Automatic Compositional Verification of Timed Systems (Tool Paper)

Shang-Wei LIN, Yang LIU, Jun SUN, Jin Song DONG, and Étienne André

Temasek Laboratories
National University of Singapore

August 29, 2012

Motivation

State Space Explosion Problem w.r.t. Model Checking

Assume-Guarantee Reasoning (AGR)

$$\frac{\begin{array}{rcl} M_1 \parallel A & \models & \varphi \\ M_2 & \models & A \end{array}}{M_1 \parallel M_2 \models \varphi}$$

How to construct the assumption A *automatically*?

- Untimed systems
  - J. M. Cobleigh, D. Giannakopoulou, and C. S. Păsăreanu. Learning assumptions for compositional verification. In TACAS, volume 2619 of LNCS, pp. 331–346, 2003.
- How about *timed systems*?

Outline

Event-Recording Automata (ERA)

The TL* Algorithm

Learning-Based Automatic Compositional Verification

Experiment Results

Conclusion and Future Work

Outline

Event-Recording Automata (ERA)

The TL* Algorithm

Learning-Based Automatic Compositional Verification

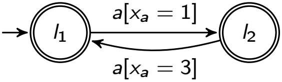
Experiment Results

Conclusion and Future Work

Event-Recording Automata (ERA)

The following ERA $\mathcal{A}_1$ accepts the timed language $U_T^{\mathcal{A}_1}$ of the form $(a, t_1)(a, t_2)(a, t_3)\cdots$ where $t_{2i} - t_{2i-1} = 3$ and $t_{2i+1} - t_{2i} = 1$ for all $i \geq 1$

- $(a, 1)(a, 4)$
- $(a, 1)(a, 4)(a, 5)(a, 8)(a, 9)$



$\mathcal{A}_1$

Outline

Event-Recording Automata (ERA)

The TL* Algorithm

Learning-Based Automatic Compositional Verification

Experiment Results

Conclusion and Future Work

The TL* Algorithm

The TL* algorithm is a *timed* extension of the L* algorithm.

The TL* algorithm is a formal method to learn a minimal event-recording automaton (ERA) that accepts an unknown timed language U_T over an alphabet Σ

- We use U to denote the untimed language of U_T

The TL* Algorithm (cont.)

The TL* algorithm has to interact with a Minimal Adequate *Teacher*

- *untimed membership query* Q_m
  - Is an untimed word in the unknown untimed language U?
- *untimed candidate query* Q_c
  - Does a DFA accept the unknown untimed language U?
- *timed membership query* Q_m^T
  - Is a guarded word in the unknown timed language U_T?
- *timed candidate query* Q_c^T
  - Does an ERA accept the unknown timed language U_T?

The TL* Algorithm (cont.)

The TL* algorithm consists of two phases

- *Untimed Learning* Phase
  - The L* algorithm is used to learn a DFA M accepting the untimed language U
- *Timed Refinement* Phase
  - The DFA M is refined into an event-recording automaton (ERA) by adding time constraints or locations

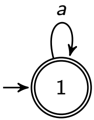
An Example

Suppose $U_T^{\mathcal{A}_1}$ is the timed language to be learned.

Untimed Learning Phase

	λ
λ	1 (s_0)
a	1

(a) T_1



(b) M_1

	λ
λ	1 (s_0)
$(a, true)$	1

(c) T_2

$$\mathcal{L}(M_1) = U^{\mathcal{A}_1} = a^*$$

An Example (cont.)

$Q_c^T(M_1) = 0$ with a negative counterexample $(a, x_a < 1)$

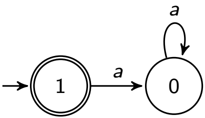
Timed Refinement 1

	λ
λ	$1\ (s_0)$
$(a, x_a < 1)$	0
$(a, x_a \geq 1)$	0

(a) T_3

	λ
λ	$1\ (s_0)$
$(a, x_a < 1)$	$0\ (s_1)$
$(a, x_a \geq 1)$	0
$(a, x_a < 1)(a, x_a < 1)$	0
$(a, x_a < 1)(a, x_a \geq 1)$	0

(b) T_4



(c) M_2

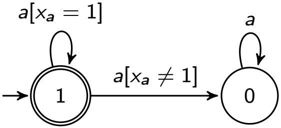
An Example (cont.)

