Dipartimento di Ingegneria informatica automatica e gestionale Antonio Ruberti

Sapienza Università di Roma

ASP-Based Declarative Process Mining

Francesco Chiariello [1] Fabrizio Maria Maggi [2] Fabio Patrizi [1]

[1]DIAG - Sapienza University of Rome [2]KRDB - Free University of Bozen-Bolzano

Freie Universität Bozen
Libera Università di Bolzano
Università Liedia de Bulsan

Objectives

We put forward Answer Set Programming (ASP) as an approach for modeling and solving problems from Declarative Process Mining (DPM)

- Three problems considered:
  - Log Generation
  - Conformance Checking
  - Query Checking
- Both *control-flow* and *data-aware* perspective

Problems

- Log generation [1]: generate a set of traces with given length compliant with a process model
- Conformance Checking [2]: check whether the traces of a log are compliant with a process model
- Query Checking [3]: compute activities that make a process satisfy a given property wrt to event log

Perspectives

- Control-flow: traces are sequences of activities
- Data-aware: traces are sequences of events (activities + data)

LTL_f with Local Conditions

- Used to formalize process models
- Syntax:

  $\varphi = true \mid A \mid a \odot a' \mid a \odot v \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U} \varphi$,

  A: activity, a: attribute, v: value for a.
- L-LTL_f extends LTL_f to express attribute conditions (comparing with attribute values and constants)
- Every L-LTL_f formula φ admits an automaton accepting exactly the traces satisfying φ [4].

Solution Technique

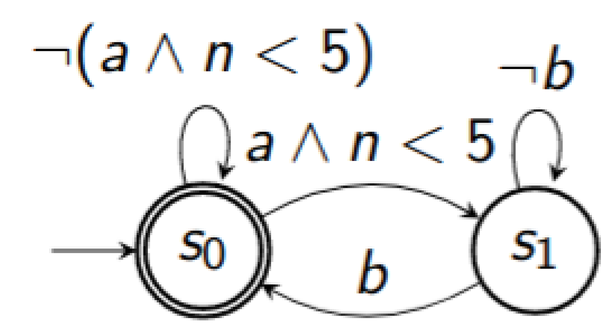
- Convert L-LTL_f formulas into automata
- Represent automata and traces in ASP
- Model how an automaton reads a trace
- Add problem-specific generation and test rules

Example

$$\varphi' = \mathbf{G}((a \wedge n < 5) \rightarrow \mathbf{F} b)$$

(Whenever activity a happens with a value less than 5 for its attribute n then eventually b happens)

Corresponding automaton:



Expressed in ASP as:

$initial(s_0)$.
$accepting(s_0)$.
$trans(s_1, 1, s_1)$.
$hold(1, T) \leftarrow trace(a, T), has\_value(n, V, T), V < 5$.
$trans(s_1, 2, s_0)$.
$hold(2, T) \leftarrow trace(b, T)$.
$trans(s_0, 3, s_0)$.
$hold(3, T) \leftarrow \texttt{not}\ hold(1, T), time(T)$.
$trans(s_1, 4, s_1)$.
$hold(4, T) \leftarrow trace(A, T), A \neq b$.

Log Generation

To generate possible answer sets/traces, add rules requiring that

- only one activity is true at a time
- all activity attributes are assigned a value

For testing check whether the automaton ends in an accepting state.

Conformance Checking

Traces are given. Check whether each of them is accepted.

Query Checking

- Given:
  - a set of traces
  - an L-LTL_f formula with activity variables
- Find the activities that assigned to the variables make the formula satisfied by the traces
- Solution concept: generate answer sets/formulas assigning activities to variables and test whether they are satisfied by the traces

Results

We show that our approach:

- outperforms SoA tool MP-Declare Log Generator [1] based on SAT, which does not exploit automata-based representation of specifications
- has slightly worst results than the SoA conformance checking tool Declare Analyzer [2], not based on declarative approach
- is the first one for data-aware query checking (show its feasibility)

For comparison, experiments focus on DECLARE [5] but approach able to handle general LTL_f specifications.

Conclusions

We have provided:

- ASP encodings of three DPM problems
- Performance evaluation
- Comparision wrt SoA

Put forward ASP as an effective modeling paradigm for DPM

Shown how to represent process models and event logs in ASP and how to easily encode various problems.

References

[1] Vasyl Skydanienko, Chiara Di Francescomarino, Chiara Ghidini, and Fabrizio Maria Maggi. A tool for generating event logs from multi-perspective declare models. In *Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018*, 2018.

[2] Andrea Burattin, Fabrizio Maria Maggi, and Alessandro Sperduti. Conformance checking based on multi-perspective declarative process models. *Expert Syst. Appl.*, 65, 2016.

[3] Margus Räim, Claudio Di Ciccio, Fabrizio Maria Maggi, Massimo Mecella, and Jan Mendling. Log-based understanding of business processes through temporal logic query checking. In *On the Move to Meaningful Internet Systems: OTM 2014 Conferences*, pages 75–92, 2014.

[4] Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence*. IJCAI/AAAI, 2013.

[5] Wil M. P. van der Aalst, Maja Pesic, and Helen Schonenberg. Declarative workflows: Balancing between flexibility and support. *Comput. Sci. Res. Dev.*, 23(2):99–113, 2009.

Acknowledgements

Work partly supported by the ERC Advanced Grant WhiteMech (No. 834228), the EU ICT-48 2020 project TAILOR (No. 952215), and the Sapienza Project DRAPE: Data- awaRe Automatic Process Execution.

ERC Advanced Grant WhiteMech: White-box Self Programming Mechanisms

TAILOR