Evaluating Conformance Measures in Process Mining using Conformance Propositions

Anja F. Syring[1], Niek Tax[2], and Wil M.P. van der Aalst[1,2,3]

[1] Process and Data Science (Informatik 9),
RWTH Aachen University, D-52056 Aachen, Germany
[2] Architecture of Information Systems,
Eindhoven University of Technology, Eindhoven, The Netherlands
[3] Fraunhofer Institute for Applied Information Technology FIT,
Sankt Augustin, Germany

Abstract. Process mining sheds new light on the relationship between process models and real-life processes. Process discovery can be used to learn process models from event logs. Conformance checking is concerned with quantifying the *quality* of a business process model in relation to event data that was logged during the execution of the business process. There exist different categories of conformance measures. *Recall*, also called fitness, is concerned with quantifying how much of the behavior that was observed in the event log fits the process model. *Precision* is concerned with quantifying how much behavior a process model allows for that was never observed in the event log. *Generalization* is concerned with quantifying how well a process model generalizes to behavior that is possible in the business process but was never observed in the event log. Many recall, precision, and generalization measures have been developed throughout the years, but they are often defined in an ad-hoc manner without formally defining the desired properties up front. To address these problems, we formulate *21 conformance propositions* and we use these propositions to evaluate current and existing conformance measures. The goal is to trigger a discussion by clearly formulating the challenges and requirements (rather than proposing new measures). Additionally, this paper serves as an overview of the conformance checking measures that are available in the process mining area.

Keywords: Process mining · Conformance checking · Evaluation measures

1 Introduction

Process mining [2] is a fast growing discipline that focuses on the analysis of event data that is logged during the execution of a business process. Events in such an event log contain information on what was done, by whom, for whom, where, when, etc. Such event data are often readily available from information systems that support the execution of the business process, such as ERP, CRM, or BPM systems. *Process discovery*, the task of automatically generating a process model that accurately describes a business process based on such event data, plays a prominent role in process mining. Throughout the years, many process discovery algorithms have been developed, producing process models in various forms, such as Petri nets, process trees, and BPMN.

Event logs are often incomplete, i.e., they only contain a sample of all possible behavior in the business process. This not only makes process discovery challenging; it is also difficult to assess the quality of the process model in relation to the log. Process discovery algorithms take an event log as input and aim to output a process model that satisfies certain properties, which are often referred to as the four quality dimensions [2] of process mining: (1) *recall*: the discovered model should allow for the behavior seen in the event log (avoiding "non-fitting" behavior), (2) *precision*: the discovered model should not allow for behavior completely unrelated to what was seen in the event log (avoiding "underfitting"), (3) *generalization*: the discovered model should generalize the example behavior seen in the event log (avoiding "overfitting"), and (4) *simplicity*: the discovered model should not be unnecessarily complex. The simplicity dimension refers to Occam's Razor: "one should not increase, beyond what is necessary, the number of entities required to explain anything". In the context of process mining, this is often operationalized by quantifying the complexity of the model (number of nodes, number of arcs, understandability, etc.). We do *not* consider the simplicity dimension in this paper, since we focus on *behavior* and abstract from the actual model representation. Recall is often referred to as *fitness* in process mining literature. Sometimes fitness refers to a combination of the four quality dimensions. To avoid later confusion, we use the term recall which is commonly used in pattern recognition, information retrieval, and (binary) classification. Many conformance measures have been proposed throughout the years, e.g., [2, 4, 6, 12–15, 24, 25, 27, 32, 33].

So far it remains an open question whether existing measures for recall, precision, and generalization measure what they are aiming to measure. This motivates the need for a formal framework for conformance measures. Users of existing conformance measures should be aware of seemingly obvious quality issues of existing approaches and researchers and developers that aim to create new measures should be clear on what conformance characteristics they aim to support. To address this open question, this paper evaluates state-of-the-art conformance measures based on 21 propositions introduced in [3]. This paper supported by a detailed publicly available report detailing the evaluations of existing techniques [29].

The remainder is organized as follows. Section 2 discusses related work. Section 3 introduces basic concepts and notations. The rest of the paper is split into two parts where the first one discusses the topics of recall and precision (Section 4) and the second part is dedicated to generalization (Section 5). In both parts, we introduce the corresponding conformance propositions and provide an overview of existing conformance measures. Furthermore, we discuss our findings of validating existing these measures on the propositions. Additionally, Section 4 demonstrates the importance of the propositions on several baseline conformance measures, while Section 5 includes a discussion about the different points of view on generalization. Section 6 concludes the paper.

2 Related work

In early years, when process mining started to gain in popularity and the community around it grew, many process discovery algorithms were developed. But at that time there was no standard method to evaluate the results of these algorithms and to compare

them to the performance of other algorithms. Based on this, Rozinat et al. [28] called on the process mining community to develop a standard framework to evaluate process discovery algorithms. This led to a variety of fitness/recall, precision, generalization and simplicity notions [2]. These notions can be quantified in different ways and there are often trade-offs between the different quality dimensions. As shown using generic algorithms assigning weights to the different quality dimensions [10], one quickly gets degenerate models when leaving out one or two dimensions. For example, it is very easy to create a simple model with perfect recall (i.e., all observed behavior fits perfectly) that has poor precision and provides no insights.

Throughout the years, several conformance measures have been developed for each quality dimension. However, it is unclear whether these measures actually measure what they are supposed to. An initial step to address the need for a framework to evaluate conformance measures was made in [30]. Five so-called *axioms* for precision measures were defined that characterize the desired properties of such measures. Additionally, [30] showed that none of the existing precision measures satisfied all of the formulated axioms. In comparison to [30] Janssenswillen et al. [19] did not rely on qualitative criteria, but quantitatively compared existing recall, precision and generalization measures under the aspect of feasibility, validity and sensitivity. The results showed that all recall and precision measures tend to behave in a similar way, while generalization measures seemed to differ greatly from each other. In [3] van der Aalst made a follow-up step to [30] by formalizing recall and generalization in addition to precision and by extending the precision requirements, resulting in a list of 21 conformance propositions. Furthermore, [3] showed the importance of probabilistic conformance measures that also take into account trace probabilities in process models. Beyond that, [30] and [3] motivated the process mining community to develop new precision measures, taking the axioms and propositions as a design criterion, resulting in the measures among others the measures that are proposed in [26] and in [8]. Using the 21 propositions of [3] we evaluate state-of-the-art recall (e.g. [5, 26, 4, 16, 23, 27, 34]), precision (e.g. [4, 16, 17, 23, 13, 26, 27, 31]) and generalization (e.g. [4, 13, 16]) measures.

This paper uses the mainstream view that there are at least four quality dimensions: fitness/recall, precision, generalization, and simplicity [2]. We deliberately do not consider simplicity, since we focus on behavior only (i.e., not the model representation). Moreover, we treat generalization separately. In a controlled experiment one can assume the existence of a so-called "system model". This model can be simulated to create a synthetic event log used for discovery. In this setting, conformance checking can be reduced to measuring the similarity between the discovered model and the system model [9, 20]. In terms of the well-known confusion matrix, one can then reason about true positives, false positives, true negatives, and false negatives. However, without a system model and just an event log, it is not possible to find false positives (traces possible in the model but not in reality). Hence, precision cannot be determined in the traditional way. Janssenswillen and Depaire [18] conclude in their evaluation of state-of-the-art conformance measures that none of the existing approaches reliably measures this similarity. However, in this paper, we follow the traditional view on the quality dimensions and exclude the concept of the system from our work.

Whereas there are many fitness/recall and precision measures there are fewer generalization measures. Generalization deals with future cases that were not yet observed. There is no consensus on how to define generalization and in [19] it was shown that there is no agreement between existing generalization metrics. Therefore, we cover generalization in a separate section (Section 5). However, as discussed in [2] and demonstrated through experimentation [10], one cannot leave out the generalization dimension. The model that simply enumerates all the traces in the log has perfect fitness/recall and precision. However, event logs cannot be assumed to be complete, thus proving that a generalization dimension is needed.

3 Preliminaries

A *multiset* over a set X is a function $B : X \to \mathbb{N}$ which we write as $[a_1^{w_1}, a_2^{w_2}, \ldots, a_n^{w_n}]$ where for all $i \in [1, n]$ we have $a_i \in X$ and $w_i \in \mathbb{N}^*$. $\mathbb{B}(X)$ denotes the set of all multisets over set X. For example, $[a^3, b, c^2]$ is a multiset over set $X = \{a, b, c\}$ that contains three a elements, one b element and two c elements. $|B|$ is the number of elements in multiset B and $B(x)$ denotes the number of x elements in B. $B_1 \uplus B_2$ is the sum of two multisets: $(B_1 \uplus B_2)(x) = B_1(x) + B_2(x)$. $B_1 \setminus B_2$ is the difference containing all elements from B_1 that do not occur in B_2. Thus, $(B_1 \setminus B_2)(x) = max\ \{B_1(x) - B_2(x), 0\}$. $B_1 \cap B_2$ is the intersection of two multisets. Hence, $(B_1 \cap B_2)(x) = min\ \{B_1(x), B_2(x)\}$. $[x \in B \mid b(x)]$ is the multiset of all elements in B that satisfy some condition b. $B_1 \subseteq B_2$ denotes that B_1 is contained in B_2, e.g., $[a^2, b] \subseteq [a^2, b^2, c]$, but $[a^2, b^3] \not\subseteq [a^2, b^2, c^2]$ and $[a^2, b^2, c] \not\subseteq [a^3, b^3]$.

Process mining techniques focus on the relationship between observed behavior and modeled behavior. Therefore, we first formalize event logs (i.e., observed behavior) and process models (i.e., modeled behavior). To do this, we consider a very simple setting where we only focus on the control-flow, i.e., sequences of activities.

3.1 Event Logs

The starting point for process mining is an event log. Each *event* in such a log refers to an *activity* possibly executed by a *resource* at a particular *time* and for a particular *case*. An event may have many more attributes, e.g., transactional information, costs, customer, location, and unit. Here, we focus on control-flow. Therefore, we only consider activity labels and the ordering of events within cases.

Definition 1 (Traces). *$\mathcal{A}$ is the universe of* activities. *A* trace *$t \in \mathcal{A}^*$ is a sequence of activities. $\mathcal{T} = \mathcal{A}^*$ is the universe of traces.*

Trace $t = \langle a, b, c, d, a \rangle$ refers to 5 events belonging to the same case (i.e., $|t| = 5$). An event log is a collection of cases each represented by a trace.

