Applications of Extended Prefix Automata in Process Mining

Maxim Vidgof[1,*]

[1]Vienna University of Economics and Business, Welthandelsplatz 1, 1020, Vienna, Austria

Abstract
Process mining is a research area focusing on extracting insights from process execution data called event logs. Different areas of process mining have developed own methods and algorithms suited for specific purposes. In this PhD project, we are developing Extended Prefix Automata – a behavioral model serving as unified representation of a business process – and study its applicability in various areas of process mining. We analyze the applicability of Extended Prefix Automata for measuring process complexity, for predictive process monitoring and automated process discovery. In this paper, we report the current state and outline future plans for each of these applications.

Keywords
Process mining, Extended prefix automata, Process complexity, Predictive process monitoring, Process discovery

1. Introduction

Process mining is a research area focusing on the design of techniques that can automatically provide insights into business processes by analyzing historic process execution data, referred to as event logs [1, 2]. Over the years, different areas of process mining have developed different tools and algorithms for specific purposes.

This PhD project started with a study of connection between process complexity and quality of discovered process models [3]. In this study, we developed a new process complexity metric that would overcome the limitations of the existing ones by combining different complexity aspects. Extended Prefix Automata were developed merely as a process representation that was used to compute this metric. As this study was finished, however, it has not only been proven that the newly developed complexity metric correlates with model quality metrics and can be used to predict them, but two further research directions emerged. First, it was proposed that the complexity metrics can be used to predict other process characteristics, such as process performance or accuracy of predictive process monitoring. Second, Extended Prefix Automata turned out to be a powerful and extensible technique, which made it reasonable to apply it in other areas of process mining.

Thus in this PhD project, the applicability of Extended Prefix Automata to various areas of process mining is studied. This project follows the framework for algorithm engineering

BPM'22: Doctoral Consortium, September 11–16, 2022, Münster, Germany
maxim.vidgof@wu.ac.at (M. Vidgof)
0000-0003-2394-2247 (M. Vidgof)

proposed in [4].

This paper is structured as follows. Section 2 introduces Extended Prefix Automata. Then, Section 3 reviews the applications related to measuring process complexity and reports the results. Sections 4 and 5 outline the proposed applications in predictive process monitoring and automated process discovery, respectively. Finally, Section 6 concludes the paper.

2. Extended Prefix Automata

The concept of a prefix automaton was introduced and formalized in [5] based on the transition systems described in [6]. [6] introduced an approach to building transition systems from event logs using 3 parameters: decision of a state (*past*, *future* or *both*), representation of the information (*sequence*, *set* and *multiset*), and horizon (*limited* or *infinite*). Prefix automata use *past,sequence* and *infinite* settings since it allows to derive a transition system with the same language as the respective event log, i.e. a behavioral representation of the log [5]. Thus, a prefix automaton is a transition system with the states corresponding to the states of the event log and transitions corresponding to the activities. In [6], transition systems were used for abstraction in order to balance between overfitting and underfitting in process discovery. In [5], prefix automata were used for conformance checking.

Based on these concepts, we introduced Extended Prefix Automata (or EPAs) [3]. An example of an EPA is shown in Figure 1. Same as the prefix automata, EPAs describe event logs they are constructed from without loss of information and abstraction. However, we introduce two important extensions. First, we map each state of an EPA to a set of events in the input event log having the same prefix, so that every event in the log corresponds to exactly one state. Second, we introduce a partitioning function that splits an EPA into $0 \leq k \leq |L|$ partitions, where $|L|$ is the number of traces in the log.

