Automatic Compositional Verification of Timed Systems

Shang-Wei Lin[1], Yang Liu[1], Jun Sun[2], Jin Song Dong[1], and Étienne André[3]

[1] National University of Singapore
{tsllsw,tslliuya,dongjs.comp}@nus.edu.sg
[2] Singapore University of Technology and Design
sunjun@sutd.edu.sg
[3] LIPN, CNRS UMR 7030, Université Paris 13, France
Etienne.Andre@lipn.univ-paris13.fr

Abstract. Specification and verification of real-time systems are important research topics with crucial applications; however, the so-called state space explosion problem often prevents model checking to be used in practice for large systems. In this work, we present a self-contained toolkit to analyze real-time systems specified using *event-recording automata* (ERAs), which supports system modeling, animated simulation and automatic compositional verification based on learning techniques. To the best of our knowledge, it is the first tool supporting fully automatic compositional verification for timed systems. Experimental results show that our tool outperforms the state-of-the-art timed model checkers.

1 Introduction

Ensuring the correctness of safety-critical systems with timing requirements is crucial and challenging. Model checking is emerging as an effective verification method and has been widely used for timed system. However, model checking suffers from the infamous state space explosion problem, and the problem is even graver in timed model checking because of the timed transitions.

To alleviate this problem, we proposed an automatic learning-based compositional verification framework for timed systems. We focus on timed systems that are modeled by *event-recording automata* (ERAs) [3], which is a determinizable class of timed automata. ERAs are as powerful as timed transition systems and are sufficiently expressive to model many interesting timed systems. The proposed framework consists of a compositional verification based on the non-circular assume-guarantee (AG-NC) proof rule and uses a learning algorithm, TL* [8], to automatically generate timed assumptions for *assume-guarantee reasoning* (AGR).

Our engineering efforts realize the proposed techniques into a self-contained toolkit for analyzing real-time systems, which is built as the ERA module (can be downloaded at [1]) in the PAT model checker [9]. Fig. 1 shows the architecture of our tool, which consists of four components, namely the *editor*, the *parser*, the *simulator* and *verifiers*. The editor is featured with a powerful graphic drawing component that allows users to design system models and specify properties by drawing ERAs. The editor also supports syntax highlighting, intellisense, and undo/redo functionality such that designers can efficiently model the systems. The parser compiles both the system models and

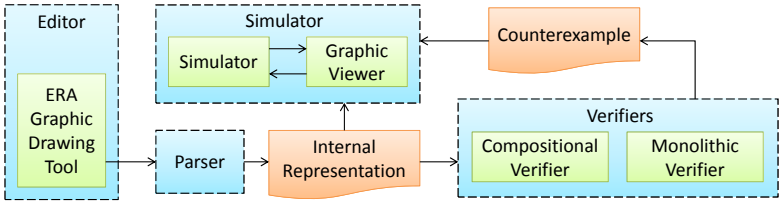
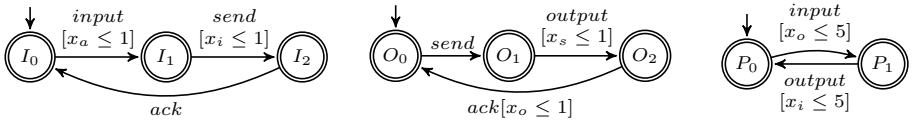


Fig. 1. Architecture of the ERA Module in PAT



(a) INPUT (b) OUTPUT (c) Property φ

Fig. 2. Models and property of the I/O system

the properties (in the form of ERAs) into internal representations for simulation and verification. The simulator allows users to perform various simulation tasks on the input model: complete state-space generation of selected model, automatic simulation via random executions, user interactive simulation, trace replay and so on. Most importantly, compositional verification is fully automated for properties specified using ERAs. To the best of our knowledge, our tool is the first one supporting fully automatic compositional verification for timed systems. Our tool also supports the traditional monolithic approach that generates the global state space based on zone abstraction. Users can choose to use either the monolithic or our compositional approach inside the verification interface. If the verification result is false, counterexamples will be produced and can be visualized using the simulator. Experimental results (Section 3) show that our tool of compositional verification for real-time systems outperforms traditional timed monolithic approaches in many cases.

2 Compositional Verification of ERAs

An *event-recording automaton* (ERA) is a special case of timed automaton where each event a on transitions is associated with a corresponding event-recording clock x_a recording the time elapsed since the last occurrence of event a. Each event-recording clock x_a is implicitly and automatically reset when a transition with event a is taken.

