AI4BPM Student Contribution
Solving Declarative Process Mining Problems Using Declarative Problem Solving

Francesco Chiariello
DIAG
Sapienza University of Rome
chiariello@diag.uniroma1.it

Abstract

My research is at the intersection of three areas: (i) Temporal Logics, (ii) Declarative Problem Solving, and (iii) Process Mining. In particular, I focus on solving problems from Declarative Process Mining - where process models are temporal logic formulae - using declarative approaches like *Satisfiability testing* (SAT), *Answer Set Programming* (ASP), and *Automated Planning* (AP). My main result is the proposal of a new technique for Temporal Reasoning in ASP, based on exploiting the automata representation of the formulae, and that finds application to many Process Mining problems.

Research Summary

Declarative Process Mining (DPM) focuses on analyzing event logs of processes modeled in a constraint-based fashion. Processes are therefore seen as sets of rules in some logical formalism like DECLARE (Pesic, Schonenberg, and van der Aalst 2007; van der Aalst, Pesic, and Schonenberg 2009) or *Linear Temporal Logic on finite traces* (LTL_f) (De Giacomo and Vardi 2013).
Using a declarative modeling approach allows one to easily specify the desired behaviors: everything that does not violate the rules is allowed. Contrast this with the imperative approach where one has to explicitly define the desired behaviors using models like Petri nets (Van der Aalst 1998) or BPMN (White 2004; Allweyer 2016). This often results in over-constraining the model, i.e., ruling-out behaviors that would be desirable but have not been modeled. Besides those type II errors (the excluding of behaviors that should be allowed) the declarative modeling approach is also less prone to type I errors (the allowing of behaviors that should be excluded) thanks to its high-level interpretability.
DECLARE consists of a collection of *constraint templates* that allows to express relations between pairs of activities (e.g. choice, coexistence, precedence) or properties of single activities (i.e. bounds on the number of occurrences of the activity). It has been shown that the semantics of DECLARE can be grounded into LTL_f, which constitutes thus a more general language, not being limited to predefined patterns and allowing to easily express constraints involving more than two activities. Since it allows more flexibility, the focus of my research is on LTL_f. An instrumental result, relating LTL_f with automata theory, shows how given an LTL_f formula, it is possible to construct a finite-state automaton that accepts exactly the traces satisfying the formula. Once such an automaton is constructed, reasoning becomes pretty straightforward.

The main idea of my published work (Chiariello, Maggi, and Patrizi 2022a,b,c) is to exploit the automata-based representation with an ASP system in order to efficiently solve problems involving temporal specifications.
Compared with non-declarative approaches, ASP allows for rapid implementation and ease of use, removing the burden of actually implementing the solving algorithm. This also reduces the possibility of errors and makes transparent what the solver is actually doing. Compared with SAT, thanks to the minimality of its semantics, ASP allows one to represent automata more easily and thus solve more efficiently the relative problems.
The problems I have successfully solved with that approach are Log Generation, Conformance Checking, and Query Checking. Log generation (Skydanienko et al. 2018) consists in generating a set of traces compliant with a process model, here represented as a set of constraints. Conformance checking (Burattin, Maggi, and Sperduti 2016) consists in checking whether the traces of a log are compliant with a given process model. Query checking (Räim et al. 2014) consists in checking constraint templates against a log in order to find the instantiations compliant with the log. In other words, query checking is about finding the properties of a process by analyzing the corresponding log. Those problems are addressed both from a control-flow and a data perspective. ASP is also particularly suited for working with data thanks to its logic-programming-based syntax, built-in arithmetic functions and comparison predicates which allow to easily express conditions over the attributes of activities and integrates them in a logical framework.

Being my proposed solution approach so general it can virtually be applied to any process mining problem. Another problem I have considered is Trace Alignment which consists in correcting, in an optimal way, the traces not compliant with a given process model. This can be formulated as cost-optimal planning (De Giacomo et al. 2017) and solved using, e.g., the `FastDownward` planning system (Helmert 2006). Planning capabilities are also provided by the ASP system `clingo` (Gebser et al. 2017) which however showed worse performances than the state-of-the-art on the trace alignment problem.

The inverse problem of trace alignment (modifying traces to conform to a process model) is Model Repair (Fahland and van der Aalst 2015; Polyvyanyy et al. 2017), where the traces are assumed to be correct and the objective is to modify an input model in order to produce a new model compliant with the traces that is as close as possible to the input. In the same fashion, this problem can be formulated as cost-optimal planning and declarative techniques can be applied. In particular, one could build on the ASP-based conformance checking technique and add actions to modify the model. Since ASP makes it easy to check conformance it could be a better option than standard AP techniques. Note also that by providing an empty model, model repair reduces to Process Discovery (van der Aalst 2010) where the optimality guarantees the minimality of the discovered model.