$Q_c^T(M_2) = 0$ with a positive counterexample $(a, x_a = 1)$

Timed Refinement 2

	λ
λ	1 (s_0)
$(a, x_a < 1)$	0 (s_1)
$(a, x_a = 1)$	1
$(a, x_a > 1)$	0
$(a, x_a < 1)(a, x_a < 1)$	0
$(a, x_a < 1)(a, x_a = 1)$	0
$(a, x_a < 1)(a, x_a > 1)$	0

(a) T_5



(b) M_3

An Example (cont.)

$Q_c^T(M_3) = 0$ with a negative counterexample $(a, x_a = 1)(a, x_a = 1)$

A suffix $(a, x_a = 1)$ shows that λ and $(a, x_a = 1)$ should not be in the same class

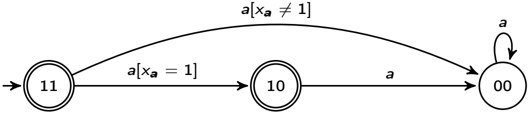
Timed Refinement 3

	λ	$(a, x_a = 1)$
λ	1	1 (s_0)
$(a, x_a < 1)$	0	0 (s_1)
$(a, x_a = 1)$	1	0
$(a, x_a > 1)$	0	0
$(a, x_a < 1)(a, x_a < 1)$	0	0
$(a, x_a < 1)(a, x_a = 1)$	0	0
$(a, x_a < 1)(a, x_a > 1)$	0	0

(a) T_6

	λ	$(a, x_a = 1)$
λ	1	1 (s_0)
$(a, x_a < 1)$	0	0 (s_1)
$(a, x_a = 1)$	1	0 (s_2)
$(a, x_a > 1)$	0	0
$(a, x_a < 1)(a, x_a < 1)$	0	0
$(a, x_a < 1)(a, x_a = 1)$	0	0
$(a, x_a < 1)(a, x_a > 1)$	0	0
$(a, x_a = 1)(a, x_a < 1)$	0	0
$(a, x_a = 1)(a, x_a = 1)$	0	0
$(a, x_a = 1)(a, x_a > 1)$	0	0

(b) T_7



(c) M_4

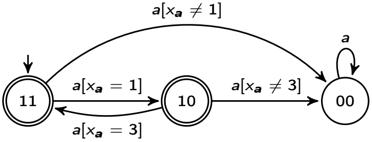
An Example (cont.)

$Q_c^T(M_4) = 0$ with a positive counterexample $(a, x_a = 1)(a, x_a = 3)$

Timed Refinement 4

	λ	$(a, x_a = 1)$
λ	1	1 (s_0)
$(a, x_a < 1)$	0	0 (s_1)
$(a, x_a = 1)$	1	0 (s_2)
$(a, 1 < x_a < 3)$	0	0
$(a, x_a = 3)$	0	0
$(a, x_a > 3)$	0	0
$(a, x_a < 1)(a, x_a < 1)$	0	0
$(a, x_a < 1)(a, x_a = 1)$	0	0
$(a, x_a < 1)(a, 1 < x_a < 3)$	0	0
$(a, x_a < 1)(a, x_a = 3)$	0	0
$(a, x_a < 1)(a, x_a > 3)$	0	0
$(a, x_a = 1)(a, x_a < 1)$	0	0
$(a, x_a = 1)(a, x_a = 1)$	0	0
$(a, x_a = 1)(a, 1 < x_a < 3)$	0	0
$(a, x_a = 1)(a, x_a = 3)$	1	1
$(a, x_a = 1)(a, x_a > 3)$	0	0

(a) T_8



(b) M_5

An Example (cont.)

$Q_c^T(M_5) = 1$, i.e., $\mathcal{L}(M_5) = U_T^{\mathcal{A}_1}$

The learning process of TL* is finished

Outline

Event-Recording Automata (ERA)