Fig. 2 gives an I/O system with two components, INPUT and OUTPUT, modeled by ERAs. The pairs of event-recording clocks and the corresponding events are $x_i : input$, $x_s : send$, $x_o : output$, and $x_a : ack$. The model of the INPUT component is shown in Fig. 2 (a). It performs an $input$ event within one time unit once it receives an ack event from OUTPUT. Subsequently, it performs a $send$ event to notify OUTPUT and waits for another ack event from OUTPUT. The model of OUTPUT is shown in Fig. 2 (b), which is similar to INPUT. The system property φ, as shown in Fig. 2 (c), is that $input$ and $output$ events should alternate and the time difference between every two con-

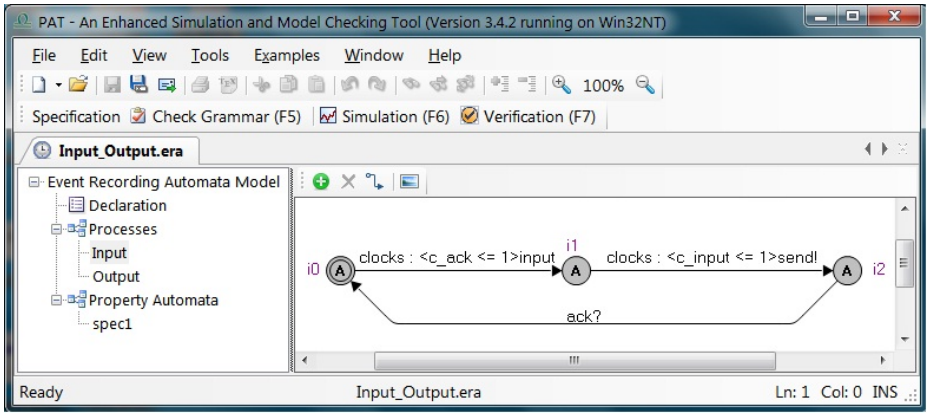


Fig. 3. GUI of the PAT Model Checker

secutive events should not exceed five time units. Fig. 3 shows the INPUT component modeled in PAT (the editor for drawing ERAs is heavily influenced by UPPAAL, one of the most popular timed automata model checkers), where a double circle represents the initial state and a state labeled with "**A**" represents an accepting state.

The flow of the proposed timed compositional verification is a two-phase process using the TL* algorithm [8] to automatically learn the timed assumption needed by AGR to prove or disprove the property. The first, *untimed verification*, phase constructs the untimed assumption, and then the second, *timed verification*, phase refines the untimed assumption into timed one and concludes the verification result (interested readers are referred to the technical report [2] for the detailed algorithm). In PAT, users can choose to use either the compositional or monolithic approach to verify the timed systems. After verification, PAT shows that the I/O system satisfies the property φ.

3 Experimental Results and Discussion

To show the feasibility and scalability of our tool, we present verification results of four different applications, namely the CSS, GSS, FMS, and AIP systems, in Table 1. The details of the four systems, their models, and the verified properties can be found in [1] or in Appendix. The experimental results were obtained by running PAT on a Windows 7 machine with a 2.27 GHz Intel(R) Core(TM) i3 processor and 4 GB RAM. We also compared our approach with the UPPAAL model checker; however, we do not list the verification time of UPPAAL for verifying the AIP system because UPPAAL does not support events on transitions such that the AIP system cannot be modeled in UPPAAL. When the system size is small, compositional approach does not outperform monolithic verification or UPPAAL because of the overhead of learning iterations; when the number of components increases, the learning iterations compensate for the large global state space and compositional approach can reduce the verification time and the memory usage significantly. For the FMS-4 system, the monolithic approach and UPPAAL cannot even finish the verification using 4 GB memory.

