Online Parametric Timed Pattern Matching with Automata-Based Skipping

Masaki Waga
National Institute of Informatics and SOKENDAI
Tokyo, Japan

Étienne André
Université Paris 13, LIPN, CNRS, UMR 7030, Villetaneuse, France
National Institute of Informatics, Tokyo, Japan
JFLI, CNRS, Tokyo, Japan

***Keywords*-monitoring; real-time systems; parametric timed automata**

I. Background and problem setting

Monitoring real-time systems consists in deciding whether a log satisfies a specification. A problem of interest is to determine *for which segment* of the log the specification is satisfied or violated. This problem can be related to string matching and pattern matching. The *timed pattern matching problem* was formulated in [1], with subsequent works varying the setting and improving the technique (e. g., [2], [3], [4], [5]). The problem takes as input a log and a specification, and decides where in the log the specification is satisfied or violated. In [3], [5], we introduced a solution to the timed pattern matching problem where the log is given in the form of a timed word (a sequence of events with their associated timestamps), and the specification in the form of a timed automaton (TA), an extension of finite-state automata with clocks [6].

Example 1. Consider the automaton in Fig. 1a, and fix $p_1 = 1$ and $p_2 = 1$—which gives a timed automaton [6]. Here \$ is a special terminal character. For this timed automaton (say $\mathcal{A}$) and the target timed word w in Fig. 1b, the output of the timed pattern matching problem is the set of matching intervals $\{(t,t') \mid w|_{(t,t')} \in \mathcal{L}(\mathcal{A})\} = \{(t,t') \mid t \in [3.7, 3.9), t' \in (6.0, \infty)\}$.

While the log is by definition concrete, it may happen that the specification is subject to uncertainty. For example, we may want to detect *cyclic patterns* with a period d, without knowing the value of d with full uncertainty. Therefore, the more abstract problem of *parametric timed pattern matching* becomes of interest: **given a (concrete) timed log and an incomplete specification where some of the timing constants may be known with limited precision or completely unknown, what are the intervals and the valuations of the parameters for which the specification holds?**

This work is partially supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), by JSPS Grants-in-Aid No. 15KT0012 & 18J22498 and by the ANR national research program PACS (ANR-14-CE28-0002).

The results of this extended abstract will be presented at NFM 2019: 11th Annual NASA Formal Methods Symposium.

Masaki Waga is a JSPS Research Fellow.

Coming back to Fig. 1, the question becomes to exhibit values for t, t', p_1, p_2 for which the specification holds on the log, i. e., $\{(t,t',v) \mid w|_{(t,t')} \in \mathcal{L}(v(\mathcal{A}))\}$, where v denotes a valuation of p_1, p_2 and $v(\mathcal{A})$ denotes the replacement of p_1, p_2 in $\mathcal{A}$ with their respective valuation in v. In [7], we showed that this problem is decidable (mainly due to the finiteness of the logs), and we proposed an approach based on parametric timed model checking using timed words and parametric timed automata [8], implemented in the IMITATOR model checker.

II. Contribution

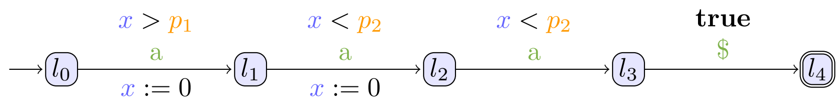
Our contribution is threefold. First, we propose a new *ad-hoc* technique for performing efficient parametric timed pattern matching. Second, we propose optimizations based on *skipping*, in the line of [5]. Third, we implement our framework, we perform a set of experiments on a set of automotive benchmarks, and show that we increase the efficiency compared to the state-of-the-art [7] by an order of magnitude. Our algorithm is suitable for online monitoring, as it does not need the whole run to be executed, and experiments show that it is fast enough to be applied at runtime.

III. Algorithms

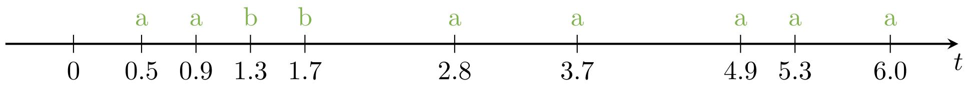
We propose three algorithm for parametric timed pattern matching, namely “no skip”, “non-parametric skip”, and “parametric skip”. See [9] for the detail of the algorithms.

No skip: Similarly to the online algorithm for timed pattern matching in [3], our first algorithm “no skip” finds all the matching triples $(t,t',v) \in \mathcal{M}(w,\mathcal{A})$ by a breadth-first search. Our algorithm is online in the following sense: after reading the i-th element (a_i, τ_i) of the timed word $w = (\overline{a}, \overline{\tau})$, it immediately outputs all the matching triples (t,t',v) over the available prefix $(a_1,\tau_1),(a_2,\tau_2),\dots,(a_i,\tau_i)$ of w.

Non-parametric skip: Our second algorithm “non-parametric skip” is an improvement of “no skip”. Following [5], we employ *FJS-style* skipping [10]. After reading the i-th element (a_i,τ_i) of the timed word $w = (\overline{a}, \overline{\tau})$, “non-parametric skip” does not immediately output the matching over the available prefix $(a_1,\tau_1),(a_2,\tau_2),\dots,(a_i,\tau_i)$ of w, but it still outputs the matching before obtaining the entire timed word with some delay.