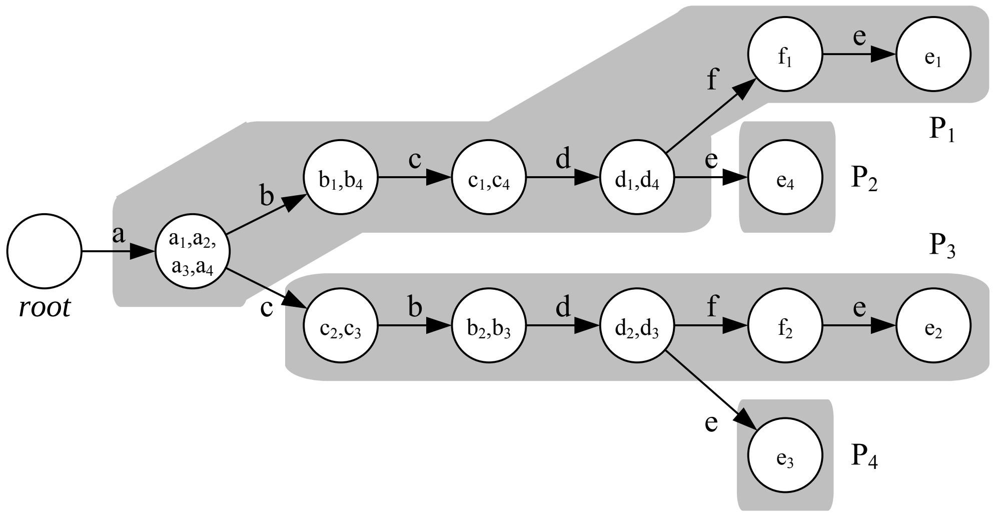

Figure 1: Extended prefix automaton with partitions derived from the event log $\langle a, b, c, d, f, e\rangle, \langle a, c, b, d, f, e\rangle, \langle a, c, b, d, e\rangle, \langle a, b, c, d, e\rangle$, reused from [3]

As will be shown in the next sections, these extensions are helpful in various regards. The partitioning function, for instance, is essential for computing complexity metrics using EPAs. The mapping between the states of EPA and events is not only helpful for complexity metrics as well, but also generally adds more transparency as for each state one can directly observe which events are responsible for the very existence of this state and use this information on the state level. Finally, it is also important to mention that the EPA is defined in such way that further extensions are also possible. This feature is extremely useful for enabling various applications, probably even beyond those covered in this PhD project.

3. Process Complexity

Research on process mining algorithms, including process discovery, has up to present moment taken the perspective that better algorithms would always yield better results [3]. However, as seen in data mining, this assumption does not necessarily hold. The characteristics of the input data can also have influence on the output.

In [3], we studied which process complexity metrics correlate with the quality of models discovered by process mining algorithms. However, before doing so, we realized that existing complexity metrics always focus on one of the three complexity aspects: *size*—how many events, traces, etc. the event log contains, *variation*—how many variants of executing the process the event log contains, and *distance*—how much these variants differ. We further observed that if two event logs are similar in one complexity dimension, e.g. *size* but different on the other dimensions, *size*-based complexity metrics will not capture these differences. Same applies to other categories of complexity metrics, i.e. *variation*-based metrics will not capture differences in *size*, etc. Thus, we first needed to construct a complexity metric that would capture all the three dimensions simultaneously.

We used Extended Prefix Automata and graph entropy to develop our complexity metrics. On a side note, EPAs were in fact initially developed in order to enable such metrics. First, we introduced *variant entropy* by directly applying the idea of graph entropy to EPAs. To calculate it, we use the number of states in each partition. We also introduced *normalized variant entropy* having the range from 0 to 1. However, variant entropy did not take the events, which were one of the main extensions in the EPA, into account, and thus did non suffice to capture all complexity dimensions, especially size. Thus, we also introduced *sequence entropy*. The main difference between sequence entropy and variant entropy is that in sequence entropy we are interested not in the number of states but in the number of events corresponding to states in each partition. Finally, we introduced *normalized sequence entropy*, which is also in range $[0, 1]$. It is especially helpful if one wants to compare structural complexity of event logs that differ in support or magnitude.

We have seen that complexity of the event logs correlates with the quality metrics of the models discovered from these logs, namely *precision*, *fitness*, *F-score* as well as *size* and *control flow complexity*. Out of 20 complexity metrics used in our study only 6 correlate highly with most model quality metrics, including our newly developed *normalized sequence entropy*.

We have also observed that complexity metrics can be used to suggest the most suitable process discovery algorithm since different complexity metrics have different impact on different process discovery algorithms, e.g. the quality of models produced by Evolutionary Tree Miner [7] and Inductive Miner [8] mainly depend on average edit distance [9], while quality of Split Miner [10] models depends on normalized sequence entropy.

Process complexity, however, is not only influencing model quality. We have also conducted a study [11] of how process complexity influences process performance. For this, we took multiple event logs, split them into time periods and observed the development of process complexity and performance over the timeframe captured in the event logs. As expected, we found a dependence between normalized sequence entropy and throughout time. These results have strong implications for research and practice. Previously, process complexity was evaluated based on interviews with process stakeholders – an approach that does not scale. Our technique allows for automated computation of process complexity based on process mining. In addition, our metric is objective and not perceptual as the interview-based ones.

It is also expected that process complexity will influence the accuracy of predictive process monitoring algorithms. In the best case, we may be able to recommend an algorithm based on complexity. We have also incorporated the concept of forgetting in our sequence entropy metrics, such that more recent events receive more weight and add more to complexity than the older ones. This study is, however, in its initial phase yet.

