\},\{h\})}, h), (e, p_{(\{e\},\{h\})}), (p_{(\{e\},\{h\})}, h), (f, p_{(\{f\},\{g\})}), (p_{(\{f\},\{g\})}, g), (g, p_{(\{g\},\{h\})}), (p_{(\{g\},\{h\})}, h), (i_{L1}, a), (h, o_{L1})\}$

(7) $\alpha^{\|}(L_1) = (P_{L1}, T_{L1}, F_{L1})$

Table 1: Footprint of the event log $L_1$.

| | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ | $h$ |
|---|---|---|---|---|---|---|---|---|
| $a$ | # | $\rightarrow$ | $\rightarrow$ | $\Rightarrow$ | $\Rightarrow$ | $\rightarrow$ | $\Rightarrow$ | $\Rightarrow$ |
| $b$ | $\leftarrow$ | # | $\|\|$ | $\|\|$ | $\|\|$ | $\|\|$ | $\|\|$ | $\rightarrow$ |
| $c$ | $\leftarrow$ | $\|\|$ | # | $\rightarrow$ | $\rightarrow$ | $\|\|$ | $\|\|$ | $\Rightarrow$ |
| $d$ | $\Leftarrow$ | $\|\|$ | $\leftarrow$ | # | $\|\|$ | $\|\|$ | $\|\|$ | $\rightarrow$ |
| $e$ | $\Leftarrow$ | $\|\|$ | $\leftarrow$ | $\|\|$ | # | $\|\|$ | $\|\|$ | $\rightarrow$ |
| $f$ | $\leftarrow$ | $\|\|$ | $\|\|$ | $\|\|$ | $\|\|$ | # | $\rightarrow$ | $\Rightarrow$ |
| $g$ | $\Leftarrow$ | $\|\|$ | $\|\|$ | $\|\|$ | $\|\|$ | $\leftarrow$ | # | $\rightarrow$ |
| $h$ | $\Leftarrow$ | $\leftarrow$ | $\Leftarrow$ | $\leftarrow$ | $\leftarrow$ | $\Leftarrow$ | $\leftarrow$ | # |

The network $N^{\|} = \alpha^{\|}(L_1)$ obtained by applying the $\alpha^{\|}$-algorithm to the log $L_1$ is equal to what is shown in Figure 1. The log $L_1$ is not complete, because there are missing elements in the relation $>_L$: $b > d$, $b > e$, $b > g$, $c > b$, $c > f$, $c > g$, $d > b$, $d > f$, $d > g$, $e > b$, $e > g$, $f > b$, $f > c$, $f > d$, $f > e$, $g > d$ and $g > e$, which could be potentially performed on the

basis of the process model given in Figure 1, and the resulting WF-net $N^{\|}$.

## 3.2 Discovering Original Networks from Weakly Complete Logs

From the said above, it can be concluded that the main task is to find a log with the causality relation that is equal to the basic causality relation, and then apply the $\alpha^{\|}$-algorithm, which then leads to the original network of a parallel processes. During our research, we have found that there are logs from which one can discover a causality relation equal to $\rightarrow^{B}_{N}$, but one cannot come to it only from the evidence recorded in the log, but the individual elements of the causality relation can be inferred in the process of applying the $\alpha^{\|}$-algorithm of them. We call a log with such properties a *weakly complete* log ($L_w$). Weakly complete logs can be significantly smaller than complete logs as well as than causally complete logs. Examples that we have analyzed show that the original network can be discovered from an incomplete log $L$ for which the following holds: $\rightarrow^{B}_{N} \subset (\rightarrow_{L} \cup \Rightarrow_{L})$, and $\rightarrow_{L} \subset \rightarrow^{B}_{N}$, such a log we call a weakly complete log.

**Definition 5.** (Weakly complete log) Let $N = (P, T, F)$ be a sound WF-net, i.e., $N \in \mathcal{W}$, and let $\rightarrow^{B}_{N}$ be the basic causality relation of $N$. $L_w$ is a *weakly complete* workflow log of $N$ iff:

1) $\rightarrow^{B}_{N} \subset (\rightarrow_{Lw} \cup \Rightarrow_{Lw})$, and $\rightarrow_{Lw} \subset \rightarrow^{B}_{N}$, and
2) for any $t \in T$ there is $\sigma \in L_w$ so that $t \in \sigma$.

A network obtained from a weakly complete log often contains dangling nodes, i.e., activities without predecessors and/or successors. The definition of a WF-net established by Aalst van der, Weijters, and Maruster (2002: 7), includes an assumption of network connectivity, which means connectivity of all nodes in the network, and which prohibits the existence of dangling nodes. In attempt to overcome the problem of dangling nodes, we observed the relations in footprints, and based on these we have defined the rules of inference of direct from indirect successors and predecessors. Thus for each activity that is a dangling node, a successor and/or predecessor can be found.

