Computing Conformance Checking Modulo Theories for Multi-Perspective Processes

Paolo Felli [1], **Alessandro Gianola** [2], **Marco Montali** [2], **Andrey Rivkin** [2], **Sarah Winkler** [2]

[1] University of Bologna, Bologna, Italy
[2] Free University of Bozen-Bolzano, Bolzano, Italy
paolo.felli@unibo.it, {gianola,montali,rivkin,winkler}@inf.unibz.it

Overview

We give an overview of recent results on the application of techniques based on Satisfiability Modulo Theories (SMT) in the context of multi-perspective Process Mining: specifically, we present an approach that leverage SMT-based techniques for performing conformance checking of data-aware processes over logs with (in)complete data.

Process mining (PM) is an active research field that originally stems from business process management (BPM), and involves frameworks and methods taken from data science, formal methods and AI. The original idea of PM is to provide a series of techniques supporting analysis of operational processes based on event log data. Such techniques can be subdivided into three main groups (van der Aalst 2016): *(i) process discovery* focuses on constructing a process model that is representative of all the behaviors observed in the event log; *(ii) conformance checking* searches for deviations and commonalities between a given process model and an event log; *(iii) enhancement*, offers a set of on-line techniques using incoming event data so as to improve and optimize an already existing process.

In this paper, we deal with the conformance checking (Carmona et al. 2018) task whose main idea, as we have mentioned above, is the detection of behavioral discrepancies between the process model and a log. Notably, one may choose among different process representations, ranging from classical workflow nets (van der Aalst 2011) to declarative data-aware processes (Burattin, Maggi, and Sperduti 2016), as well as logs, considering those with additional data on activity payloads, incomplete information, relations between objects used by operational processes etc. In this multitude of approaches, notice that conformance checking quite often becomes more challenging when one goes beyond the control-flow perspective and, as we are interested here, tries to account for, e.g., unbounded data.

Here, we use data Petri nets (DPNs for short) as the reference process models. DPNs have been extensively studied in the context of formal verification (de Leoni, Felli, and Montali 2018; Felli, de Leoni, and Montali 2019) as well as process mining (Mannhardt et al. 2016b,a; Mannhardt 2018). Notably, various techniques proposed for DPNs are rather ad-hoc and work under restrictive assumptions on the data dimension (e.g., limited support of data types).

To alleviate such restrictions, we introduced in (Felli et al. 2021) a framework based on well-established automated reasoning techniques such as the ones provided by Satisfiability Modulo Theories (SMT). SMT provides a more universal, algorithmically robust and flexible approach, whose use allows us to support complex forms of unbounded data constrained by expressive first-order theories: SAT-based approaches would be sufficient to capture the control-flow behavior in isolation, but not its interaction with data. Instead, by relying on an SMT backend, we can support data and operations from a variety of theories such as arithmetics, without significantly restricting the data dimension. We also discuss how this framework can be adapted to address data-aware conformance checking over uncertain logs, by providing a suitably extended notion of alignment. In this case, event logs incorporate complex forms of uncertainty, regarding the recorded timestamps, data values, and/or events.

The framework

In this section we present our SMT-based approach to conformance checking (Felli et al. 2021). In a nutshell, the conformance checking problem takes as input a log/a trace and a process model, and computes various artefacts linking observed and modeled behaviors. Such artefacts are then used for such purposes for deviation detection, (process model) quality metrics computation etc. We first briefly discuss the type of logs we are interested in as well as the process models we use, and then introduce the conformance checking problem together with the artefacts of interest. Finally, we discuss how to encode all these elements into SMT.

Logs with uncertainties and Petri nets with payloads. Given a set A of activity labels and a set of attributes names Y, an event is a tuple $(b, \alpha, \mathsf{c}, \mathsf{LA}, \mathsf{TS})$, where $b \in A$, α is a (partial) payload function associating values to attributes from Y, $\mathsf{c} \in (0, 1]$ is a confidence that the event actually happened, LA is a set of activities with related confidence values, and TS is a finite set of elements from (or an interval over) a totally ordered set of timestamps. The last three values are crucial for specifying the following uncertainties: *(i) uncertain events*, which come with confidence values describing how certain we are about the fact that recorded events actually happened during the process execution; *(ii) uncertain timestamps* that are either coarse or come as intervals, which in turn affect the ordering of the events in

the log; *(iii) uncertain activities* that in bundle (together with corresponding confidence values) are assigned to a concrete event when we are not certain which of the activities actually caused it; *(iv) uncertain data values* which, for an attribute, come as a set or an interval of possible values. For a set of events $\mathcal{E}$, a log trace is a sequence $\mathbf{e} \in \mathcal{E}^*$ and a log is a multiset of log traces.