4. Predictive Process Monitoring

While initially developed to measure process complexity, EPAs have a number of features that make it promising to apply them in other process mining areas. Among such features is the extensibility of EPAs as well as the fact that EPAs represent the event logs without loss of information or abstraction.

Predictive process monitoring (PPM) is a technique that continuously provides information (next activity, outcome, risk, remaining time, etc.) about the future of ongoing traces [12, 13]. [14] shows that prefix-based trace clustering combined with classification can accurately predict whether some predicate will be fulfilled upon completion of a running trace. Another popular technique in predictive process monitoring are recurring neural networks, especially Long Short-Term Memory (LSTM) networks [15]. The latter approach, however, also brings the common disadvantages of neural networks: long training times and/or high computational power requirements as well as black-box nature.

The basic idea of a PPM technique proposed in this project is to attach a separate *predictor* to every state of the Extended Prefix Automaton. A predictor is a machine learning algorithm with unified interface. Such setting, first of all, allows for flexibility in choosing the machine learning algorithm (decision tree, nearest-neighbours, regression, neural networks), especially allowing to have different types of algorithms used in different states of the EPA. This means that in some states with simpler choices simpler algorithms may be used to save up computational power for states what require more complex algorithms. Moreover, it is possible that different algorithms of comparable complexity still perform differently for different traces, also making it more lucrative to use different algorithms for such different cases instead of applying just one to all traces. To summarize, EPA can act as a framework that allows to transparently use multiple predictive algorithms.

This is, however, not the only potential advantage of PPM on EPAs. Such framework uses multiple ML models instead of one, but these models are essentially simpler and in addition can be trained separately. This may reduce the time required for training, especially for re-training with additional data, as it is very probable that not all models will be affected. In the best case, this can even allow for efficient incremental training where each newly observed event can be simultaneously incorporated into the predictive model instead of training in batches.

Finally, EPAs are transparent by design as they show which actual events in the input log correspond to an EPA state. Combined with simpler ML models like kNN or decision trees, this can lead to a predictive process monitoring algorithm that is explainable by design as opposed to black-box neural networks.

It is expected that EPAs will primarily be used for next activity and suffix prediction. However, given the flexibility of the approach, by merely changing the predictors one can apply such framework for other types of predictions, e.g. categorial outcome or even remaining time predictions.

Preliminary evaluation on some of the BPIC logs has shown that even the most simple algorithm ZeroR that always predicts the majority class can yield 13 to 68% accuracy in next activity prediction depending on the event log. While this result is by no means impressive, it provides a good baseline. It is expected that using slightly more complex but still white-box ML algorithms can increase the accuracy without sacrificing explainability.

5. Automated Process Discovery

To close the circle, EPAs also have potential to be used in process discovery. Automated process discovery is an area of process mining aiming to produce business process models from event logs [1]. [6] already showed how transition systems similar to EPAs can be transformed into Petri Nets. Thus, adapting this method for actual EPAs is trivial.

However, without any additional modifications EPAs and process models constructed from them are merely enumerations of observed traces, meaning excellent precision and fitness, however no generalization and oftentimes also low readability. [6] used ”massaging” the transition system and abstraction to overcome the latter two issues. [16] also highlights the necessity of abstraction and proposes to use sets for state representation instead of sequences and also to consider suffixes for automaton construction instead of prefixes.

While these proposals will almost certainly improve process discovery, it must be noted, however, that they destroy some crucial properties of EPAs. Thus these techniques should only be applied as the last step and only for process discovery, and automata obtained in such way can only be seen as an intermediary step towards a process modes but cannot be used for other purposes described in the previous sections.

Nevertheless, for process discovery, we plan to incorporate some of these techniques in their exact or modified form as well as some additional steps not mentioned before. In addition, since EPAs also include actual events from the input event log, frequency-based filtering of branches can be incorporated.

Preliminary evaluation on PDC 2021[1] data showed that the algorithm gives perfect (1.0)

[1]https://icpmconference.org/2021/process-discovery-contest/

negative accuracy, i.e. classifies all not-fitting traces correctly, but extremely low (0.01) positive accuracy, i.e. it does not recognize most of the conforming traces. This effectively means it rejects almost all test traces. Given that at current stage EPAs in their unmodified state were used, such results are not surprising. However, as already mentioned, it is relatively simple to add e.g. loop recognition, parallel execution recognition or exclusive choice recognition before producing a final Petri Net. Also more advanced techniques such as automated frequency-based or statistical significance-based filtering may increase the performance even further. With these and probably even further modifications, process discovery with EPAs theoretically can perform at least better than the alpha-algorithm. The upper boundaries of this method are yet to be discovered.

