Model Checking Bounded Continuous-time Extended Linear Duration Invariants

Jie An[1], NaiJun Zhan[2], XiaoShan Li[3], MiaoMiao Zhang[1], Wang Yi[4]

[1] School of Software Engineering, Tongji University
[2] State Key Lab of Computer Science, Institute of Software, CAS
[3] Faculty of Science and Technology, University of Macau
[4] Department of Information Technology, Uppsala University

1510796@tongji.edu.cn

April 11, 2018

Overview

1. Introduction
   - Extended Linear Duration Invariants
   - Bounded upper bound and Discrete time semantics
   - Motivation
2. Model checking bounded ELDIs properties
   - Basic idea and framework
   - Main procedure with examples
   - A small case study
   - Benchmark
3. Conclusion and discussion

Overview

Extended Linear Duration Invariants

- **Extended Linear Duration Invariants**, a subset of *Duration Calculus*, extends well-studied *Linear Duration Invariants* with logical connectives and the chop modality.

Extended Linear Duration Invariants

Duration Calculus (DC)

- Real arithmetic extension of ITL with duration. $\int_{t_1}^{t_2} s$.
- [ZHR91, ZH04]

Linear Duration Invariants (LDIs)

- A subset of DC. [ZZYL94]
- Model checking LDIs. [LD96, SPC05, ZLZ09] and other works.
- $a \leq \ell \leq b \implies \sum_{s \in S} c_s \int s \leq M$.
- Gas burner, "the proportion of leak time is not more than one-twentieth of the elapsed time for any time interval at least one minute". $\ell \geq 60 \implies 20 \int Leak \leq \ell$.

$$\ell \geq 60 \implies 19 \int Leak - \int Nonleak \leq 0$$

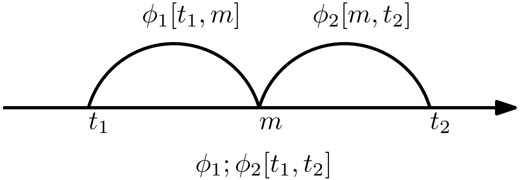
Extended Linear Duration Invariants

Extended Linear Duration Invariants (ELDIs)

- A subset of DC, extending LDIs with logic connective and the chop modality. [FH08]
- State expressions $S ::= 0 \mid P \mid \neg S \mid S_1 \vee S_2$. ($P$ is state variable.)
- Linear duration formulas $\mathcal{D} ::= \sum_{i \in \Omega} c_i \int S_i \leq M$.
- ELDIs formulas $\phi ::= \mathcal{D} \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 ; \phi_2$.
- ELDIs property $\Phi ::= a \leq \ell \leq b \implies \phi$, where b is **bounded** or **unbounded(∞)** and time domain is **discrete** or **continuous**.

Extended Linear Duration Invariants

- An example in Figure 1, $a \leq t_2 - t_1 \leq b \implies \phi_1;\phi_2$.



$$\phi_1;\phi_2[t_1, t_2] : \exists m \cdot (t_1 \leq m \leq t_2 \wedge \phi_1[t_1, m] \wedge \phi_2[m, t_2])$$

Figure 1: The chop modality

- **ELDIs model checking problem on TA $\mathcal{A}$.**

$$\mathcal{A} \models (a \leq \ell \leq b \implies \phi) \quad ?$$

b is bounded or unbounded and time domain is discrete or continuous.

Extended Linear Duration Invariants

- The whole problem is undecidable.
- Comparison with Fränzle and Hansen's work [FH08].

Fränzle and Hansen's work [FH08]

- Approximation semantics
- Presburger Arithmetic
- 4-fold exponential

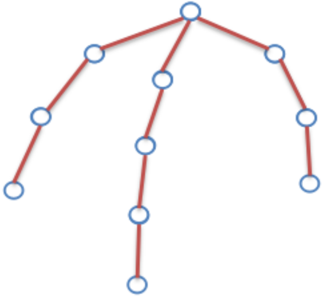
Our work [ZZZZ13]

- Bounded reference time with discrete semantics
- CTL reachability problem
- Single exponential

Overview

Bounded and Discrete time

- Basic idea: verifying every valid execution segments.
- The upper bound of the observation time interval length b is bounded.
- The discrete time semantics.



Bounded and Discrete time

- Reduction to Reachability Problem: checking a CTL property.