Events come with activity payloads, differing from the majority of process mining tasks in which one essentially deals with processes characterized by timestamps and activity labels only (van der Aalst 2011). Processes with payloads can be represented using data Petri nets (DPNs for short) (Mannhardt et al. 2016b,a; Mannhardt 2018), our reference process model. A DPN is a labeled Petri net extended with *read* and *write* guards defined over a finite set of (typed) net variables V. Every guard is a boolean combination of atoms $x_1 \odot x_2$, where $\odot$ is a type-specific predicate.

The execution semantics of DPNs extends the one of place-transition nets with the ability to check and manipulate the net's variables via transition guards. A transition is *enabled* iff all the input places of the transition contain sufficiently many tokens to consume, and its read and write guards are satisfied under a given "firing mode" β assigning values only to variables of the guard. For the read guards, values are picked from the current net state variable assignment, whereas for write guards values are provided by β. An enabled transition may *fire*, consuming the necessary amount of tokens from its input places and producing tokens in its output places, and also updating the state using β. All other values assigned to V remain the same.

Here DPNs are considered relaxed data sound, i.e., there exists at least one sequence of transition firings bringing to a given final marking. Such transitions are called valid.

The conformance checking problem. Conformance checking looks into comparing the observed behavior in the log with the one allowed by the process model. This comparison can be achieved by computing various analytic (or conformance) artefacts (Carmona et al. 2018). One of the main artefacts is given by (optimal) alignments, where an *alignment* is a finite sequence of *moves* (event-firing pairs) s.t. each (e, f) is *(i)* a log move, if $e \in \mathcal{E}$ and $f = \gg$ ($\gg$ is a symbol denoting skipping); *(ii)* a model move, if $e = \gg$ and f is a valid transition firing; *(iii)* a synchronous move otherwise. Given a log trace and a process model, a corresponding alignment is computed by finding a complete process run that is "the best" among all those the process model can generate. This is done by searching for an *optimal* alignment using a specific cost function, which can be realized as a distance between two finite strings (Adriansyah 2014). In case of logs with uncertainties, one looks into alignments obtained for *realizations* – possible sequentializations of (a subset of the) events with uncertainty in a given trace so that timestamp uncertainties are resolved (Felli et al. 2022).

Encoding into SMT. We now present how the conformance checking problem and related artefacts can be *modularly* encoded using SMT. The detailed encoding can be found in (Felli et al. 2021, 2022), whereas here we focus on how conformance checking can be solved using SMT and on the modular organization of the encoding, where each module accounts for one of the problem components and can be seamlessly replaced when, for example, other types of process models or conformance artefacts are required. The main modules of this encoding are as follows: ***(i)*** the executable process model (**EPM**) module, which realizes the execution semantics of the input net by symbolically encoding all possible valid process runs; ***(ii)*** the log trace (**LT**) module, which accounts for the trace realization in case it contains any uncertainties; ***(iii)*** the conformance artefacts (**CA**) module, which accounts for a conformance artefact of interest as well as for its related *decision/optimization problem*, i.e., the task that needs to be solved (or decided) in order to compute such artefact. Each module (but the part of **CA** related to the decision problem) is represented as a conjunction of SMT constraints. It is important to notice that the process model **EPM** is usually given by DPNs, and encodes process runs whose length does not exceed a given upper bound. This, however, does not affect the optimality of the final solution and has been extensively discussed in (Felli et al. 2021, 2022). The **LT** module naturally encodes all the aforementioned uncertainties (if any) and, in case of the uncertainty on timestamps, for related events it provides an additional encoding to find a suitable ordering for them.