Definition 2 (Event Log). *$\mathcal{L} = \mathbb{B}(\mathcal{T})$ is the universe of event logs. An* event log *$l \in \mathcal{L}$ is a finite multiset of observed traces. $\tau(l) = \{t \in l\} \subseteq \mathcal{T}$ is the set of traces appearing in $l \in \mathcal{L}$. $\overline{\tau}(l) = \mathcal{T} \setminus \tau(l)$ is the complement of the set of non-observed traces.*

Event log $l = [\langle a, b, c \rangle^5, \langle b, a, d \rangle^3, \langle a, b, d \rangle^2]$ refers to 10 cases (i.e., $|l| = 10$). Five cases are represented by the trace $\langle a, b, c \rangle$, three cases are represented by the trace $\langle b, a, d \rangle$, and two cases are represented by the trace $\langle a, b, d \rangle$. Hence, $l(\langle a, b, d \rangle) = 2$.

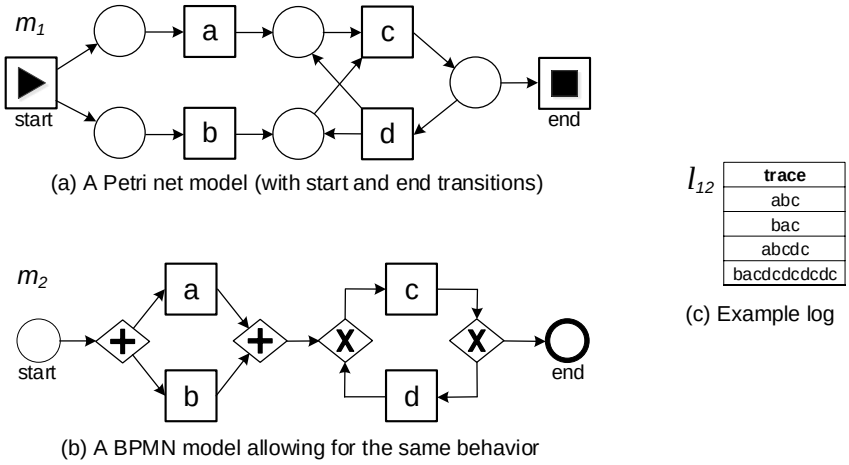

Fig. 1: Two process models m_1 and m_2 allowing for the same set of traces ($\tau(m_1) = \tau(m_2)$) with an example log l_{12} (c).

3.2 Process Models

The behavior of a process model m is simply the set of traces allowed by m. In our definition, we will abstract from the actual representation (e.g. Petri nets or BPMN).

Definition 3 (Process Model). *$\mathcal{M}$ is the set of process models. A process model $m \in \mathcal{M}$ allows for a set of traces $\tau(m) \subseteq \mathcal{T}$. $\overline{\tau}(m) = \mathcal{T} \setminus \tau(m)$ is the complement of the set of traces allowed by model $m \in \mathcal{M}$.*

A process model $m \in \mathcal{M}$ may abstract from the real process and leave out unlikely behavior. Furthermore, this abstraction can result in $\tau(m)$ allowing for traces that cannot happen (e.g., particular interleavings or loops).

We distinguish between *representation* and *behavior* of a model. Process model $m \in \mathcal{M}$ can be represented using a plethora of modeling languages, e.g., Petri nets, BPMN models, UML activity diagrams, automata, and process trees. Here, we abstract from the actual representation and focus on behavioral characteristics $\tau(m) \subseteq \mathcal{T}$.

Figure 1 (a) and (b) show two process models that have the same behavior: $\tau(m_1) = \tau(m_2) = \{\langle a, b, c\rangle, \langle a, c, b\rangle, \langle a, b, c, d, c\rangle, \langle b, a, c, d, c\rangle, \dots\}$. Figure 1(c) shows a possible event log generated by one of these models $l_{12} = [\langle a, b, c\rangle^3, \langle b, a, c\rangle^5, \langle a, b, c, d, c\rangle^2, \langle b, a, c, d, c, d, c, d, c\rangle^2]$.

The behavior $\tau(m)$ of a process model $m \in \mathcal{M}$ can be of infinite size. We use Figure 1 to illustrate this. There is a "race" between a and b. After a and b, activity c will occur. Then there is a probability that the process ends or d can occur. Let $t_{a,k} = \langle a, b\rangle \cdot (\langle c, d\rangle)^k \cdot \langle c\rangle$ be the trace that starts with a and where d is executed k times. $t_{b,k} = \langle b, a\rangle \cdot (\langle c, d\rangle)^k \cdot \langle c\rangle$ is the trace that starts with b and where d is executed k times. $\tau(m_1) = \tau(m_2) = \bigcup_{k \geq 0}\{t_{a,k}, t_{b,k}\}$. Some examples are given in Figure 1(c).

Since any log contains only a finite number of traces, one can never observe all traces possible in m_1 or m_2.

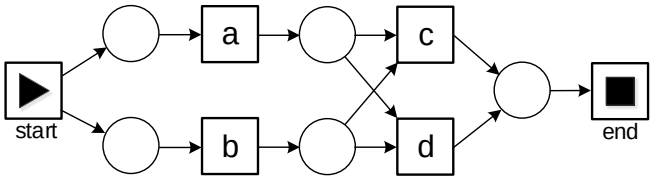

Fig. 2: A process model m_3 discovered based on log $l_3 = [\langle a,b,c\rangle^5, \langle b,a,d\rangle^3, \langle a,b,d\rangle^2]$.

3.3 Process Discovery

A discovery algorithm takes an event log as input and returns a process model. For example, the model m_3 in Figure 2 could have been discovered based on event log $l_3 = [\langle a,b,c\rangle^5, \langle b,a,d\rangle^3, \langle a,b,d\rangle^2]$. Ideally, the process model should capture the (dominant) behavior observed but it should also generalize without becoming too imprecise. For example, the model allows for trace $t = \langle b,a,c\rangle$ although this was never observed.

Definition 4 (Discovery Algorithm). *A discovery algorithm can be described as a function $disc \in \mathcal{L} \rightarrow \mathcal{M}$ mapping event logs onto process models.*

We abstract from concrete discovery algorithms. Over 100 discovery algorithms have been proposed in literature [2]. Merely as a reference to explain basic notions, we define three simple, but extreme, algorithms: $disc_{ofit}$, $disc_{ufit}$, and $disc_{nfit}$. Let $l \in \mathcal{L}$ be a log. $disc_{ofit}(l) = m_o$ such that $\tau(m_o) = \tau(l)$ produces an overfitting model that allows only for the behavior seen in the log. $disc_{ufit}(l) = m_u$ such that $\tau(m_u) = \mathcal{T}$ produces an underfitting model that allows for any behavior. $disc_{nfit}(l) = m_n$ such that $\tau(m_n) = \overline{\tau}(l)$ produces a non-fitting model that allows for all behavior *not* seen in the log.

4 Recall and Precision

Many recall measures have been proposed in literature [2, 4, 6, 12–15, 24, 25, 27, 32, 33]. In recent years, also several precision measures have been proposed [7, 30]. Only few generalization measures have been proposed [4]. The goal of this paper is to evaluate these quality measures. To achieve this, in the following the propositions introduced in [3] are applied to existing conformance measures.

The notion of recall and precision are well established in the process mining community. Definitions are in place and there is an agreement on what these two measures are supposed to measure. However, this is not the case for generalization. There exist different points of view on what generalization is supposed to measure. Depending on these, existing generalization measures might greatly differ from each other.

To account for the different levels of maturity in recall, precision and generalization and to address the controversy in the generalization area, the following section will solely handle recall and precision while Section 5 focuses on generalization. Both

sections establish baseline measures, introduce the corresponding propositions of [3], present existing conformance measures and evaluate them using the propositions.

4.1 Baseline Recall and Precision measures

We assume the existence of two functions: $rec()$ and $prec()$ respectively denoting recall and precision. Both take a log and model as input and return a value between 0 and 1. The higher the value, the better.

Definition 5 (Recall). *A* recall measure $rec \in \mathcal{L} \times \mathcal{M} \rightarrow [0, 1]$ *aims to quantify the fraction of observed behavior that is allowed by the model.*

Definition 6 (Precision). *A* precision measure $prec \in \mathcal{L} \times \mathcal{M} \rightarrow [0, 1]$ *aims to quantify the fraction of behavior allowed by the model that was actually observed.*

If we ignore frequencies of traces, we can simply count fractions of traces yielding the following two simple measures.

Definition 7 (Trace-Based L2M Precision and Recall). *Let $l \in \mathcal{L}$ and $m \in \mathcal{M}$ be an event log and a process model. Trace-based L2M precision and recall are defined as follows:*

$$rec_{TB}(l, m) = \frac{|\tau(l) \cap \tau(m)|}{|\tau(l)|} \quad prec_{TB}(l, m) = \frac{|\tau(l) \cap \tau(m)|}{|\tau(m)|} \tag{1}$$

Since $|\tau(l)|$ is bounded by the size of the log, $rec_{TB}(l, m)$ is well-defined. However, $prec_{TB}(l, m)$ is undefined when $\tau(m)$ is unbounded (e.g., in case of loops).

One can argue, that the frequency of traces should be taken into account when evaluating conformance which yields the following measure. Note that it is not possible to define frequency-based precision based on a process model that does not define the probability of its traces. Since probabilities are specifically excluded from the scope of this paper, the following approach only defines frequency-based recall.

Definition 8 (Frequency-Based L2M Recall). *Let $l \in \mathcal{L}$ and $m \in \mathcal{M}$ be an event log and a process model. Frequency-based L2M recall is defined as follows:*

$$rec_{FB}(l, m) = \frac{|[t \in l \mid t \in \tau(m)]|}{|l|} \tag{2}$$

4.2 A Collection of Conformance Propositions

In [3], 21 *conformance propositions* covering the different conformance dimensions (except simplicity) were given. In this section, we focus on the general, recall and precision propositions introduced in [3]. We discuss the generalization propositions separately, because they reason about unseen cases not yet recorded in the event log. Most of the conformance propositions have broad support from the community, i.e., there is broad consensus that these propositions should hold. These are marked with a "+". More controversial propositions are marked with a "0" (rather than a "+").

General Propositions The first two propositions are commonly accepted; the computation of a quality measure should be deterministic (**DetPro**$^+$) and only depend on behavioral aspects (**BehPro**$^+$). The latter is a design choice. We deliberately exclude simplicity notions.