Table 1. Verification Results

				Monolithic				Compositional				UPPAAL
System	n	$\|C_\Sigma\|$	$\frac{\|P_{\not\models}\|}{\|P\|}$	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	Time (secs)
CSS	3	6	0/6	11	20	0.03	0.16	19	50	0.06	0.77	0.05
GSS	3	3	2/3	29	46	0.03	0.13	56	107	0.03	0.69	0.06
FMS-1	5	3	1/3	193	514	0.03	1.18	60	138	0.03	0.89	0.08
FMS-2	10	6	3/6	76,305	396,789	40.71	114.08	1,492	4,952	0.66	6.60	2.05
FMS-3	11	6	5/7	201,601	1,300,566	70.02	295.89	3,150	16,135	1.14	12.07	9.87
FMS-4	14	8	3/9	–	–	–	ROM	26,320	127,656	51.02	41.41	ROM
AIP	10	4	5/10	104,651	704,110	78.05	149.68	2,992	12,971	1.90	7.39	N/A

n: # of components; $|C_\Sigma|$: # of event-recording clocks; $|P|$: # of properties; $|P_{\not\models}|$: # of violated properties; $|L|_{max}$: # of visited locations during verification; $|\delta|_{max}$: # of visited transitions during verification; ROM: run out of memory

Discussion. AGR has been applied to model checking to alleviate the state space explosion problem [5]. However, the construction of the assumptions for AGR usually requires nontrivial creativity and experience, which limits the impact of AGR. Cobleigh et al. [6] proposed a framework that generates the assumptions of components automatically using the L* algorithm [4]. This work was a breakthrough of automating compositional verification for untimed systems. Grinchtein et al. [7] proposed three algorithms for learning ERAs; however, the time complexity of the algorithms depend exponentially on the largest constant appearing in the time constraints. In [8], we proposed a more efficient polynomial time algorithm, TL*, for learning ERAs. Starting from 2010, ERA module in PAT has come to a stable stage with solid testing. We successfully applied it to verify real-time systems ranging from classical concurrent algorithms to real world problems. In the future, we plan to use different techniques to generate the assumptions and to extend the framework using other proof rules of AGR.

References

1. https://sites.google.com/site/shangweilin/era-pat.
2. https://sites.google.com/site/shangweilin/technical-reports.
3. R. Alur, L. Fix, and T. A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theoretical Computer Science*, 211(1-2):253–273, 1999.
4. D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.
5. E. M. Clarke, D. E. Long, and M. K. L. Compositional model checking. In *LICS 1989*, pages 353–362, 1989.
6. J. M. Cobleigh, D. Giannakopoulou, and C. S. Păsăreanu. Learning assumptions for compositional verification. In *TACAS*, volume 2619 of *LNCS*, pages 331–346, 2003.
7. O. Grinchtein, B. Jonsson, and M. Leucker. Learning of event-recording automata. *Theorectical Computer Science*, 411(47):4029–4054, 2010.
8. S. W. Lin, E. André, J. S. Dong, J. Sun, and Y. Liu. An efficient algorithm for learning event-recording automata. In *ATVA*, volume 6996 of *LNCS*, pages 463–472, 2011.
9. J. Sun, Y. Liu, J. S. Dong, and J. Pang. PAT: Towards flexible verification under fairness. In *CAV*, volume 5643 of *LNCS*, pages 709–714, 2009.

A Details of the Experiments

To show the feasibility and scalability of our tool with compositional timed verification, we applied PAT with the ERA module on the following four applications.

- **CSS**. A client-server system consists of three components, namely one server and two clients. A resource is to be shared by the two clients in a mutually exclusive way. The server is responsible for synchronizing the use of the resource by the two clients. Three properties requiring that two clients cannot access the resource simultaneously and that each client can access the resource within two time units after its request are verified.
- **GSS**. A gas station system consists of five components: one operator, one queue, one pump, and two customers. Two customers can fill gas at this gas station. Three properties requiring that customers should be served in order and that each customer can start filling gas within three time units after his payment are verified.
- **FMS**. A flexible manufacturing system (FMS) produces blocks with a cylindrical painted pin from raw blocks and raw pegs. It consists of fourteen devices: three conveyors, two mills, a lathe, a painting device, six robots, and an assembly machine. The devices are connected through nine buffers, and the capacity of each buffer is one part. We modeled the FMS system in a constructive way such that four versions of models have been obtained, namely FMS-1 (the simplest one), FMS-2 (the medium one), FMS-3 (a complex one), and FMS-4 (the most complex one). Properties requiring that each buffer should not overflow or underflow and that output of each buffer should be within three time units after its input are verified.
- **AIP**. The AIP manufacturing system produces two products from two types of materials. It consists of ten components, namely an I/O station, three transport units, two assembly stations, three external loops, and a central loop. Properties requiring that the routes of the two types of materials should be opposite and that output of each loop should be within three time units after its input are verified.