All the SMT constraints are then put into one conjunction, which is passed as input to an SMT solver in order to solve the associated decision problem. For the case of alignments, **CA** encodes a selected distance-based cost function defined as the sum of contributions that only depend on the local discrepancies in the moves of the considered alignment (i.e., inductively), and this is done using a finite set of variables that are later used to extract the optimal alignment solution. However, since there are in principle many possible satisfiable solutions, in order to find the 'best/optimal one', we solve a constrained optimization problem for which a satisfying assignment to all the variables in the encoding is so that the total cost (represented as a single variable carrying the final result of the aforementioned inductive cost encoding) is minimal. This allows us to find a model run which is the 'closest one' to the trace.

In the same manner, one can adjust **CA** by changing the decision problem associated to the specific conformance artefact of interest. We proposed in (Felli et al. 2021) slight encoding modifications so as to account for multi- and anti-alignments (Chatain and Carmona 2016; Boltenhagen, Chatain, and Carmona 2019). To capture different results while computing conformance artefacts, one can also modify **CA** so as to account for different cost functions. For example, in (Felli et al. 2022) we demonstrated that one can employ a generic cost function whose components can be flexibly instantiated to homogeneously account for a variety of concrete situations where uncertainty come into play.

After an SMT solver finds a satisfiable solution, we use the known correspondence between an edit distance and alignments (Needleman and Wunsch 1970) to decode the solution and obtain the optimal alignment.

References

Adriansyah, A. 2014. *Aligning observed and modeled behavior*. Ph.D. thesis, Technische Universiteit Eindhoven.

Boltenhagen, M.; Chatain, T.; and Carmona, J. 2019. Encoding Conformance Checking Artefacts in SAT. In *Proc. Business Process Management Workshops 2019*, volume 362 of *LNCS*, 160–171.

Burattin, A.; Maggi, F. M.; and Sperduti, A. 2016. Conformance checking based on multi-perspective declarative process models. *Expert Syst. Appl.*, 65: 194–211.

Carmona, J.; van Dongen, B. F.; Solti, A.; and Weidlich, M. 2018. *Conformance Checking - Relating Processes and Models*. Springer.

Chatain, T.; and Carmona, J. 2016. Anti-alignments in Conformance Checking – The Dark Side of Process Models. In *Proc. 37th Petri Nets*, volume 9698 of *LNCS*, 240–258.

de Leoni, M.; Felli, P.; and Montali, M. 2018. A Holistic Approach for Soundness Verification of Decision-Aware Process Models. In *Proc. 37th International Conference on Conceptual Modeling*, volume 11157 of *LNCS*, 219–235.

Felli, P.; de Leoni, M.; and Montali, M. 2019. Soundness Verification of Decision-Aware Process Models with Variable-to-Variable Conditions. In *Proc. 19th ACSD*, 82–91. IEEE.

Felli, P.; Gianola, A.; Montali, M.; Rivkin, A.; and Winkler, S. 2021. CoCoMoT: Conformance Checking of Multi-perspective Processes via SMT. In *Proc. of BPM 2021*, volume 12875 of *LNCS*, 217–234. Springer.

Felli, P.; Gianola, A.; Montali, M.; Rivkin, A.; and Winkler, S. 2022. Conformance Checking with Uncertainty via SMT. In *Proc. of BPM 2022*, volume 13420 of *LNCS*, 199–216. Springer.

Mannhardt, F. 2018. *Multi-perspective Process Mining*. Ph.D. thesis, Technical University of Eindhoven.

Mannhardt, F.; de Leoni, M.; Reijers, H.; and van der Aalst, W. 2016a. Balanced multi-perspective checking of process conformance. *Computing*, 98(4): 407–437.

Mannhardt, F.; de Leoni, M.; Reijers, H. A.; and van der Aalst, W. M. P. 2016b. Decision Mining Revisited - Discovering Overlapping Rules. In *Proc. 28th CAiSE*, volume 9694 of *LNCS*, 377–392.

Needleman, S. B.; and Wunsch, C. D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3): 443–453.

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

van der Aalst, W. M. P. 2016. *Process Mining - Data Science in Action, Second Edition*. Springer.