Proposition 1 (DetPro$^+$). *$rec()$, $prec()$, $gen()$ are deterministic functions, i.e., the measures $rec(l, m)$, $prec(l, m)$, $gen(l, m)$ are fully determined by $l \in \mathcal{L}$ and $m \in \mathcal{M}$.*

Proposition 2 (BehPro$^+$). *For any $l \in \mathcal{L}$ and $m_1, m_2 \in \mathcal{M}$ such that $\tau(m_1) = \tau(m_2)$: $rec(l, m_1) = rec(l, m_2)$, $prec(l, m_1) = prec(l, m_2)$, and $gen(l, m_1) = gen(l, m_2)$, i.e., the measures are fully determined by the behavior observed and the behavior described by the model (representation does not matter).*

Recall Propositions In this subsection, we consider a few *recall propositions*. $rec \in \mathcal{L} \times \mathcal{M} \to [0, 1]$ aims to quantify the fraction of observed behavior that is allowed by the model. Proposition **RecPro1**$^+$ states that extending the model to allow for more behavior can never result in a lower recall. From the definition follows, that this proposition implies **BehPro**$^+$. Recall measures violating **BehPro**$^+$ also violate **RecPro1**$^+$ which is demonstrated as follows:

For two models m_1, m_2 with $\tau(m_1) = \tau(m_2)$ it follows from **RecPro1**$^+$ that $rec(l, m_1) \leq rec(l, m_2)$ because $\tau(m_1) \subseteq \tau(m_2)$. From **RecPro1**$^+$ follows that $rec(l, m_2) \leq rec(l, m_1)$ because $\tau(m_2) \subseteq \tau(m_1)$. Combined, $rec(l, m_2) \leq rec(l, m_1)$ and $rec(l, m_1) \leq rec(l, m_2)$ gives $rec(l, m_2) = rec(l, m_1)$, thus, recall measures that fulfill **RecPro1**$^+$ are fully determined by the behavior observed and the behavior described by the model, i.e., representation does not matter.

Proposition 3 (RecPro1$^+$). *For any $l \in \mathcal{L}$ and $m_1, m_2 \in \mathcal{M}$ such that $\tau(m_1) \subseteq \tau(m_2)$: $rec(l, m_1) \leq rec(l, m_2)$.*

Similarly to **RecPro1**$^+$, it cannot be the case that adding fitting behavior to the event logs, lowers recall (**RecPro2**$^+$).

Proposition 4 (RecPro2$^+$). *For any $l_1, l_2, l_3 \in \mathcal{L}$ and $m \in \mathcal{M}$ such that $l_2 = l_1 \uplus l_3$ and $\tau(l_3) \subseteq \tau(m)$: $rec(l_1, m) \leq rec(l_2, m)$.*

Similarly to **RecPro2**$^+$, one can argue that adding non-fitting behavior to event logs should not be able to increase recall (**RecPro3**^{0}). However, one could also argue that recall should not be measured on a trace-level, but should instead distinguish between non-fitting traces by measuring the *degree* in which a non-fitting trace is still fitting. Therefore, unlike the previous propositions, this requirement is debatable as is indicated by the "0" tag.

Proposition 5 (RecPro3^0). *For any $l_1, l_2, l_3 \in \mathcal{L}$ and $m \in \mathcal{M}$ such that $l_2 = l_1 \uplus l_3$ and $\tau(l_3) \subseteq \overline{\tau}(m)$: $rec(l_1, m) \geq rec(l_2, m)$.*

For any $k \in \mathbb{N}$: $l^k(t) = k \cdot l(t)$, e.g., if $l = [\langle a, b\rangle^3, \langle c\rangle^2]$, then $l^4 = [\langle a, b\rangle^{12}, \langle c\rangle^8]$. We use this notation to enlarge event logs without changing the original distribution. One could argue that this should not influence recall (**RecPro4**^{0}), e.g., $rec([\langle a, b\rangle^3,$

$\langle c\rangle^2], m) = rec([\langle a, b\rangle^{12}, \langle c\rangle^8], m)$. On the other hand, larger logs can provide more confidence that the log is indeed a representative sample of the possible behavior. Therefore, it is debatable whether the size of the event log should have influence on recall as indicated by the "0" tag.

Proposition 6 (RecPro4^0). *For any $l \in \mathcal{L}$, $m \in \mathcal{M}$, and $k \geq 1$: $rec(l^k, m) = rec(l, m)$.*

Finally, we provide a proposition stating that recall should be 1 if all traces in the log fit the model (**RecPro5**$^+$). As a result, the empty log has recall 1 for any model. Based on this proposition, $rec(l, disc_{ofit}(l)) = rec(l, disc_{ufit}(l)) = 1$ for any log l.

Proposition 7 (RecPro5$^+$). *For any $l \in \mathcal{L}$ and $m \in \mathcal{M}$ such that $\tau(l) \subseteq \tau(m)$: $rec(l, m) = 1$.*

Precision Propositions Precision ($prec \in \mathcal{L} \times \mathcal{M} \to [0, 1]$) aims to quantify the fraction of behavior allowed by the model that was actually observed. Initial work in the area of checking requirements of conformance checking measures started with [30], where five axioms for precision measures were introduced. The precision propositions that we state below partly overlap with these axioms, but some have been added and some have been strengthened. Axiom 1 of [30] specifies **DetPro**$^+$ for the case of precision, while we have generalized it to the recall and generalization dimension. Furthermore, **BehPro**$^+$ generalizes axiom 4 of [30] from its initial focus on precision to also cover recall and generalization. **PrecPro1**$^+$ states that removing behavior from a model that does not happen in the event log cannot lead to a lower precision. From the definition follows, that this proposition implies **BehPro**$^+$. Precision measures violating **BehPro**$^+$ also violate **PrecPro1**$^+$. Adding fitting traces to the event log can also not lower precision (**PrecPro2**$^+$). However, adding non-fitting traces to the event log should not change precision (**PrecPro3**^{0}).

Proposition 8 (PrecPro1$^+$). *For any $l \in \mathcal{L}$ and $m_1, m_2 \in \mathcal{M}$ such that $\tau(m_1) \subseteq \tau(m_2)$ and $\tau(l) \cap (\tau(m_2) \setminus \tau(m_1)) = \emptyset$: $prec(l, m_1) \geq prec(l, m_2)$.*

This proposition captures the same idea as axiom 2 in [30], but it is more general. Axiom 2 only put this requirement on precision when $\tau(l) \subseteq \tau(m_1)$, while **PrecPro1**$^+$ also concerns the situation where this does not hold.

Proposition 9 (PrecPro2$^+$). *For any $l_1, l_2, l_3 \in \mathcal{L}$ and $m \in \mathcal{M}$ such that $l_2 = l_1 \uplus l_3$ and $\tau(l_3) \subseteq \tau(m)$: $prec(l_1, m) \leq prec(l_2, m)$.*

This proposition is identical to axiom 5 in [30].

Proposition 10 (PrecPro3^0). *For any $l_1, l_2, l_3 \in \mathcal{L}$ and $m \in \mathcal{M}$ such that $l_2 = l_1 \uplus l_3$ and $\tau(l_3) \subseteq \overline{\tau}(m)$: $prec(l_1, m) = prec(l_2, m)$.*

One could also argue that duplicating the event log should not influence precision because the distribution remains the same (**PrecPro4**^{0}), e.g., $prec([\langle a, b\rangle^{20}, \langle c\rangle^{20}], m) = prec([\langle a, b\rangle^{40}, \langle c\rangle^{40}], m)$. Similar to (**RecPro3**^{0}) and (**RecPro4**^{0}), the equivalents on the precision side are tagged with "0".

Proposition 11 (PrecPro4^0). *For any $l \in \mathcal{L}$, $m \in \mathcal{M}$, and $k \geq 1$: $prec(l^k, m) = prec(l, m)$.*

If the model allows for the behavior observed and nothing more, precision should be maximal (**PrecPro5**$^+$). One could also argue that if all modeled behavior was observed, precision should also be 1 (**PrecPro6**^{0}). The latter proposition is debatable because it implies that the non-fitting behavior cannot influence perfect precision, as indicated by the "0" tag. Consider for example extreme cases where the model covers just a small fraction of all observed behavior (or even more extreme situations like $\tau(m) = \emptyset$). According to **PrecPro5**$^+$and **PrecPro6**^{0}, $rec(l, disc_{ofit}(l)) = 1$ for any log l.

Proposition 12 (PrecPro5$^+$). *For any $l \in \mathcal{L}$ and $m \in \mathcal{M}$ such that $\tau(m) = \tau(l)$: $prec(l, m) = 1$.*

Proposition 13 (PrecPro6^0). *For any $l \in \mathcal{L}$ and $m \in \mathcal{M}$ such that $\tau(m) \subseteq \tau(l)$: $prec(l, m) = 1$.*

4.3 Evaluation of Baseline Conformance Measures

To illustrate the presented propositions and justify their formulation, we evaluate the conformance measures defined as baselines in Section 4.1. Note that these 3 baseline measures were introduced to provide simple examples that can be used to discuss the propositions. We conduct this evaluation under the assumption that $l \neq [\,]$, $\tau(m) \neq \emptyset$ and $\langle\rangle \notin \tau(m)$.

General Propositions. Based on the definition of rec_{TB} and rec_{FB} it is clear that all measures can be fully determined by the log and the model. Consequently, **DetPro**$^+$ hold for these two baseline conformance measures. However, $prec_{TB}$ is undefined when $\tau(m)$ is unbound and, therefore, non-deterministic.

The behavior of the model is defined as sets of traces $\tau(m)$, which abstracts from the representation of the process model itself. Therefore, all recall and precision baseline conformance measures fulfill **BehPro**$^+$.

Recall Propositions. Considering measure rec_{TB}, it is obvious that **RecPro1**$^+$ holds if $\tau(m_1) \subseteq \tau(m_2)$, because the intersection between $\tau(m_2)$ and $\tau(l)$ will always be equal or bigger to the intersection of $\tau(m_1)$ and $\tau(l)$. The **RecPro2**$^+$ proposition holds for rec_{TB}, if $l_2 = l_1 \uplus l_3$ and $\tau(l_3) \subseteq \tau(m)$, because the additional fitting behavior is added to the nominator as well as the denominator of the formula: $|\tau(l_1) \cap \tau(m)| + |\tau(l_3)|)/(|\tau(l_1)| + |\tau(l_3)|$. This can never decrease recall. Furthermore, **RecPro3**^{0} propositions holds for rec_{TB} since adding unfitting behavior cannot increase the intersection between traces of the model and the log if $l_2 = l_1 \uplus l_3$ and $\tau(l_3) \subseteq \overline{\tau}(m)$. Consequently, only the denominator of the formula grows, which decreases recall. Similarly, we can show that these two proposition hold for rec_{FB}.