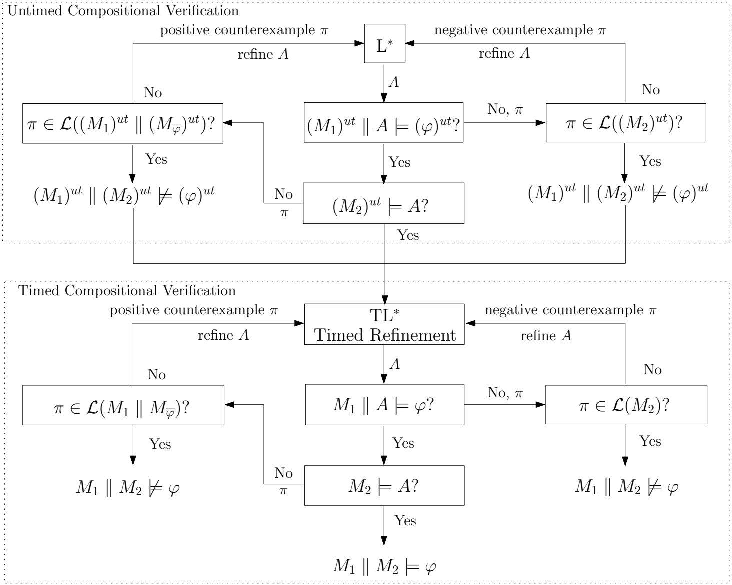
The TL* Algorithm

Learning-Based Automatic Compositional Verification

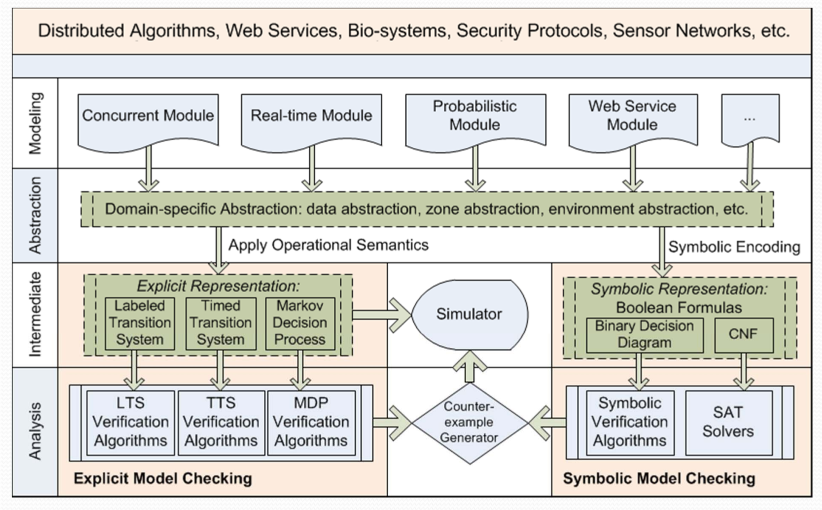
Experiment Results

Conclusion and Future Work

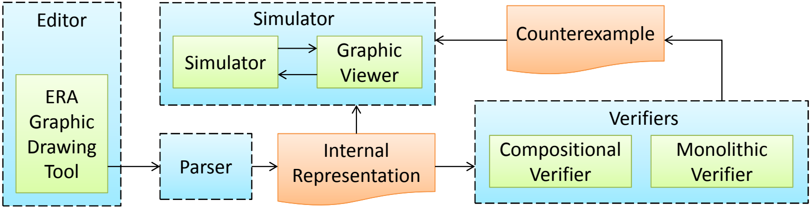
Overall Flow



Architecture of Process Analysis Toolkit (PAT)



The ERA Module



Outline

Event-Recording Automata (ERA)

The TL* Algorithm

Learning-Based Automatic Compositional Verification

Experiment Results

Conclusion and Future Work

Experiment Results

Table 1. Verification Results

				Monolithic				Compositional				UPPAAL
System	n	$\lvert C_\Sigma \rvert$	$\frac{\lvert P_{\not\models} \rvert}{\lvert P \rvert}$	$\lvert L \rvert_{max}$	$\lvert \delta \rvert_{max}$	Time (secs)	Mem (MB)	$\lvert L \rvert_{max}$	$\lvert \delta \rvert_{max}$	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; $\lvert C_\Sigma \rvert$: # of event-recording clocks; $\lvert P \rvert$: # of properties; $\lvert P_{\not\models} \rvert$: # of violated properties; $\lvert L \rvert_{max}$: # of visited locations during verification; $\lvert \delta \rvert_{max}$: # of visited transitions during verification; ROM: run out of memory

Outline

Event-Recording Automata (ERA)

The TL* Algorithm

Learning-Based Automatic Compositional Verification

Experiment Results

Conclusion and Future Work

Conclusion and Future Work

We propose

- a learning algorithm, TL*, for ERAs
- a learning-based compositional verification for timed systems modeled by ERAs

In the future, we plan to

- use different techniques to generate the assumptions
- use different proof rules for AGR