Direct successors obtained this way constitute the elements of the causality relation that we call *inferred causality relations*, $\rightarrow^{I}_{Lw}$. The causality relation to be inserted into the final footprints (denoted with $\rightarrow_{Lf}$), and over which the $\alpha^{\|}$-algorithm is applied becomes: $\rightarrow_{Lf} = \rightarrow_{Lw} \cup \rightarrow^{I}_{Lw}$, which gives $\rightarrow_{Lf} = \rightarrow^{B}_{N}$. The use of these rules in order to find a causality relation equal to the basic causality relation, and thus to discover the original network, is shown in the following example.

Let us consider a parallel process model shown in Figure 1, and a log $L_2$ with records obtained after several executions of the process.

$L_2 = [< a, b, c, d, e, f, g, h >^3, < a, f, g, c, e, d, b, h >^2]$

The basic causality relation for this example is:

$\rightarrow^{B}_{N} = \{(a, b), (a, c), (a, f), (b, h), (c, d), (c, e), (d, h), (e, h), (f, g), (g, h)\}$

The footprint of the event log $L_2$ is given in Table 2.

From the footprints shown in Table 2, it can be seen that the causality relation of $L_2$ is:

$\rightarrow_{L2} = \{(a, b), (a, f), (b, h), (c, d), (c, e), (f, g), (g, h)\}$

It can be noted that the causality relation of log $L_2$ is not equal to the basic causality relation, i.e., $\rightarrow_{L2} \neq \rightarrow^{B}_{N}$, but also it holds: $\rightarrow^{B}_{N} \subset (\rightarrow_{L2} \cup \Rightarrow_{L2})$, $\rightarrow_{L2} \subset \rightarrow^{B}_{N}$, which makes $L_2$ a weakly complete log. It can also be seen from the footprints that the activity $c$ does not have its direct predecessors (the row which relates to $c$ does not contain a $\leftarrow$), and the activities $d$ and $e$ do not have direct successors (the rows $d$ and $e$ do not have $\rightarrow$), which means that there will be dangling nodes in the network.

According to the rules of inference of direct from indirect successors and predecessors in networks with dangling nodes, we get:

$c \Leftarrow_{L2} a$, $b \leftarrow_{L2} a$, $c \|_{L2} b$ then $c \leftarrow^{I}_{L2} a$, or $c \Leftarrow_{L2} a$, $f \leftarrow_{L2} a$, $c \|_{L2} f$ then $c \leftarrow^{I}_{L2} a$ i.e., $a \rightarrow^{I}_{Lw} c$

Table 2: Footprint of the log $L_2$.

| | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ | $h$ |
|---|---|---|---|---|---|---|---|---|
| $a$ | # | $\rightarrow$ | $\Rightarrow$ | $\Rightarrow$ | $\Rightarrow$ | $\rightarrow$ | $\Rightarrow$ | $\Rightarrow$ |
| $b$ | $\leftarrow$ | # | $\|\|$ | $\|\|$ | $\|\|$ | $\|\|$ | $\|\|$ | $\rightarrow$ |
| $c$ | $\Leftarrow$ | $\|\|$ | # | $\rightarrow$ | $\rightarrow$ | $\|\|$ | $\|\|$ | $\Rightarrow$ |
| $d$ | $\Leftarrow$ | $\|\|$ | $\leftarrow$ | # | $\|\|$ | $\|\|$ | $\|\|$ | $\Rightarrow$ |
| $e$ | $\Leftarrow$ | $\|\|$ | $\leftarrow$ | $\|\|$ | # | $\|\|$ | $\|\|$ | $\Rightarrow$ |
| $f$ | $\leftarrow$ | $\|\|$ | $\|\|$ | $\|\|$ | $\|\|$ | # | $\rightarrow$ | $\Rightarrow$ |
| $g$ | $\Leftarrow$ | $\|\|$ | $\|\|$ | $\|\|$ | $\|\|$ | $\leftarrow$ | # | $\rightarrow$ |
| $h$ | $\Leftarrow$ | $\leftarrow$ | $\Leftarrow$ | $\Leftarrow$ | $\Leftarrow$ | $\Leftarrow$ | $\leftarrow$ | # |

$d \Rightarrow_{L2} h$, $b \rightarrow_{L2} h$, $d \|_{L2} b$ then $d \rightarrow^{I}_{L2} h$, or $d \Rightarrow_{L2} h$, $g \rightarrow_{L2} h$, $d \|_{L2} g$ then $d \rightarrow^{I}_{L2} h$

$e \Rightarrow_{L2} h$, $b \rightarrow_{L2} h$, $e \|_{L2} b$ then $e \rightarrow^{I}_{L2} h$, or $e \Rightarrow_{L2} h$, $g \rightarrow_{L2} h$, $e \|_{L2} g$ then $e \rightarrow^{I}_{L2} h$

Thus: $\rightarrow^{I}_{L2} = \{(a, c), (d, h), (e, h)\}$, i.e.:

$\rightarrow_{Lf} = \rightarrow_{L2} \cup \rightarrow^{I}_{L2} = \{(a, b), (a, c), (a, f), (b, h), (c, d), (c, e), (d, h), (e, h), (f, g), (g, h)\}$

It can be noted that now it holds $\rightarrow_{Lf} = \rightarrow^{B}_{N}$.