Duplication of the event log cannot affect rec_{TB}, since it is defined based on the set of traces and not the multiset. The proposition also holds for rec_{FB} since nominator and denominator of the formula will grow in proportion. Hence, **RecPro4**^{0} holds for both

baseline measures. Considering rec_{TB}, **RecPro5**$^{+}$ holds, since $\tau(l) \cap \tau(m) = \tau(l)$ if $\tau(l) \subseteq \tau(m)$ and consequently $|\tau(l) \cap \tau(m)| / |\tau(l)| = |\tau(l)| / |\tau(l)| = 1$. The same conclusions can be drawn for rec_{FB}.

Precision Propositions. Consider proposition **PrecPro1**$^{+}$ together with $prec_{TB}$. The proposition holds, since removing behavior from the model that does not happen in the event log will not affect the intersection between the traces of the model and the log: $\tau(l) \cap \tau(m_2) = \tau(l) \cap \tau(m_1)$ if $\tau(m_1) \subseteq \tau(m_2)$ and $\tau(l) \cap (\tau(m_2) \setminus \tau(m_1)) = \emptyset$. At the same time the denominator of the formula decreases, which can never decrease precision itself. **PrecPro2**$^{+}$ also holds for $prec_{TB}$, since the fitting behavior increases the intersection between traces of the model and the log, while the denominator of the formula stays the same. Furthermore, **PrecPro3**^{0} holds for $prec_{TB}$, since unfitting behavior cannot affect the intersection between traces of the model and the log.

Duplication of the event log cannot affect $prec_{TB}$, since it is defined based on the set of traces and not the multiset, i.e. **PrecPro4**^{0} holds.

Considering $prec_{TB}$, **PrecPro5**$^{+}$ holds, since $\tau(l) \cap \tau(m) = \tau(m)$ if $\tau(m) = \tau(l)$ and consequently $|\tau(l) \cap \tau(m)| / |\tau(m)| = |\tau(m)| / |\tau(m)| = 1$. Similarly, **PrecPro6**^{0} holds for $prec_{TB}$.

4.4 Existing Recall Measures

The previous evaluation of the simple baseline measures shows that the recall measures fulfill all propositions and the baseline precision measure only violates one proposition. However, the work presented in [30] demonstrated for precision, that most of the existing approaches violate seemingly obvious requirements. This is surprising compared to the results of our simple baseline measure. Inspired by [30], this paper takes a broad look at existing conformance measures with respect to the previously presented propositions. In the following section, existing recall and precision measures are introduced, before they will be evaluated in Section 4.6.

Causal footprint recall (rec_A). Van der Aalst et al. [5] introduce the concept of the footprint matrix, which captures the relations between the different activities in the log. The technique relies on the principle that if activity a is followed by b but b is never followed by a, then there is a causal dependency between a and b. The log can be described using four different relations types. In [2] it is stated that a footprint matrix can also be derived for a process model by generating a complete event log from it. Recall can be measured by counting the mismatches between both matrices. Note that this approach assumes an event log which is complete with respect to the directly follows relations.

Token replay recall (rec_B). Token replay measures recall by replaying the log on the model and counting mismatches in the form of missing and remaining tokens. This approach was proposed by Rozinat and van der Aalst [27]. During replay, four types of tokens are distinguished: p the number of *produced* tokens, c the number of *consumed* tokens, m the number of *missing* tokens that had to be added because a transition was

not enabled during replay and r the number of *remaining* tokens that are left in the model after replay. In the beginning, a token is produced in the initial place. Similarly, the approach ends by consuming a token from the final place. The more missing and remaining tokens are counted during replay the lower recall: $rec_B = \frac{1}{2}(1 - \frac{m}{c}) + \frac{1}{2}(1 - \frac{r}{p})$ Note that the approach assumes a relaxed sound workflow net, but it allows for duplicate and silent transitions.

Alignment recall (rec_C). Another approach to determine recall was proposed by van der Aalst et al. [4]. It calculates recall based on alignments, which detect process deviations by mapping the steps taken in the event log to the ones of the process model. This map can contain three types of steps (so-called moves): *synchronous* moves when event log and model agree, *log* moves if the event was recorded in the event log but should not have happened according to the model and *model* moves if the event should have happened according to the model but did not in the event log. The approach uses a function that assigns costs to log moves and model moves. This function is used to compute the optimal alignment for each trace in the log (i.e. the alignment with the least cost associated).

To compute recall, the total alignment cost of the log is normalized with respect to the cost of the worst-case scenario where there are only moves in the log and in the model but never together. Note, that the approach assumes an accepting Petri net with an initial and final state. However, it allows for duplicate and silent transitions in the process model.

Behavioral recall (rec_D). Goedertier et al. [16] define recall according to its definition in the data mining field using true positive (TP) and false negative (FN) counters. $TP(l, m)$ denotes the number of true positives, i.e., the number of events in the log that can be parsed correctly in model m by firing a corresponding enabled transition. $FN(l, m)$ denotes the number of false negatives, i.e., the number of events in the log for which the corresponding transition that was needed to mimic the event was not enabled and needed to be force-fired. The recall measure is defined as follows: $rec_D(l, m) = \frac{TP(l,m)}{TP(l,m)+FN(l,m)}$.

Projected recall (rec_E). Leemans et al. [23] developed a conformance checking approach that is also able to handle big event logs. This is achieved by projecting the event log as well as the model on all possible subsets of activities of size k. The behavior of a projected log and projected model is translated into the minimal deterministic finite automata (DFA)[4]. Recall is calculated by checking the fraction of the behavior that is allowed for by the minimal log-automaton that is also allowed for by the minimal model-automaton for each projection and by averaging the recall over each projection.

Continued parsing measure (rec_F). This continued parsing measure was developed in the context of the heuristic miner by Weijters et al. [34]. It abstracts from the representation of the process model by translating the Petri net into a causal matrix. This

[4] Every regular language has a unique minimal DFA according to the Myhill–Nerode theorem.

matrix defines input and output expressions for each activity, which describe possible in- and output behavior. When replaying the event log on the causal matrix, one has to check whether the corresponding input and output expressions are activated and therefore enable the execution of the activity. To calculate the continued parsing measure the number of events e in the event log, as well as the number of missing activated input expressions m and remaining activated output expressions r are counted. Note, that the approach allows for silent transitions in the process model but excludes duplicate transitions.

Eigenvalue recall (rec_G). Polyvyanyy et al. [26] introduce a framework for the definition of language quotients that guarantee several properties similar to the propositions introduced in [3]. To illustrate this framework, they apply it in the process mining context and define a recall measure. Hereby they rely on the relation between the language of a deterministic finite automaton (DFA) that describes the behavior of the model and the language of the log. In principle, recall is defined as in Definition 7. However, the measure is undefined if the language of the model or the log are infinite. Therefore, instead of using the cardinality of the languages and their intersection, the measure computes their corresponding eigenvalues and sets them in relation. To compute these eigenvalues, the languages have to be irreducible. Since this is not the case for the language of event logs, Polyvyanyy et al. [26] introduce a short-circuit measure over languages and proved that it is a deterministic measure over any arbitrary regular language.

4.5 Existing Precision Measures

Soundness ($prec_H$). The notion of soundness as defined by Greco et al. [17] states that a model is precise if all possible enactments of the process have been observed in the event log. Therefore, it divides the number of unique traces in the log compliant with the process model by the number of unique traces through the model. Note, that this approach assumes the process model in the shape of a workflow net. Furthermore, it is equivalent to the baseline precision measure $prec_T B$.

Simple behavioral appropriateness ($prec_I$). Rozinat and van der Aalst [27] introduce simple behavioral appropriateness to measure the precision of process models. The approach assumes that imprecise models enable a lot of transitions during replay. Therefore, the approach computes the mean number of enabled transitions x_i for each unique trace i and puts it in relation to the visible transitions T_V in the process model. Note, that the approach assumes a sound workflow net. However, it allows for duplicate and silent transitions in the process model.

Advanced behavioral appropriateness ($prec_J$). In the same paper, Rozinat and van der Aalst [27] define advanced behavioral appropriateness. This approach abstracts from the process model by describing the relation between activities of both the log and model with respect to whether these activities *follow* and/or *precede* each other. Hereby

they differentiate between *never*, *sometimes* and *always* precede/follow relations. To calculate precision the set of sometimes followed relations of the log S_F^l and the model S_F^m are considered, as well as their sometimes precedes relations S_P^l and S_P^m. The fraction of sometimes follows/precedes relations of the model which are also observed by the event log defines precision. Note, that the approach assumes a sound workflow net. However, it allows for duplicate and silent transitions in the process model.

ETC-one/ETC-rep ($prec_K$) and ETC-all ($prec_L$). Munoz-Gama and Carmona [25] introduced a precision measure which constructs an automaton that reflects the states of the model which are visited by the event log. For each state, it is evaluated whether there are activities which were allowed by the process model but not observed by the event log. These activities are added to the automaton as so-called escaping edges. Since this approach is not able to handle unfitting behavior, [7] and [4] extended the approach with a preprocessing step that aligned the log to the model before the construction of the automaton. Since it is possible that traces result in multiple optimal alignments, there are three variations of the precision measure. One can randomly pick one alignment and construct the alignment automaton based on it (ETC-one), select a representative set of multiple alignments (ETC-rep) or use all optimal alignments (ETC-all). For each variation, [4] defines an approach that assigns appropriate weights to the edges of the automaton. Precision is then computed by comparing for each state of the automaton, the weighted number of non-escaping edges to the total number of edges.

Behavioral specificity ($prec_M$) and Behavioral precision ($prec_N$). Goedertier et al. [16] introduced a precision measure based on the concept of negative events that is defined based on the concept of a confusion matrix as used in the data mining field. In this confusion matrix, the induced negative events are considered to be the ground truth and the process model is considered to be a prediction machine that predicts whether an event can or cannot occur. A negative event expresses that at a certain position in a trace, a particular event cannot occur. To induce the negative events into an event log, the traces are split in subsequences of length k. For each event e in the trace, it is checked whether another event e_n could be a negative event. Therefore the approach searches whether the set of subsequences contains a similar sequence to the one preceding e. If no matching sequence is found that contains e_n at the current position of e, e_n is recorded as a negative event of e. To check conformance the log, that was induced with negative events, is replayed on the model.