Figure 2: Timed Automaton $\mathcal{A}$

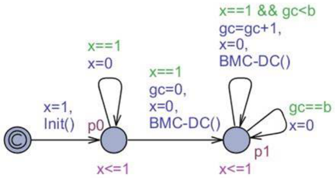


Figure 3: Assistant TA $\mathcal{S}$

ELDIs property:

$$\Phi ::= a \leq \ell \leq b \implies \phi$$

CTL property:

$$\Psi ::= E <> \neg \text{BMC-DC()}$$

Theorem (Bounded and Discrete time [ZZZZ13])

$$\mathcal{A} \models \Phi \quad \textit{iff} \quad \mathcal{A} \parallel \mathcal{S} \not\models \Psi$$

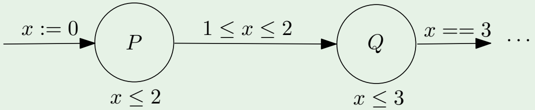
Overview

Motivation

- Checking the satisfaction of LDI $a \leq \ell \leq b \implies \mathcal{D}$ by timed automaton $\mathcal{A}$ in continuous time semantics is equivalent to checking the property in discrete time semantics. [TH04]
- Is there a similar result to ELDIs? **NO !**

A counterexample

- A simple incomplete TA $\mathcal{A}$:

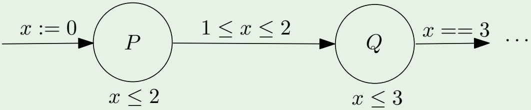


- An ELDIs property Φ:
$3 \leq \ell \leq 3 \implies 2(\int P + \int Q) \geq 3\,;\, 2(\int P + \int Q) \geq 3$

Motivation

A counterexample

- A simple incomplete TA $\mathcal{A}$:

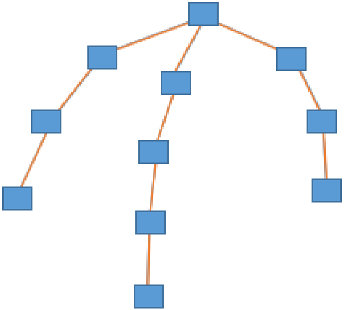


- An ELDIs property Φ:
  $3 \leq \ell \leq 3 \implies 2(\int P + \int Q) \geq 3\,;\, 2(\int P + \int Q) \geq 3$
- Discrete time :
  There are two valid execution segments: P,P,Q and P,Q,Q (one time unit for each state). The chop point can locate at time 0,1,2,3. Obviously, the formula is unsatisfiable.
- Continuous time :
  Whatever the valid execution segment is, there is always a chop point at time 1.5.

Overview

Bounded and Continuous time

- Basic idea: verifying every valid symbolic execution fragments.
- The upper bound of the observation time interval length b is bounded.
- The continuous time: it still has infinite execution fragments of which lengths are in bound. The symbolic execution fragments are finite.
- Finding symbolic execution fragments by Zone.



Bounded and Continuous time

- Reduction to the validity problem of Quantified Linear Real Arithmetic (QLRA).
- QLRA validity problem can be solved by Quantifier Elimination (QE).
- The framework is in Figure 4.

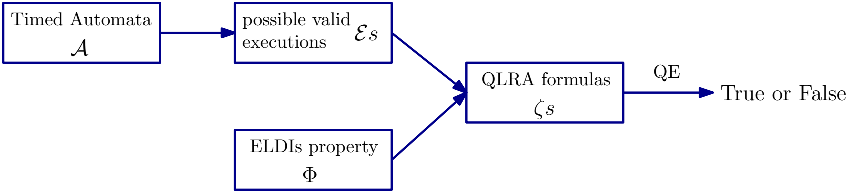
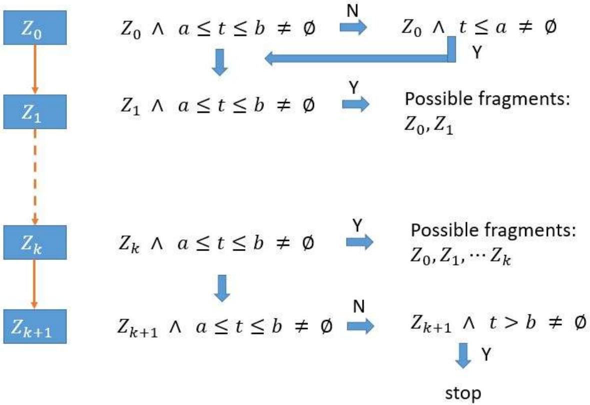


Figure 4: The framework of BMCCELDI

Overview

Finding possible bounded execution fragments

- We use the zone technique which has been implicated in many model checking tools like UPPAAL.
- We introduce an implicit extra clock variable t added to the DBMs.
- The clock variable t will record the time length.