The details of the four systems, their models, and the verified properties can be found in [1]. Tables 2 to 5 show the verification results of the four timed systems using the proposed approach and traditional monolithic timed model checking that constructs the timed global state space based on zone abstraction, respectively. For the FMS system, we only show the FMS-4 verification results here because FMS 1-3 are subsystems of FMS-4; however, they can still be found in [1]. Note that both the monolithic verification and our approach adopt on-the-fly technique, i.e., the verification generates a counterexample without constructing the whole state space if the property is violated. Thus, both monolithic verification and our compositional approach may find a counterexample quickly if the property is violated. When the number of components increases provided that the property is satisfied, our compositional approach outperform monolithic one significantly. For the FMS system, monolithic approach cannot even finish the verification process using 4G memory for the satisfied properties. We also compared our approach with the UPPAAL model checker; however, we do not list the verification time of UPPAAL for verifying the AIP system because UPPAAL does not support events on transitions such that the AIP system cannot be modeled in UPPAAL. We can also observe that our compositional approach outperforms UPPAAL.

Table 2. Verification Results of CCS

			Monolithic				Compositional				UPPAAL
Spec	$\|C_\Sigma\|$	Satisfied?	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	Time (secs)
1	6	YES	8	16	0.01	0.13	19	50	0.04	0.77	0.04
2	6	YES	11	20	0.01	0.16	12	21	0.01	0.47	0.04
3	6	YES	11	20	0.01	0.16	12	21	0.01	0.48	0.04
Total					0.03				0.06		0.12

Table 3. Verification Results of GSS

			Monolithic				Compositional				UPPAAL
Spec	$\|C_\Sigma\|$	Satisfied?	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	Time (secs)
1	3	NO	16	17	0.01	0.09	56	107	0.01	0.66	0.04
2	3	NO	17	19	0.01	0.11	56	107	0.01	0.69	0.05
3	3	YES	29	46	0.01	0.13	33	79	0.01	0.60	0.04
Total					0.03				0.03		0.13

Table 4. Verification Results of FMS-4

			Monolithic				Compositional				UPPAAL
Spec	$\|C_\Sigma\|$	Satisfied?	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	Time (secs)
1	8	YES	–	–	–	ROM	26,320	127,656	9.17	41.14	ROM
2	8	YES	–	–	–	ROM	26,320	127,656	9.13	41.04	ROM
3	8	NO	64	105	0.12	0.84	15	20	0.01	0.70	0.78
4	8	NO	100	182	0.03	1.33	24	35	0.01	0.68	ROM
5	8	YES	–	–	–	ROM	26,320	127,656	8.79	40.19	ROM
6	8	YES	–	–	–	ROM	26,320	127,656	9.03	41.00	ROM
7	8	YES	–	–	–	ROM	19,440	101,832	7.73	41.41	ROM
8	8	NO	118	244	0.02	0.31	24	35	0.01	0.52	0.06
9	8	YES	–	–	–	ROM	26,320	127,656	7.14	37.06	ROM
Total					N/A				51.02		N/A

Table 5. Verification Results of AIP

			Monolithic				Compositional				UPPAAL
Spec	$\|C_\Sigma\|$	Satisfied?	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	$\|L\|_{max}$	$\|\delta\|_{max}$	Time (secs)	Mem (MB)	Time (secs)
1	4	NO	137	382	0.02	0.25	224	709	0.02	1.69	–
2	4	NO	368	1,173	0.04	3.39	524	1,951	0.07	0.99	–
3	4	NO	651	2,207	0.07	4.39	524	1,951	0.09	2.32	–
4	4	NO	63	145	0.01	0.68	224	689	0.02	1.13	–
5	4	YES	104,651	704,110	17.19	149.68	2,745	12,861	0.34	5.86	–
6	4	YES	104,651	704,110	16.98	145.43	2,745	12,861	0.34	6.05	–
7	4	YES	86,051	562,682	13.41	120.28	2,271	10,267	0.26	6.14	–
8	4	YES	86,051	563,582	13.28	120.44	2,253	10,170	0.26	5.71	–
9	4	YES	104,651	704,110	17.04	145.73	2,922	12,971	0.49	7.39	–
10	4	NO	14	15	0.01	0.12	14	15	0.01	0.82	–
Total					78.05				1.90		N/A