Finally, a comment on the word "declarative". In declarative process mining, it refers to process modeling. In declarative problem solving, instead, such a word refers to an approach consisting in declaring what counts as a solution to a particular problem and using out-of-the-box tools to find them. While it is natural to try a declarative approach to solve a problem stated (partially) declaratively, it is worth emphasizing that we go through finite-state automata, constituting a procedural representation of the process. Therefore it is clear that, once made the appropriate changes, one could apply declarative problem solving techniques also for other procedural models such as Petri nets.

Acknowledgments

Work partly supported by the ERC Advanced Grant WhiteMech (No. 834228), the EU ICT-48 2020 project TAILOR (No. 952215), the EU ICT-49 2021 project AIPlan4EU (No. 101016442), and the PRIN project RIPER (No. 20203FFYLK).

References

Allweyer, T. 2016. *BPMN 2.0: introduction to the standard for business process modeling*. BoD–Books on Demand.

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

Chiariello, F.; Maggi, F. M.; and Patrizi, F. 2022a. ASP-Based Declarative Process Mining. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5): 5539–5547. Number: 5.

Chiariello, F.; Maggi, F. M.; and Patrizi, F. 2022b. ASP-Based Declarative Process Mining (Extended Abstract). In *Proceedings of the 38th International Conference on Logic Programming (Technical Communications) (ICLP)*. Electronic Proceedings in Theoretical Computer Science (EPTCS).

Chiariello, F.; Maggi, F. M.; and Patrizi, F. 2022c. A tool for compiling Declarative Process Mining problems in ASP. *Software Impacts*, 100435.

De Giacomo, G.; Maggi, F. M.; Marrella, A.; and Patrizi, F. 2017. On the Disruptive Effectiveness of Automated Planning for LTL$_f$-Based Trace Alignment. In Singh, S.; and Markovitch, S., eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 3555–3561. AAAI Press.

De Giacomo, G.; and Vardi, M. Y. 2013. Linear Temporal Logic and Linear Dynamic Logic on Finite Traces. In Rossi, F., ed., *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 854–860. IJCAI/AAAI.

De Giacomo, G.; and Vardi, M. Y. 2015. Synthesis for LTL and LDL on Finite Traces. In Yang, Q.; and Wooldridge, M. J., eds., *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 1558–1564. AAAI Press.

Fahland, D.; and van der Aalst, W. M. P. 2015. Model repair - aligning process models to reality. *Inf. Syst.*, 47: 220–243.

Gebser, M.; Kaminski, R.; Kaufmann, B.; and Schaub, T. 2017. Multi-shot ASP solving with clingo. *CoRR*, abs/1705.09811.

Helmert, M. 2006. The Fast Downward Planning System. *J. Artif. Intell. Res.*, 26: 191–246.

Pesic, M.; Schonenberg, H.; and van der Aalst, W. M. 2007. DECLARE: Full Support for Loosely-Structured Processes. In *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*, 287–287. ISSN: 1541-7719.

Polyvyanyy, A.; van der Aalst, W. M. P.; ter Hofstede, A. H. M.; and Wynn, M. T. 2017. Impact-Driven Process Model Repair. *ACM Trans. Softw. Eng. Methodol.*, 25(4): 28:1–28:60.

Räim, M.; Ciccio, C. D.; Maggi, F. M.; Mecella, M.; and Mendling, J. 2014. Log-Based Understanding of Business Processes through Temporal Logic Query Checking. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences - Confederated International Conferences: CoopIS, and ODBASE 2014, Amantea, Italy, October 27-31, 2014, Proceedings*, 75–92.

Skydanienko, V.; Francescomarino, C. D.; Ghidini, C.; and Maggi, F. M. 2018. A Tool for Generating Event Logs from Multi-Perspective Declare Models. In van der Aalst, W. M. P.; Casati, F.; Kumar, A.; Mendling, J.; Nepal, S.; Pentland, B. T.; and Weber, B., eds., *Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018 co-located with 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, September 9-14, 2018*, volume 2196 of *CEUR Workshop Proceedings*, 111–115. CEUR-WS.org, Raffaele Conforti and Massimiliano de Leoni and Marlon Dumas.

Van der Aalst, W. M. 1998. The application of Petri nets to workflow management. *Journal of circuits, systems, and computers*, 8(01): 21–66.

van der Aalst, W. M. P. 2010. Process Discovery: Capturing the Invisible. *IEEE Comput. Intell. Mag.*, 5(1): 28–41.

van der Aalst, W. M. P.; Pesic, M.; and Schonenberg, H. 2009. Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development*, 23(2): 99–113.

White, S. A. 2004. Introduction to BPMN. *Ibm Cooperation*, 2(0): 0.