6. Conclusion

In this paper, we have reported the current state and outlined the future plans of the proposed PhD project. We introduced Extended Prefix Automata (EPAs) and gave an overview of process mining areas where EPAs can be applied. We reported the first results of applications related to measuring process complexity as well as outlined the planned applications in predictive process monitoring and automated process discovery.

References

[1] W. M. van der Aalst, Process Mining: Data Science in Action, Second Edition, Springer, 2016.

[2] M. Dumas, M. L. Rosa, J. Mendling, H. A. Reijers, Fundamentals of Business Process Management, Second Edition, Springer, 2018.

[3] A. Augusto, J. Mendling, M. Vidgof, B. Wurm, The connection between process complexity of event sequences and models discovered by process mining, CoRR abs/2106.07990 (2021). URL: https://arxiv.org/abs/2106.07990. arXiv:2106.07990.

[4] J. Mendling, B. Depaire, H. Leopold, Theory and practice of algorithm engineering, CoRR abs/2107.10675 (2021). URL: https://arxiv.org/abs/2107.10675. arXiv:2107.10675.

[5] J. Munoz-Gama, J. Carmona, A fresh look at precision in process conformance, in: R. Hull, J. Mendling, S. Tai (Eds.), Business Process Management - 8th International Conference, BPM 2010, Hoboken, NJ, USA, September 13-16, 2010. Proceedings, volume 6336 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 211–226.

[6] W. M. P. van der Aalst, V. A. Rubin, H. M. W. Verbeek, B. F. van Dongen, E. Kindler, C. W. Günther, Process mining: a two-step approach to balance between underfitting and overfitting, Software and Systems Modeling 9 (2010) 87–111.

[7] J. Buijs, B. van Dongen, W. van der Aalst, On the role of fitness, precision, generalization and simplicity in process discovery, in: CoopIS, Springer, 2012.

[8] S. Leemans, D. Fahland, W. van der Aalst, Discovering block-structured process models from event logs containing infrequent behaviour, in: BPM Workshops, Springer, 2014.

[9] B. T. Pentland, Conceptualizing and measuring variety in the execution of organizational work processes, Management Science 49 (2003) 857–870.

[10] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, A. Polyvyanyy, Split miner: automated discovery of accurate and simple business process models from event logs, Knowledge and Information Systems (2018).

[11] M. Vidgof, B. Wurm, J. Mendling, A study of the connection between process complexity and performance based on digital trace data, 2022.

[12] F. M. Maggi, C. D. Francescomarino, M. Dumas, C. Ghidini, Predictive monitoring of business processes, in: M. Jarke, J. Mylopoulos, C. Quix, C. Rolland, Y. Manolopoulos, H. Mouratidis, J. Horkoff (Eds.), Advanced Information Systems Engineering - 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings, volume 8484 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 457–472. URL: https://doi.org/10.1007/978-3-319-07881-6_31. doi:10.1007/978-3-319-07881-6_31.

[13] C. D. Francescomarino, C. Ghidini, F. M. Maggi, F. Milani, Predictive process monitoring methods: Which one suits me best?, in: M. Weske, M. Montali, I. Weber, J. vom Brocke (Eds.), Business Process Management - 16th International Conference, BPM 2018, Sydney, NSW, Australia, September 9-14, 2018, Proceedings, volume 11080 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 462–479. URL: https://doi.org/10.1007/978-3-319-98648-7_27. doi:10.1007/978-3-319-98648-7_27.

[14] C. D. Francescomarino, M. Dumas, F. M. Maggi, I. Teinemaa, Clustering-based predictive process monitoring, IEEE Trans. Serv. Comput. 12 (2019) 896–909. URL: https://doi.org/10.1109/TSC.2016.2645153. doi:10.1109/TSC.2016.2645153.

[15] N. Tax, I. Verenich, M. L. Rosa, M. Dumas, Predictive business process monitoring with LSTM neural networks, in: E. Dubois, K. Pohl (Eds.), Advanced Information Systems Engineering - 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings, volume 10253 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 477–492. URL: https://doi.org/10.1007/978-3-319-59536-8_30. doi:10.1007/978-3-319-59536-8_30.

[16] A. Adriansyah, J. Munoz-Gama, J. Carmona, B. F. van Dongen, W. M. P. van der Aalst, Measuring precision of modeled behavior, Inf. Syst. E Bus. Manag. 13 (2015) 37–67. URL: https://doi.org/10.1007/s10257-014-0234-7. doi:10.1007/s10257-014-0234-7.