For both measures, the log that was induced with negative events is replayed on the model. Specificity and precision are measured according to their data mining definition using true positive (TP), false positive (FP) and true negative (TN) counts.

Goedertier et al. [16] ($prec_M$) defined behavioral specificity precision as $prec_M(l, m) = \frac{TN(l,m)}{TN(l,m)+FP(l,m)}$, i.e., the ratio of the induced negative events that were also disallowed by m. More recently, De Weerdt et al. [33] gave an inverse definition, called behavioral precision ($prec_N$), as the ratio of behavior that is allowed by m that does not conflict an induced negative event, i.e. $prec_N(l, m) = \frac{TP(l,m)}{TP(l,m)+FP(l,m)}$.

Weighted negative event precision ($prec_O$). Van den Broucke et al. [31] proposed an improvement to the approach of Goedertier et al. [16], which assigns weights to negative events. These weights indicate the confidence of the negative events actually being negative. To compute the weight, the approach takes the sequence preceding event e and searches for the matching subsequences in the event log. All events that have never followed such a subsequence are identified as negative events for e and their weight is computed based on the length of the matching subsequence. To calculate precision the enhanced log is replayed on the model, similar to the approach introduced in [33]. However, instead of increasing the counters by 1 they are increased by the weight of the negative event. Furthermore, van den Broucke et al. [31] also introduced a modified trace replay procedure which finds the best fitting firing sequence of transitions, taking force firing of transitions as well as paths enabled by silent transitions into account.

Projected precision ($prec_P$). Along with projected recall (rec_E) Leemans et al. [23] introduce projected precision. To compute precision, the approach creates a DFA which describes the conjunction of the behavior of the model and the event log. The number of outgoing edges of $DFA(m|_A)$ and the conjunctive automaton $DFAc(l, m, A)$ are compared. Precision is calculated for each subset of size k and averaged over the number of subsets.

Anti-alignment precision ($prec_Q$). Van Dongen et al. [13] propose a conformance checking approach based on anti-alignments. An anti-alignment is a run of a model which differs from all the traces in a log. The principle of the approach assumes that a very precise model only allows for the observed traces and nothing more. If one trace is removed from the log, it becomes the anti-alignment for the remaining log.

Therefore, trace-based precision computes an anti-alignment for each trace in the log. Then the distance d between the anti-alignment and the trace σ is computed. This is summed up for each trace and averaged over the number of traces in the log. The more precise a model, the lower the distance. However, the anti-alignment used for trace-based precision is limited by the length of the removed trace $|\sigma|$. Therefore, log-based precision uses an anti-alignment between the model and the complete log which has a length which is much greater than the traces observed in the log. Anti-alignment precision is the weighted combination of trace-based and log-based anti-alignment precision. Note, that the approach allows for duplicate and silent transitions in the process model.

Eigenvalue precision ($prec_R$). Polyvyanyy et al. [26] also define a precision measure along with the Eigenvalue recall (rec_G). For precision, they rely on the relation between the language of a deterministic finite automaton (DFA) that describes the behavior of the model and the language of the log. To overcome the problems arising with infinite languages of the model or log, they compute their corresponding eigenvalues and set them in relation. To compute these eigenvalues, the languages have to be irreducible. Since this is not the case for the language of event logs, Polyvyanyy et al. [26] introduce a short-circuit measure over languages and proof that it is a deterministic measure over any arbitrary regular language.

Table 1: Overview of the recall propositions that hold for the existing measures (under the assumption that $l \neq [\,]$, $\tau(m) \neq \emptyset$ and $\langle\rangle \notin \tau(m)$): $\surd$ means that the proposition holds for any log and model and $\times$ means that the proposition does not always hold.

Proposition	Name	rec_A	rec_B	rec_C	rec_D	rec_E	rec_F	rec_G
1	**DetPro**$^+$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\times$	$\surd$
2	**BehPro**$^+$	$\surd$	$\times$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$
3	**RecPro1**$^+$	$\times$	$\times$	$\surd$	$\times$	$\surd$	$\surd$	$\surd$
4	**RecPro2**$^+$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
5	**RecPro3**^{0}	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$
6	**RecPro4**^{0}	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
7	**RecPro5**$^+$	$\times$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$

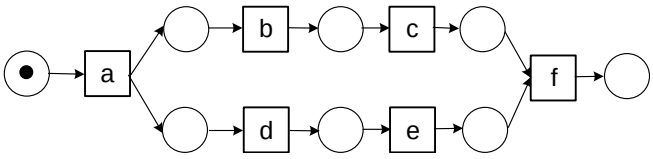

Fig. 3: A process model $m4$.

4.6 Evaluation of Existing Recall and Precision Measures

Several of the existing precision measures are not able to handle non-fitting behavior and remove it by aligning the log to the model. We use a baseline approach for the alignment, which results in a deterministic event log: l is the original event log, which is aligned in a deterministic manner. The resulting event log l' corresponds to unique paths through the model. We use l' to evaluate the propositions.

Evaluation of Existing Recall Measures The previously presented recall measures are evaluated using the corresponding propositions. The results of the evaluation are displayed in Table 1. To ensure the readability of this paper, only the most interesting findings of the evaluation are addressed in the following section. For more details, we refer to [29].

The evaluation of the *causal footprint recall measure* (rec_A) showed that it is deterministic and solely relies on the behavior of the process model. However, the measure violates several propositions such as **RecPro**1^+, **RecPro**3^0, and **RecPro**5^+. These violations are caused by the fact that recall records every difference between the footprint of the log and the model. Behavior that is described by the model but is not observed in the event log has an impact on recall, although Definition 5 states otherwise. To illustrate this, consider m_4 in Figure 3, event log $l_4 = [\langle a, b, c, d, e, f\rangle, \langle a, b, d, c, e, f\rangle]$ and **RecPro**5^+. The traces in l_4 perfectly fit process model m_4. The footprint of l_4 is shown in Table 2 (b). Comparing it to the footprint of m_4 in Table 2 (a) shows mismatches although l_4 is perfectly fitting. These mismatches are caused by the fact that the log does not show all possible behavior of the model and, therefore, the footprint cannot completely detect the parallelism of the model. Consequently 10 of 36 relations of the

Table 2: The causal footprints of m_4 (a), l_4 (b). Mismatching relations are marked in red.

(a)

	a	**b**	**c**	**d**	**e**	**f**
a	#	→	#	→	#	#
b	←	#	→	||	||	#
c	#	←	#	||	||	→
d	←	||	||	#	→	#
e	#	||	||	←	#	→
f	#	#	←	#	←	#

(b)

	a	**b**	**c**	**d**	**e**	**f**
a	#	→	#	#	#	#
b	←	#	→	→	#	#
c	#	←	#	||	→	#
d	#	←	||	#	→	#
e	#	#	←	←	#	→
f	#	#	#	#	←	#

footprint represent mismatches: $rec_A(l_4, m_4) = 1 - \frac{10}{36} = 0.72 \neq 1$. Van der Aalst mentions in [2] that checking conformance using causal footprints is only meaningful if the log is complete in term of directly followed relations. Moreover, the measure also includes precision and generalization aspects, next to recall.

In comparison, recall based on *token replay* (rec_B) depends on the path taken through the model. Due to duplicate activities and silent transitions, multiple paths through a model can be taken when replaying a single trace. Different paths can lead to different numbers of produced, consumed, missing and remaining tokens. Therefore, the approach is neither deterministic nor independent from the structure of the process model and, consequently, violates **RecPro**1^+. The *continued parsing measure* (rec_F) builds on a similar replay principle as token-based replay and also violates **DetPro**$^+$. However, the approach translates the process model into a causal matrix and is therefore independent of its structure.

Table 1 also shows that most measures violate **RecPro**3^0. This is caused by the fact, that we define non-fitting behavior in this paper on a trace level: traces either fit the model or they do not. However, the evaluated approaches measure non-fitting behavior on an event level. A trace consists of fitting and non-fitting events. In cases where the log contains traces with a large number of deviating events, recall can be improved by adding non-fitting traces which contain several fitting and only a few deviating events. To illustrate this, consider *token replay* (rec_B), process model m_5 in Figure 4, $l_5 = [\langle a, b, f, g]$ and $l_6 = l_5 \uplus [\langle a, d, e, f, g \rangle]$. The log l_5 is not perfectly fitting and replaying it on the model results in 6 produced and 6 consumed tokens, as well as 1 missing and 1 remaining token. $rec_B(l_5, m_5) = \frac{1}{2}(1 - \frac{1}{6}) + \frac{1}{2}(1 - \frac{1}{6}) = 0.833$. Event log l_6 was created by adding non-fitting behavior to l_5. Replaying l_6 on m_5 results in $p = c = 13$, $r = m = 2$ and $rec_B(l_7, m_6) = \frac{1}{2}(1 - \frac{2}{13}) + \frac{1}{2}(1 - \frac{2}{13}) = 0.846$. Hence, the additional unfitting trace results in proportionally more fitting events than deviating ones which improves recall: $rec_B(l_6, m_6) < rec_B(l_7, m_6)$.

To overcome the problems arising with the differences between trace-based and event-based fitness, one could alter the definition of **RecPro**3^0 by requiring, that the initial log l_1 only contains fitting behavior ($\tau(l_1) \subseteq \tau(m)$). However, to stay within the scope of this paper, we decide to use the propositions as defined in [3] and keep this suggestion for future work.

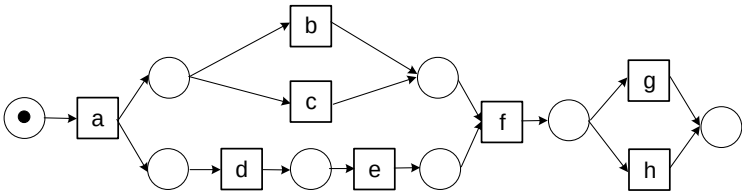

Fig. 4: Petri net m_5

Table 3: Overview of the precision propositions that hold for the existing measures (under the assumption that $l \neq [\,]$, $\tau(m) \neq \emptyset$ and $\langle\rangle \notin \tau(m)$): ✓ means that the proposition holds for any log and model and × means that the proposition does not always hold.