Finding possible bounded execution fragments

- We can use Deep-First Search (DFS) with bound to finding all possible bounded execution fragments.
- Firstly, we check $currentZ \wedge a \leq t \leq b$. If the result is not an empty set, we find a possible fragment and turn to check the post zones.
- Secondly, If the result is an empty set, we check whether $currentZ \wedge t \leq a \neq \emptyset$. If true, we just go deep to check the post zones.
- At last, If the two results are false, we check $currentZ \wedge t > b \neq \emptyset$. It must be true and we can stop and backtrack now.

Reduction to QLRA

- Now, we present a translation from a given possible execution fragment whose length is within the given bound and an ELDI formula into a QLRA formula.

Quantified Linear Real Arithmetic (QLRA)

- A theory of first order logic, with the specific signature $\langle \mathbb{R}, 0, +, =, < \rangle$, i.e., in which all terms are linear.
- Syntax $\zeta := c_0 + c_1x_1 + c_2x_2 + \cdots + c_nx_n \rhd 0 \,|\, \neg\zeta \,|\, \zeta_1 \wedge \zeta_2 \,|\, \forall x.\zeta$, where $c_i \in \mathbb{R}$, $\rhd \in \{=, <\}$.

Reduction to QLRA

An Example

- Timed Automaton $\mathcal{A}$:

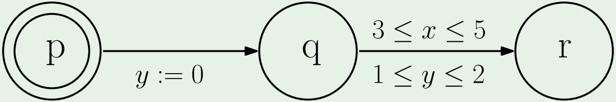


Figure 6: The timed automaton $\mathcal{A}$

- ELDIs property Φ:

$$\ell \leq 6 \implies \int p \leq 2 \,;\, \int q \leq 1$$

Reduction to QLRA

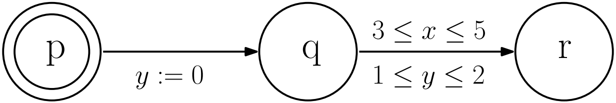
- At First, we derive timing constraints from the execution fragment.
- We introduce a variable δ_i for each location of the execution fragments to indicate the dwelling time length.

Timing Constraints

- Each δ_i should be non-negative.
- Their sum should be within the bound $a \leq \sum_{i=1}^{k} \delta_i \leq b$.
- Replacing the clock variables in each zone of the fragment with δ_is.

Reduction to QLRA

- Given a fragment within the bound: $[p, x = y], [q, x \leq 5 \wedge y \leq 2 \wedge y \leq x], [r, x \geq 3 \wedge y \geq 1 \wedge 1 \leq x - y \leq 4]$
- We introduce three variables $\delta_1, \delta_2, \delta_3$.



Timing Constraints

- $\delta_1 \geq 0 \wedge \delta_2 \geq 0 \wedge \delta_3 \geq 0$;
- $\delta_1 + \delta_2 + \delta_3 \leq 6$;
- $\delta_1 = \delta_1 \wedge \delta_1 + \delta_2 \leq 5 \wedge \delta_2 \leq 2 \wedge \delta_1 + \delta_2 + \delta_3 \geq 3 \wedge \delta_2 + \delta_3 \geq 1 \wedge 1 \leq \delta_1 \leq 4$.

Reduction to QLRA

- At second, we encode the ELDI formula into linear inequations.
- We introduce a variable ϵ_i for each chop. The variable stands for the time length from the entrance point of some location to the chop point if the chop point is at the location.
- Other logic connectives($\wedge, \vee, \neg$) can be encoded directly. The combination formulas can be encoded into QLRA formulas recursively.