(a) A parametric timed automaton



(b) A timed word

Figure 1: An example of parametric timed pattern matching [5]

Table I: Execution time for GEAR (above) and ONLYTIMING (below) in [9] [s]

$\lvert w \rvert$	No Skip	Non-Param. Skip	Param. Skip	IMITATOR
1467	0.04	0.05	0.05	1.781
2837	0.0725	0.0805	0.09	3.319
4595	0.124	0.13	0.1405	5.512
5839	0.1585	0.156	0.17	7.132
7301	0.201	0.193	0.2115	8.909
8995	0.241	0.2315	0.2505	10.768
10315	0.2815	0.269	0.2875	12.778
11831	0.322	0.301	0.325	14.724
13183	0.3505	0.3245	0.353	16.453
14657	0.392	0.361	0.395	18.319

$\lvert w \rvert$	No Skip	Non-Param. Skip	Param. Skip	IMITATOR
1000	0.0995	0.1305	0.11	1.690
2000	0.191	0.23	0.191	3.518
3000	0.2905	0.3265	0.273	5.499
4000	0.3905	0.426	0.3525	7.396
5000	0.488	0.5225	0.4325	9.123
6000	0.588	0.6235	0.517	11.005

Parametric skip: Our third algorithm "parametric skip" is also an improvement of "no skip" by FJS-style skipping. In "parametric skip", we also use the synthesized parameters in the latest matching trial for a larger skipping, while it is ignored in "non-parametric skip".

IV. EXPERIMENTS

We implemented our three algorithms for parametric timed pattern matching in C++ and we compiled them using GCC 7.3.0. As a baseline, we used our previous implementation of parametric timed pattern matching based on IMITATOR [7] (version 2.10.4). See [9] for the detail of the experiment setting and the benchmarks. Experiment data can be found on https://github.com/MasWag/monaa/blob/PTPM/doc/NFM2019.md.

In Table I, we observe that our algorithms are faster than the IMITATOR-based implementation [7] by orders of magnitude. This is mainly because our implementation is specific to parametric timed pattern matching while IMITATOR is a general tool for parametric verification. This shows the much better efficiency of our new approach compared to [7]. We also observe that either "parametric skip" or "non-parametric skip" is the fastest. The preference between two skipping methods depends on the pattern PTA, but it seems "parametric skip" is a better option because even if it is slower, the overhead is not very large.

Natural future works include more expressive specifications than (parametric) timed automata-based specifications, e. g., using more expressive logics such as [11].

REFERENCES

[1] D. Ulus, T. Ferrère, E. Asarin, and O. Maler, "Timed pattern matching," in *FORMATS*, ser. LNCS, A. Legay and M. Bozga, Eds., vol. 8711. Springer, 2014, pp. 222–236.

[2] ——, "Online timed pattern matching using derivatives," in *TACAS*, ser. LNCS, M. Chechik and J. Raskin, Eds., vol. 9636. Springer, 2016, pp. 736–751.

[3] M. Waga, T. Akazaki, and I. Hasuo, "A Boyer-Moore type algorithm for timed pattern matching," in *FORMATS*, ser. LNCS, M. Fränzle and N. Markey, Eds., vol. 9884. Springer, 2016, pp. 121–139.

[4] E. Asarin, O. Maler, D. Nickovic, and D. Ulus, "Combining the temporal and epistemic dimensions for MTL monitoring," in *FORMATS*, ser. LNCS, A. Abate and G. Geeraerts, Eds., vol. 10419. Springer, 2017, pp. 207–223.

[5] M. Waga, I. Hasuo, and K. Suenaga, "Efficient online timed pattern matching by automata-based skipping," in *FORMATS*, ser. LNCS, A. Abate and G. Geeraerts, Eds., vol. 10419. Springer, 2017, pp. 224–243.

[6] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, Apr. 1994.

[7] É. André, I. Hasuo, and M. Waga, "Offline timed pattern matching under uncertainty," in *ICECCS*, A. W. Lin and J. Sun, Eds. IEEE CPS, 2018, pp. 10–20.

[8] R. Alur, T. A. Henzinger, and M. Y. Vardi, "Parametric real-time reasoning," in *STOC*, S. R. Kosaraju, D. S. Johnson, and A. Aggarwal, Eds. New York, NY, USA: ACM, 1993, pp. 592–601.

[9] M. Waga and É. André, "Online parametric timed pattern matching with automata-based skipping," 2019. [Online]. Available: http://arxiv.org/abs/1903.07328

[10] F. Franek, C. G. Jennings, and W. F. Smyth, "A simple fast hybrid pattern-matching algorithm," *Journal of Discrete Algorithms*, vol. 5, no. 4, pp. 682–695, 2007.

[11] D. A. Basin, F. Klaedtke, S. Müller, and E. Zalinescu, "Monitoring metric first-order temporal properties," *Journal of the ACM*, vol. 62, no. 2, pp. 15:1–15:45, 2015.