Prop.	Name	$prec_H$	$prec_I$	$prec_J$	$prec_K$	$prec_L$	$prec_M$	$prec_N$	$prec_O$	$prec_P$	$prec_Q$	$prec_R$
1	**DetPro**$^+$	×	×	×	×	✓	×	×	×	✓	✓	✓
2	**BehPro**$^+$	✓	×	×	×	×	×	×	×	✓	✓	✓
8	**PrecPro1**$^+$	✓	×	×	×	×	×	×	×	×	✓	✓
9	**PrecPro2**$^+$	✓	×	✓	×	×	×	×	×	×	×	✓
10	**PrecPro3**^{0}	✓	×	×	×	×	×	×	×	×	×	✓
11	**PrecPro4**^{0}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
12	**PrecPro5**$^+$	✓	×	✓	×	×	✓	✓	✓	✓	✓	✓
13	**PrecPro6**^{0}	✓	×	✓	×	×	✓	✓	✓	✓	✓	✓

Evaluation of Existing Precision Measures The previously presented precision measures are evaluated using the corresponding propositions. The results of the evaluation are displayed in Table 3. To ensure the readability of this paper, only the most interesting findings of the evaluation are addressed in the following section. For more details, we refer to [29].

The evaluation showed that several measures violate the determinism **DetPro**$^+$ proposition. For example, the soundness measure ($prec_H$) solely relies on the number of unique paths of the model $|\tau(m)|$ and unique traces in the log that comply with the process model $|\tau(l) \cap \tau(m)|$. Hence, precision is not defined if the model has infinite possible paths. Additionally to **DetPro**$^+$, behavioral specificity (rec_M) and behavioral precision (rec_N) also violate **BehPro**$^+$. If during the replay of the trace duplicate or silent transitions are encountered, the approach explored which of the available transitions enables the next event in the trace. If no solution is found, one of the transitions is randomly fired, which can lead to different recall values for traces with the same behavior.

Table 3 shows that simple behavioral appropriateness ($prec_I$) violates all but one of the propositions. One of the reason is that it relies on the average number of enabled transitions during replay. Even when the model allows for all exactly observed behavior (and nothing more), precision is not maximal when the model is not strictly sequential. Advanced behavioral appropriateness ($prec_J$) overcomes these problems by relying on

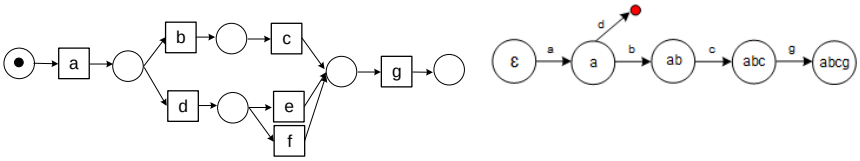

Fig. 5: Petri net m_6 (a) and the alignment automaton describing the state space of $\sigma = \langle a, b, c, g \rangle$ (b)

follow relations. However, it is not deterministic and depends on the structure of the process model.

The results presented in [30] show that ETC precision ($prec_K$ and $prec_L$), weighted negative event precision ($prec_O$) and projected precision ($prec_P$) violate **PrecPro1**$^+$. Additionally, all remaining measures aside from anti-alignment precision ($prec_Q$) and eigenvalue precision ($prec_R$) violate the proposition. The proposition states that removing behavior from a model that does not happen in the event log cannot lower precision. Consider, projected precision ($prec_P$) and a model with a length-one-loop. We remove behavior from the model by restricting the model to only execute the looping activity twice. This changes the DFA of the model since future behavior now depends on how often the looping activity was executed: the DFA contains different states for each execution. If these states show a low local precision, overall precision decreases. Furthermore, [30] showed that ETC precision ($prec_K$ and $prec_L$), projected precision ($prec_P$) and anti-alignment precision ($prec_Q$) also violate **PrecPro2**$^+$.

In general, looking at Table 3 shows that all precision measures, except for soundness ($prec_H$) and eigenvalue precision ($prec_R$) violate **PrecPro3**^{0}, which states that adding unfitting behavior to the event log should not change precision. However, for example, all variations of the ETC-measure ($prec_K$, $prec_L$) align the log before constructing the alignment automaton. Unfitting behavior can be aligned to a trace that was not seen in the log before and introduce new states to the automaton. Consider process model m_6, together with trace $\sigma = \langle a, b, c, g \rangle$ and its alignment automaton displayed in Figure 5. Adding the unfitting trace $\langle a, d, g \rangle$ could result in the aligned trace $\langle a, d, e, g \rangle$ or $\langle a, d, f, g \rangle$. Both aligned traces introduce new states into the alignment automaton, alter the weights assigned to each state and, consequently, change precision. Weighted negative precision ($prec_O$) also violates this proposition. The measure accounts for the number of negative events that actually could fire during trace replay (FP). These false positives are caused by behavior that is shown in the model but not observed in the log. As explained in the context of **RecPro**3^0, although the trace is not fitting when considered as a whole, certain parts of the trace can fit the model. These parts can possibly represent the previously missing behavior in the event log that leads to the wrong classification of negative events. Adding these traces will, therefore, lead to a decrease in false positives and changes precision. $FP(l_1, m) > FP(l_2, m)$ and $\frac{TP(l_1,m)}{(TP(l_1,m)+FP(l_1,m))} < \frac{TP(l_2,m)}{(TP(l_2,m)+FP(l_2,m))}$.

Table 3 shows that $prec_I$, $prec_K$ and $prec_L$ violate proposition **PrecPro6**^{0}, which states that if all modeled behavior was observed, precision should be maximal and unfitting behavior cannot effect precision. $prec_I$ only reports maximal precision if the model is strictly sequential and both ETC measures ($prec_K$ and $prec_L$) can run into problems with models containing silent or duplicate transitions.

The ETC ($prec_K$, $prec_L$) and anti-alignment measures ($prec_Q$) form a special group of measures as they are unable to handle unfitting behavior without pre-processing unfitting traces and aligning them to the process model. Accordingly, we evaluate the conformance measure based on this aligned log. The evaluation of **PrecPro3**^{0} and the ETC measure ($prec_K$, $prec_L$) is an example of the alignment of the log resulting in a violation. However, there are also cases where the proposition only holds because of this alignment. Consider, for example, anti-alignment precision ($prec_Q$) and proposition **PrecPro6**^{0}. By definition, an anti-alignment will always fit the model. Consequently, when computing the distance between the unfitting trace and the anti-alignment it will never be minimal. However, after aligning the log, it exactly contains the modeled behavior, precision is maximal and the proposition holds.

5 Generalization

Generalization is a challenging concept to define, in contrast to recall and precision. As a result, there are different viewpoints within the process mining community on what generalization precisely means. The main reason for this is, that generalization needs to reason about behavior that was not observed in the event log and establish its relation to the model.

The need for a generalization dimension stems from the fact that, given a log, a model can be fitting and precise, but be overfitting. The algorithm that simply creates a model m such that $\tau(m) = \{t \in l\}$ is useless because it is simply enumerating the event log. Consider an unknown process. Assume we observe the first four traces $l_1 = [\langle a, b, c\rangle, \langle b, a, c\rangle, \langle a, b, d\rangle, \langle b, a, d\rangle]$. Based on this we may construct the model m_3 in Figure 2 with $\tau(m_3) = \{\langle a, b, c\rangle, \langle b, a, c\rangle, \langle a, b, d\rangle, \langle b, a, d\rangle\}$. This model allows for all the traces in the event log and nothing more. However, because the real underlying process in unknown, this model may be overfitting event log l_1. Based on just four example traces we cannot be confident that the model m_3 in Figure 2 will be able to explain future behavior of the process. The next trace may as well be $\langle a, c\rangle$ or $\langle a, b, b, c\rangle$. Now assume that we observe the same process for a longer time and consider the first 100 traces (including the initial four): $l_2 = [\langle a, b, c\rangle^{25}, \langle b, a, c\rangle^{25}, \langle a, b, d\rangle^{25}, \langle b, a, d\rangle^{25}]$. After observing 100 traces, we are more confident that model m_3 in Figure 2 is the right model. Intuitively, the probability that the next case will have a trace not allowed by m_3 gets smaller. Now assume that we observe the same process for an even longer time and obtain the event log $l_2 = [\langle a, b, c\rangle^{53789}, \langle b, a, c\rangle^{48976}, \langle a, b, d\rangle^{64543}, \langle b, a, d\rangle^{53789}]$. Although we do not know the underlying process, intuitively, the probability that the next case will have a trace not allowed by m_3 is close to 0. This simple example shows that recall and precision are not enough for conformance checking. We need a generalization notion to address the risk of overfitting example data.

It is difficult to reason about generalization because this refers to unseen cases. Van der Aalst et al. [4] was the first to quantify generalization. In [4], each event is seen as an observation of an activity a in some state s. Suppose that state s is visited n times and that w is the number of different activities observed in this state. Suppose that n is very large and w is very small, then it is unlikely that a new event visiting this state will correspond to an activity not seen before in this state. However, if n and w are of the same order of magnitude, then it is more likely that a new event visiting state s will correspond to an activity not seen before in this state. This reasoning is used to provide a generalization metric. This estimate can be derived under the Bayesian assumption that there is an unknown number of possible activities in state s and that probability distribution over these activities follows a multinomial distribution.

It is not easy to develop an approach that accurately measures generalization. Therefore, some authors define generalization using the notion of a "system" (i.e., a model of the real underlying process). The system refers to the real behavior of the underlying process that the model tries to capture. This can also include the context of the process such as the organization or rules. For example, employees of a company might exceptionally be allowed to deviate from the defined process model in certain situations [20]. In this view, *system fitness* measures the fraction of the behavior of the system that is captured by the model and *system precision* measures how much of the behavior of the model is part of the system. Buijs et al. [11] link this view to the traditional understanding of generalization. They state that both system fitness and system precision are difficult to obtain under the assumption that the system is unknown. Therefore, state-of-the-art discovery algorithms assume that the process model discovered from an event log does not contain behavior outside of the system. In other words, they assume system precision to be 1. Given this assumption, system fitness can be seen as generalization [11]. Janssenswillen et al. [20] agree that in this comparison between the system and the model, especially the system fitness, in fact is what defines generalization. Furthermore, Janssenswillen and Depaire [18] demonstrated the differences between the traditional and the system-based view on conformance checking by showing that state-of-the-art conformance measures cannot reliably assess the similarity between a process model and the underlying system.

Although capturing the unobserved behavior by assuming a model of the system is a theoretically elegant solution, practical applicability of this solution is hindered by the fact that is often impossible to retrieve full knowledge about the system itself. Furthermore, [3] showed the importance of trace probabilities in process models. To accurately represent reality, the system would also need to include probabilities for each of its traces. However, to date, there is only one conformance measure that can actually support probabilistic process models [22]. This approach uses the Earth Movers' distance which measures the effort to transform the distributions of traces of the event log into the distribution of traces of the model.

