Light Methods for Conformance Checking of Business Processes

Farbod Taymouri

Universitat Politècnica de Catalunya, Barcelona (Spain)
taymouri@cs.upc.edu

In Process Mining, Conformance Checking of business process models ad- dresses the deviations happened in their real executions. Conformance checking is a very important dimension on its own, enabling the detection and visualization of deviations on top of process models. Its consequences expand not only to detection of misalignment between observed and modeled behavior, but also to improve the value of process models by projecting event data on them. For example, model repair techniques open the door for enhancement of processes, by modifying those parts of the process model that no longer represent the underlying process.

Identifying deviations mathematically boils down to the notion of alignment conceptually. An alignment quantifies to what degree a process model can imitate what happened in its observed behavior, i.e., an event log. Formally, an alignment is a matrix data structure with two rows, where each column represents whether an observed event can be lined up with the corresponding activity of the given process model. With that said, an optimal alignment is the best combination by which the process model can imitate the corresponding observed behavior. Unfortunately, state of the art techniques for alignment computation have exponential time and space complexity, hindering their applicability for medium and large instances even for a single alignment. Additionally, extra challenges will appear if other dimensions, like multi-perspective or online computation, are considered.

This thesis presents several approaches for alignment computation, not only to tackle the mentioned challenges, but also, to provide new insights and facilities for conformance checking of large process models. Indeed, each of the proposed approaches has a specific set of advantages, and globally are complementary to each other; for instance, we propose techniques to handle very large process models in limited amount of memory or time. In general, they have some goals in common, namely, proposing light and efficient methods for alignment computation. Above that, one of the presented contributions for alignment computation is an abstraction framework that can be used in combination with any other techniques in the literature. Doing this not only alleviates the computational challenge, but also provides a greater logical perspective of deviations.

All methods presented in this thesis are kinds of combinatorial optimizations for alignment computation. They make close approximation to an optimal solution. The corresponding contributions represent novel ways in which they are proposed and their effectiveness for the mentioned challenge with respect to state of the art approach in different perspectives. Generally speaking, methods presented in this thesis can be categorized as:

– **Classical approaches [5, 8]:** These techniques exploit Integer Linear Programming (ILP) as well as structural theory of Petri nets to formulate alignment computation as an optimization of a set of linear equations. In other words, these approaches translate the computation of an optimal alignment to an ILP instance. In general, providing an alignment over speci

  ed intervals, called *Approximate Alignment*, is an outcome of these approaches. Three different formulations with various objectives are pro- posed. In short, the

  first one is optimal in terms of Parikh vector but it does not scale [1]; the second technique follows a recursive approach, hence it is fast but its optimality is not guaranteed [1]. The last one computes an alignment in an incremental way, adapting the A^* technique from [2], and a real solution (perhaps not optimal) is guaranteed [3].
– **Heuristic approaches [6, 7]:** These techniques adopt different policy with respect to the previous approaches, and take advantages of heuristic functions to explore the search space of alignments to find the optimal one(s). This can be done by obtaining an initial solution, and iteratively improving it until saturation or by reaching a certain criterion. In more details, the iterations must be light so that the complexity of the overall framework is not worsen, and more importantly, to guide the algorithm toward a better solution. The first technique in this class, called Local Search, poses an initial solution. The initial solution can be located somewhere in the corresponding search space. Then, tailored operators will be applied so as to improve the initial solution iteratively. Intuitively, the operators cause to examine neighbors of the current solution, and if a better solution is found, it will be considered as the current solution. The improvement loop continues until no better solution can be obtained [4]. The other technique in this class adopts a Genetic Algorithm, with well specific designed operators, by which exploration of the corresponding search space can be guided toward the best solution(s). The main idea behind this technique is similar to the previous one, i.e., exploring the search space until finding the best solution. Although this approach is slower than the previous heuristic approach, it can generate more than one solution, and moreover due to the evolutionary setting, the probability of getting stuck in local optimum is much smaller [5].
– **Model reduction [4]:** Regardless of the two main approaches just mentioned, one way to boost the effectiveness of alignment computation is reducing model and observed behavior without loosing alignment information. This structure reduction not only boosts the alignment computation, but also provides a big picture of detected deviations. Above that, a divide-and-conquer strategy will be provided for the first approach such that it breaks the original problem into a set of smaller independent problems which can be solved independently. This technique in more details is a framework for the size reduction of a given problem, i.e., model and event log. In other words, this approach can be integrated by any approaches which compute alignments to boost its efficiency in different perspectives [6].

Experiments witness the merit of proposed approaches with respect to state of the art techniques in different perspectives, such as memory consumption, execution time, quality and accuracy of solutions found. All methods have been implemented as a stand-

alone tool box called ALI [1] [7]. The tool box and datasets used in this thesis are publicly available at https://www.cs.upc.edu/ taymouri/.

References

1. F. Taymouri, J. Carmona, A recursive paradigm for aligning observed behavior of large structured process models, in: 14th International Conference of Business Process Management (BPM), Rio de Janeiro, Brazil, September 18 - 22, 2016.
2. A. Adriansyah, Aligning observed and modeled behavior, Ph.D. thesis, Technische Universiteit Eindhoven (2014).
3. B. van Dongen, J. Carmona, Th. Chatain, F. Taymouri, Aligning modeled and observed behavior: A compromise between complexity and quality, in: E. Dubois, K. Pohl (Eds.), Proceedings of the 29th International Conference on Advanced Information Systems Engineering (CAiSE'17), Vol. 10253 of Lecture Notes in Computer Science, Springer, Essen, Germany, 2017, to appear.
4. F. Taymouri, J. Carmona, Computing alignments of well-formed process models using local search, Submitted to ACM Transactions on Software Engineering and Methodology.
5. F. Taymouri, J. Carmona, An evolutionary technique to approximate multiple optimal alignments, in: 16th International Conference of Business Process Management (BPM), Sydney, Australia. September 9-14, Brazil, September 18 - 22, 2018.
6. F. Taymouri, J. Carmona, Model and event log reductions to boost the computation of alignments, in: Data-Driven Process Discovery and Analysis - 6th IFIP WG 2.6 International Symposium, SIMPDA 2016, Graz, Austria, December 15-16, 2016, Revised Selected Papers, 2016, pp. 1–21.
7. F. Taymouri, ALI: Alignment for Large Instances (2017).
   URL `https://www.cs.upc.edu/~taymouri/tool.html`

[1] 1Alignment of Large Instances