As it was shown in the previous example, using the $\alpha^{\|}$-algorithm over the log whose causality relation is equal to the basic causality relation ($\rightarrow_{Lf} = \rightarrow^{B}_{N}$), the obtained network will be the same as the original network.

From the preliminary results presented in this paper, it can be seen that our assumption that the model of a parallel process can be obtained from the logs that do not meet the requirement of completeness is valid, and we are working on its formal proof or a counterexample (in the latter case, we will work on identifying the conditions in which the property still holds and on its experimental evaluation).

# 4 CONCLUSIONS

In this paper we were faced with one of the biggest challenges in the research process mining, which is the problem of completeness of the logs in the discovering process model based on the example of the process of behavior recorded in the workflow logs. Solving problems of completeness logs in parallel processes, presented in this paper, has led to change in the technique discovering the process model as well as in the $\alpha$-algorithm, (as it was cited previously).

Although the $\alpha$-algorithm is basically simple, it has not been particularly practical because of many problems that it cannot overcome, as it was described by Aalst van der (2011: 129). Besides the basic $\alpha$-algorithm, the examples of variation $\alpha$-algorithm given by L. Wen et al (2007), heuristic mining, given by A.J.M.M. Weijters and J.T.S. Ribeiro (2010), a genetic process mining, given by A.K.A de Medeiros (2006), fuzzy mining, given by Guenther and Aalst van der (2007), process mining from a basis of regions, given by M. Sole and J. Carmona (2010), are known, also. Most of these algorithms are emerged in order to overcome problems encountered in using the basic $\alpha$-algorithm, but the problem of completeness of logs has not been overcome, and it still remains a challenge for future researchers.

As such, it became the subject of our observations and modifications as a part of a broader research on discovering business process models by examples. Preliminary results presented in this paper address concurrent processes without loops. The examples show that with our modification of the discovering technique, we are able to overcome the problem of completeness of logs in parallel processes that occurs in the basic $\alpha$-algorithm, and to improve the efficiency of obtaining the process model. Our future work will be focused on finding the theoretical or empirical confirmation of the obtained results. Also, the obtained results encourage us to continue to investigate the effects of introduced modifications to other types of processes.

# REFERENCES


Aalst van der, W.M.P., 2011. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag, Berlin.

Aalst van der, W.M.P., Stahl, C., 2011. Modeling Business Processes: A Petri Net Oriented Approach. *MIT press*, Cambridge, MA.

Aalst van der, W.M.P., Weijters, A.J.M.M., Maruster, L., 2002. Workflow Mining: Which Processes can be Rediscovered? *BETA Working Paper Series,* WP 74, Eindhoven Univ. of Technology, Eindhoven.

Aalst van der, W.M.P., Weijters, A.J.M.M., Maruster, L., 2004. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128-1142.

M. Sole and J. Carmona, "Process Mining from a Basis of Regions," in Applications and Theory of Petri Nets 2010, ser. Lecture Notes in Computer Science, J. Lilius and W. Penczek, Eds., vol. 6128. Springer-Verlag, Berlin, 2010, pp. 226–245.

Guenther, C. W., Aalst van der, W.M.P., 2007. Fuzzy Mining: Adaptive Process Simplication Based on Multi-perspective Metrics, *in: BPM 2007*, Vol. 4714 of LNCS, Springer, pp. 328–343.

A.K.A de Medeiros. "Genetic Process Mining". PhD Thesis, Eindhoven University of Technology, 2006.

Rozinat, A., Aalst van der, W.M.P., 2008. Conformance Checking of Processes Based on Monitoring Real Behavior. *Information Systems*, 33(1):64–95.

A.J.M.M. Weijters and J.T.S. Ribeiro. Flexible Heuristics Miner (FHM). BETA Working Paper Series, WP 334, Eindhoven University of Technology, Eindhoven, 2010.

Wen, L., Aalst van der, W.M.P., Wang, J., Sun, J., 2007. Mining Process Models with Non-Free-Choice Constructs, *Data Mining and Knowledge Discovery* 15 (2) 145–180.