Some people would argue that one should use cross-validation (e.g., k-fold checking). However, this is a very different setting. Cross validation aims to estimate the quality of a discovery approach and not the quality of a given model given an event log. Of course, one could produce multiple process models using fragments of the event log

and compare them. However, such forms of cross-validation evaluate the quality of the discovery technique and are unrelated to generalization.

For these reasons, we define generalization in the traditional sense.

Definition 9 (Generalization). *A* generalization measure $gen \in \mathcal{L} \times \mathcal{M} \rightarrow [0,1]$ *aims to quantify the probability that new unseen cases will fit the model.*[5]

This definition assumes that a process generates a stream of newly executed cases. The more traces that are fitting and the more redundancy there is in the event, the more certain one can be that the next case will have a trace that fits the model. Note that we deliberately do not formalize the notion of probability, since in real-life we cannot know the real process. Also phenomena like concept drift and contextual factors make it unrealistic to reason about probabilities in a formal sense.

Based on this definition, we present a set of propositions. Note that we do not claim our set of propositions to be complete and invite other researchers who represent a different viewpoint on generalization to contribute to the discussion.

5.1 Generalization Propositions

Generalization ($gen \in \mathcal{L} \times \mathcal{M} \rightarrow [0,1]$) aims to quantify the probability that new unseen cases will fit the model. This conformance dimension is a bit different than the two previously discussed conformance dimensions because it reasons about future *unseen* cases (i.e., not yet in the event log). If the recall is good and the log is complete with lots of repeating behavior, then future cases will most likely fit the model. Analogous to recall, model extensions cannot lower generalization (**GenPro1**$^+$), extending the log with fitting behavior cannot lower generalization (**GenPro2**$^+$), and extending the log with non-fitting behavior cannot improve generalization (**GenPro3**^{0}).

Proposition 14 (GenPro1$^+$**).** *For any* $l \in \mathcal{L}$ *and* $m_1, m_2 \in \mathcal{M}$ *such that* $\tau(m_1) \subseteq \tau(m_2)$*:* $gen(l, m_1) \leq gen(l, m_2)$.

Similar to recall, this proposition implies **BehPro**$^+$. Generalization measures violating **BehPro**$^+$ also violate **GenPro1**$^+$.

Proposition 15 (GenPro2$^+$**).** *For any* $l_1, l_2, l_3 \in \mathcal{L}$ *and* $m \in \mathcal{M}$ *such that* $l_2 = l_1 \uplus l_3$ *and* $\tau(l_3) \subseteq \tau(m)$*:* $gen(l_1, m) \leq gen(l_2, m)$.

Proposition 16 (GenPro3^0**).** *For any* $l_1, l_2, l_3 \in \mathcal{L}$ *and* $m \in \mathcal{M}$ *such that* $l_2 = l_1 \uplus l_3$ *and* $\tau(l_3) \subseteq \overline{\tau}(m)$*:* $gen(l_1, m) \geq gen(l_2, m)$.

Duplicating the event log does not necessarily influence recall and precision. According to propositions **RecPro4**^{0} and **PrecPro4**^{0} this should have no effect on recall and precision. However, making the event log more redundant, should have an effect

[5] Note that the term "probability" is used here in an informal manner. Since we only have example observations and no knowledge of the underlying (possibly changing) process, we cannot compute such a probability. Of course, unseen cases can have traces that have been observed before.

on generalization. For fitting logs, adding redundancy without changing the distribution can only improve generalization (**GenPro4**$^+$). For non-fitting logs, adding redundancy without changing the distribution can only lower generalization (**GenPro5**$^+$). Note that **GenPro4**$^+$ and **GenPro5**$^+$ are special cases of **GenPro6**^{0} and **GenPro7**^{0}. **GenPro6**^{0} and **GenPro7**^{0} consider logs where some traces are fitting and others are not. For a log where more than half of the traces is fitting, duplication can only improve generalization (**GenPro6**^{0}). For a log where more than half of the traces is non-fitting, duplication can only lower generalization (**GenPro7**^{0}).

Proposition 17 (GenPro4$^+$). *For any $l \in \mathcal{L}$, $m \in \mathcal{M}$, and $k \geq 1$ such that $\tau(l) \subseteq \tau(m)$: $gen(l^k, m) \geq gen(l, m)$.*

Proposition 18 (GenPro5$^+$). *For any $l \in \mathcal{L}$, $m \in \mathcal{M}$, and $k \geq 1$ such that $\tau(l) \subseteq \overline{\tau}(m)$: $gen(l^k, m) \leq gen(l, m)$.*

Proposition 19 (GenPro6^0). *For any $l \in \mathcal{L}$, $m \in \mathcal{M}$, and $k \geq 1$ such that most traces are fitting ($|[t \in l \mid t \in \tau(m)]| \geq |[t \in l \mid t \notin \tau(m)]|$): $gen(l^k, m) \geq gen(l, m)$.*

Proposition 20 (GenPro7^0). *For any $l \in \mathcal{L}$, $m \in \mathcal{M}$, and $k \geq 1$ such that most traces are non-fitting ($|[t \in l \mid t \in \tau(m)]| \leq |[t \in l \mid t \notin \tau(m)]|$): $gen(l^k, m) \leq gen(l, m)$.*

When the model allows for any behavior, clearly the next case will also be fitting (**GenPro8**^{0}). Nevertheless, it is marked as controversial because the proposition would also need to hold for an empty event log.

Proposition 21 (GenPro8^0). *For any $l \in \mathcal{L}$ and $m \in \mathcal{M}$ such that $\tau(m) = \mathcal{T}$: $gen(l, m) = 1$.*

5.2 Existing Generalization Measures

The following sections introduce several state-of-the-art generalization measures, before they will be evaluated using the corresponding propositions.

Alignment generalization (gen_S). Van der Aalst et al. [4] also introduce a measure for generalization. This approach considers each occurrence of a given event e as observation of an activity in some state s. The approach is parameterized by a $state_M$ function that maps events onto states in which they occurred. For each event e that occurred in state s the approach counts how many different activities w were observed in that state. Furthermore, it counts the number of visits n to this state. Generalization is high if n is very large and w is small, since in that case, it is unlikely that a new trace will correspond to unseen behavior in that state.

Weighted negative event generalization (gen_T). Aside from improving the approach of Goedertier et al. [16] van den Broucke et al. [31] also developed a generalization measure based on weighted negative events. It defines allowed generalizations AG which represent events, that could be replayed without errors and confirm that the model is

Table 4: An overview of the generalization propositions that hold for the measures: (assuming $l \neq [\,]$, $\tau(m) \neq \emptyset$ and $\langle\rangle \notin \tau(m)$): $\surd$ means that the proposition holds for any log and model and $\times$ means that the proposition does not always hold.

Proposition	Name	gen_S	gen_T	gen_U
1	**DetPro**$^+$	$\surd$	$\times$	$\surd$
2	**BehPro**$^+$	$\surd$	$\times$	$\times$
14	**GenPro1**$^+$	$\times$	$\times$	$\times$
15	**GenPro2**$^+$	$\times$	$\times$	$\times$
16	**GenPro3**^{0}	$\times$	$\times$	$\times$
17	**GenPro4**$^+$	$\surd$	$\surd$	$\surd$
18	**GenPro5**$^+$	$\times$	$\surd$	$\surd$
19	**GenPro6**^{0}	$\surd$	$\surd$	$\surd$
20	**GenPro7**^{0}	$\times$	$\surd$	$\surd$
21	**GenPro8**^{0}	$\times$	$\surd$	$\times$

general and disallowed generalizations DG which are generalization events, that could not be replayed correctly. If during replay a negative event e is encountered that actually was enabled the AG value is increased by $1 - weight(e)$. Similarly, if a negative event is not enabled the DG value is increased by $1 - weight(e)$. The more disallowed generalizations are encountered during log replay the lower generalization.

Anti-alignment generalization (gen_U). Van Dongen et al. [13] also introduce an anti-alignment generalization and build on the principle that with a generalizing model, newly seen behavior will introduce new paths between the states of the model, however no new states themselves. Therefore, they define a recovery distance d_{rec} which measures the maximum distance between the states visited by the log and the states visited by the anti-alignment γ. A perfectly generalizing model according to van Dongen et al. [13] has the maximum distance to the anti-alignment with minimal recovery distance. Similar to recall they define trace-based and log-based generalization. Finally, anti-alignment generalization is the weighted combination of trace-based and log-based anti-alignment generalization.

5.3 Evaluation of Existing Generalization Measures

The previously presented generalization measures are evaluated using the corresponding propositions. The results of the evaluation are displayed in Table 4. To improve the readability of this paper, only the most interesting findings of the evaluation are addressed in the following section. For a detailed evaluation, we refer to [29].

Table 4 displays that alignment based generalization (gen_S) violates several propositions. Generalization is not defined if there are unfitting traces since they cannot be mapped to states of the process model. Therefore, unfitting event logs should be aligned to fit to the model before calculating generalization. Aligning a non-fitting log and duplicating it will result in more visits to each state visited by the log. Therefore, adding

non-fitting behavior increases generalization and violates the propositions **GenPro3**^{0}, **GenPro5**$^+$ and **GenPro7**^{0}.

In comparison, weighted negative event generalization (gen_T) is robust against the duplication of the event log, even if it contains non-fitting behavior. However, this measure violates **DetPro**$^+$, **BehPro**$^+$, **GenPro1**$^+$, **GenPro2**$^+$ and **GenPro3**^{0}, which states that extending the log with non-fitting behavior cannot improve generalization. However, in this approach, negative events are assigned a weight which indicates how certain the log is about these events being negative ones. Even though the added behavior is non-fitting it might still provide evidence for certain negative events and therefore increase their weight. If these events are then not enabled during log replay the value for disallowed generalizations (DG) decreases $DG(l_1, m) < DG(l_2, m)$ and generalization improves: $\frac{AG(l_1,m)}{AG(l_1,m)+DG(l_1,m)} < \frac{AG(l_2,m)}{AG(l_2,m)+DG(l_2,m)}$.

Table 4 shows that anti-alignment generalization (gen_U) violates several propositions. The approach considers markings of the process models as the basis for the generalization computation which violates the behavioral proposition. Furthermore, the measure cannot handle if the model displays behavior that has not been observed in the event log. If the unobserved model behavior and therefore also the anti-alignment introduced a lot of new states which were not visited by the event log, the value of the recovery distance increases and generalization is lowered. This clashes with propositions **GenPro1**$^+$ and **GenPro8**$^+$. Finally, the approach also excludes unfitting behavior from its scope. Only after aligning the event log, generalization can be computed. As a result, the measure fulfills **GenPro5**$^+$, **GenPro6**^{0} and **GenPro7**^{0}, but violates **GenPro3**^{0}.

6 Conclusion

With the process mining field maturing and more commercial tools becoming available [21], there is an urgent need to have a set of agreed-upon measures to determine the quality of discovered processes models. We have revisited the 21 conformance propositions introduced in [3] and illustrated their relevance by applying them to baseline measures. Furthermore, we used the propositions to evaluate currently existing conformance measures. This evaluation uncovers large differences between existing conformance measures and the properties that they possess in relation to the propositions. It is surprising that seemingly obvious requirements are not met by today's conformance measures. However, there are also measures that do meet all the propositions.

It is important to note that we do not consider the set of propositions to be complete. Instead, we consider them to be an initial step to start the discussion on what properties are to be desired from conformance measures, and we encourage others to contribute to this discussion. Moreover, we motivate researchers to use the conformance propositions as design criteria for the development of novel conformance measures.

One relevant direction of future work is in the area of conformance propositions that have a more fine-grained focus than the trace-level, i.e., that distinguish between *almost fitting* and *completely non-fitting* behavior. Another relevant area of future work is in the direction of *probabilistic conformance measures*, which take into account branching probabilities in models, and their desired properties. Even though event logs provide

insights into the probability of traces, thus far all existing conformance checking techniques ignore this point of view. In [1, 3], we already showed that probabilities can be used to provide more faithful definitions of recall and precision. In [22], we moved one step further and provide the first *stochastic conformance checking technique* using the so-called Earth Movers' Distance (EMD). This conformance checking approach considers the stochastic characteristics of both the event log and the process model. It measures the effort to transform the distribution of traces of the event log into the distribution of traces of the model. This way one can overcome many of the challenges identified in this paper.

Acknowledgements We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

1. W.M.P. van der Aalst. Mediating Between Modeled and Observed Behavior: The Quest for the "Right" Process. In *IEEE International Conference on Research Challenges in Information Science (RCIS 2013)*, pages 31–43. IEEE Computing Society, 2013.
2. W.M.P. van der Aalst. *Process Mining: Data Science in Action*. Springer-Verlag, Berlin, 2016.
3. W.M.P. van der Aalst. Relating Process Models and Event Logs: 21 Conformance Propositions. In W.M.P. van der Aalst, R. Bergenthum, and J. Carmona, editors, *Workshop on Algorithms & Theories for the Analysis of Event Data (ATAED 2018)*, pages 56–74. CEUR Workshop Proceedings, 2018.
4. W.M.P. van der Aalst, A. Adriansyah, and B. van Dongen. Replaying History on Process Models for Conformance Checking and Performance Analysis. *WIREs Data Mining and Knowledge Discovery*, 2(2):182–192, 2012.
5. W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.
6. A. Adriansyah, B. van Dongen, and W.M.P. van der Aalst. Conformance Checking using Cost-Based Fitness Analysis. In C.H. Chi and P. Johnson, editors, *IEEE International Enterprise Computing Conference (EDOC 2011)*, pages 55–64. IEEE Computer Society, 2011.
7. A. Adriansyah, J. Munoz-Gama, J. Carmona, B. F. van Dongen, and W. M. P. van der Aalst. Alignment based precision checking. In M. La Rosa and P. Soffer, editors, *Business Process Management Workshops*, pages 137–149. Springer, 2013.
8. A. Augusto, A. Armas-Cervantes, R. Conforti, M. Dumas, M. La Rosa, and D. Reissner. Abstract-and-compare: A family of scalable precision measures for automated process discovery. In M. Weske, M. Montali, I. Weber, and J. vom Brocke, editors, *Proceedings of the International Conference on Business Process Management*, pages 158–175, Cham, 2018. Springer International Publishing.
9. J.C.A.M. Buijs. *Flexible evolutionary algorithms for mining structured process models*. PhD thesis, Department of Mathematics and Computer Science, 2014.
10. J.C.A.M. Buijs, B.F. van Dongen, and W.M.P. van der Aalst. On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery. In R. Meersman, S. Rinderle, P. Dadam, and X. Zhou, editors, *OTM Federated Conferences, 20th International Conference on Cooperative Information Systems (CoopIS 2012)*, volume 7565 of *Lecture Notes in Computer Science*, pages 305–322. Springer-Verlag, Berlin, 2012.

11. J.C.A.M. Buijs, B.F. van Dongen, and W.M.P. van der Aalst. Quality Dimensions in Process Discovery: The Importance of Fitness, Precision, Generalization and Simplicity. *International Journal of Cooperative Information Systems*, 23(1):1–39, 2014.
12. J. Carmona, B. van Dongen, A. Solti, and M. Weidlich. *Conformance Checking: Relating Processes and Models*. Springer-Verlag, Berlin, 2018.
13. B.F. van Dongen, J. Carmona, and T. Chatain. A Unified Approach for Measuring Precision and Generalization Based on Anti-alignments. In M. La Rosa, P. Loos, and O. Pastor, editors, *International Conference on Business Process Management (BPM 2016)*, volume 9850 of *Lecture Notes in Computer Science*, pages 39–56. Springer-Verlag, Berlin, 2016.
14. B.F. van Dongen, J. Carmona, T. Chatain, and F. Taymouri. Aligning Modeled and Observed Behavior: A Compromise Between Computation Complexity and Quality. In E. Dubois and K. Pohl, editors, *International Conference on Advanced Information Systems Engineering (Caise 2017)*, volume 10253 of *Lecture Notes in Computer Science*, pages 94–109. Springer-Verlag, Berlin, 2017.
15. L. Garcia-Banuelos, N. van Beest, M. Dumas, M. La Rosa, and W. Mertens. Complete and Interpretable Conformance Checking of Business Processes. *IEEE Transactions on Software Engineering*, 44(3):262–290, 2018.
16. S. Goedertier, D. Martens, J. Vanthienen, and B. Baesens. Robust Process Discovery with Artificial Negative Events. *Journal of Machine Learning Research*, 10:1305–1340, 2009.
17. G. Greco, A. Guzzo, L. Pontieri, and D. Saccà. Discovering Expressive Process Models by Clustering Log Traces. *IEEE Transaction on Knowledge and Data Engineering*, 18(8):1010–1027, 2006.
18. G. Janssenswillen and B. Depaire. Towards confirmatory process discovery: Making assertions about the underlying system. *Business & Information Systems Engineering*, Dec 2018.
19. G. Janssenswillen, N. Donders, T. Jouck, and B. Depaire. A comparative study of existing quality measures for process discovery. *Information Systems*, 50(1):2:1–2:45, 2017.
20. G. Janssenswillen, T. Jouck, M. Creemers, and B. Depaire. Measuring the quality of models with respect to the underlying system: An empirical study. In M. La Rosa, P. Loos, and O. Pastor, editors, *Business Process Management*, pages 73–89, Cham, 2016. Springer International Publishing.
21. M. Kerremans. Gartner Market Guide for Process Mining, Research Note G00353970. www.gartner.com, 2018.
22. S. Leemans, A. Syring, and W.M.P. van der Aalst. Earth Movers' Stochastic Conformance Checking. In T. Hildebrandt, B.F. van Dongen, M. Röglinger, and J. Mendling, editors, *Business Process Management Forum (BPM Forum 2019)*, volume 360 of *Lecture Notes in Business Information Processing*, pages 127–143. Springer-Verlag, Berlin, 2019.
23. S.J.J. Leemans, D. Fahland, and W.M.P. van der Aalst. Scalable Process Discovery and Conformance Checking. *Software and Systems Modeling*, 17(2):599–631, 2018.
24. F. Mannhardt, M. de Leoni, H.A. Reijers, and W.M.P. van der Aalst. Balanced Multi-Perspective Checking of Process Conformance. *Computing*, 98(4):407–437, 2016.
25. J. Munoz-Gama and J. Carmona. A Fresh Look at Precision in Process Conformance. In R. Hull, J. Mendling, and S. Tai, editors, *Business Process Management (BPM 2010)*, volume 6336 of *Lecture Notes in Computer Science*, pages 211–226. Springer-Verlag, Berlin, 2010.
26. A. Polyvyanyy, A. Solti, M. Weidlich, C. Di Ciccio, and J. Mendling. Behavioural Quotients for Precision and Recall in Process Mining. Technical report, University of Melbourne, 2018.
27. A. Rozinat and W.M.P. van der Aalst. Conformance Checking of Processes Based on Monitoring Real Behavior. *Information Systems*, 33(1):64–95, 2008.
28. A. Rozinat, A.K. Alves de Medeiros, C.W. Günther, A.J.M.M. Weijters, and W.M.P. van der Aalst. The Need for a Process Mining Evaluation Framework in Research and Practice. In

M. Castellanos, J. Mendling, and B. Weber, editors, *Informal Proceedings of the International Workshop on Business Process Intelligence (BPI 2007)*, pages 73–78. QUT, Brisbane, Australia, 2007.
29. A.F. Syring, N. Tax, and W.M.P. van der Aalst. Evaluating Conformance Measures in Process Mining using Conformance Propositions (Extended Version). *CoRR*, arXiv:1909.02393, 2019.
30. N. Tax, X. Lu, N. Sidorova, D. Fahland, and W.M.P. van der Aalst. The Imprecisions of Precision Measures in Process Mining. *Information Processing Letters*, 135:1–8, 2018.
31. S. K. L. M. vanden Broucke, J. De Weerdt, J. Vanthienen, and B. Baesens. Determining process model precision and generalization with weighted artificial negative events. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1877–1889, Aug 2014.
32. J. De Weerdt, M. De Backer, J. Vanthienen, and B. Baesens. A Multi-Dimensional Quality Assessment of State-of-the-Art Process Discovery Algorithms Using Real-Life Event Logs. *Information Systems*, 37(7):654–676, 2012.
33. J. De Weerdt, M. De Backer, J. Vanthienen, and B. Baesens. A Robust F-measure for Evaluating Discovered Process Models. In N. Chawla, I. King, and A. Sperduti, editors, *IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2011)*, pages 148–155, Paris, France, April 2011. IEEE.
34. A.J.M.M. Weijters, W.M.P. van der Aalst, and A.K. Alves de Medeiros. Process Mining with the Heuristics Miner-algorithm. BETA Working Paper Series, WP 166, Eindhoven University of Technology, Eindhoven, 2006.