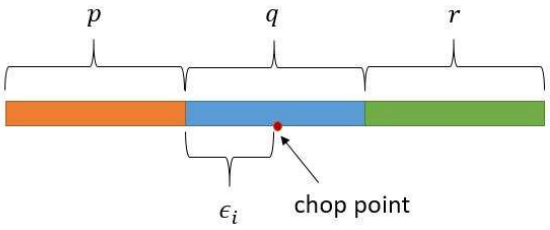
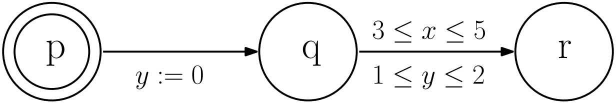


Figure 7: A chop point located at location q

Reduction to QLRA

- Given the execution fragment p, q, r and ELDI formula $\ell \leq 6 \implies \int p \leq 2 \,;\, \int q \leq 1$.
- The chop point could be at location p, q or r.



Three conditions for the chop point

- At location p: $0 \leq \epsilon \leq \delta_1 \wedge \epsilon \leq 2 \wedge 0 \leq \delta_2 \leq 1$;
- At location q: $0 \leq \epsilon \leq \delta_2 \wedge 0 \leq \delta_1 \leq 2 \wedge \delta_2 - \epsilon \leq 1$;
- At location r: $0 \leq \epsilon \leq \delta_3 \wedge 0 \leq \delta_1 \leq 2 \wedge 0 \leq 1$.

Reduction to QLRA

The QLRA formula for the example fragment

$$\begin{aligned}\zeta :=& \forall \delta_1, \delta_2, \delta_3.(\delta_1 \geq 0 \wedge \delta_2 \geq 0 \wedge \delta_3 \geq 0 \wedge \delta_1 + \delta_2 + \delta_3 \leq 6 \wedge \\ & \delta_1 = \delta_1 \wedge \delta_1 + \delta_2 \leq 5 \wedge \delta_2 \leq 2 \wedge \delta_1 + \delta_2 + \delta_3 \geq 3 \wedge \delta_2 + \delta_3 \geq 1 \\ & \wedge 1 \leq \delta_1 \leq 4) \implies \\ & \exists \epsilon.(0 \leq \epsilon \leq \delta_1 \wedge \epsilon \leq 2 \wedge 0 \leq \delta_2 \leq 1) \vee \\ & \quad (0 \leq \epsilon \leq \delta_2 \wedge 0 \leq \delta_1 \leq 2 \wedge \delta_2 - \epsilon \leq 1) \vee \\ & \quad (0 \leq \epsilon \leq \delta_3 \wedge 0 \leq \delta_1 \leq 2 \wedge 0 \leq 1).\end{aligned}$$

Solving derived QLRA formulas

- After encoding all possible execution fragments into the QLRA formulas, we can solve the derived formulas by quantifier elimination (QE).
- Given a TA $\mathcal{A}$ and an ELDI formula ϕ, we can get the conclusion $\mathcal{A}, [a, b] \models \phi$ iff the results of all the QLRA formulas are true.

Theorem (Bounded and Continuous time)

Given a TA $\mathcal{A}$ and an ELDI formula ϕ, $\mathcal{A}, [b, e] \models \phi$ is decidable.

Overview

A small case study

- The anomalous behaviour of priority-driven systems. [Liu00]
- Four independent jobs J_1, J_2, J_3, J_4 are scheduled on two identical processors P_1 and P_2 in a priority-driven manner $J_1 > J_2 > J_3 > J_4$.
- The informations of jobs is shown in Figure 8.
- The question is whether the jobs can be finished within 20 time units.

	r	d	$[e^-, e^+]$
J_1	0	10	5
J_2	0	10	$[2, 6]$
J_3	4	15	8
J_4	0	20	10

Figure 8: The informations of jobs

A small case study

- The job J_1 can run on P_1 with the highest priority.
- The timed automaton of the processor P_2 is shown in Figure 9.
- We can check the ELDI property:
  $20 \le \ell \le 20 \implies [(2 \le \int run_{J_2} \le 6 \wedge \int run_{J_2} - \int 1 = 0); ((\int run_{J_3} = 0 \vee \int run_{J_3} = 8) \wedge (\int run_{J_4} = 0 \vee \int run_{J_4} = 10) \wedge (0 < \int run_{J_3} + \int run_{J_4} \le 18))]$.
- The checking result is false, which means the jobs could not be finished within 20 time units.

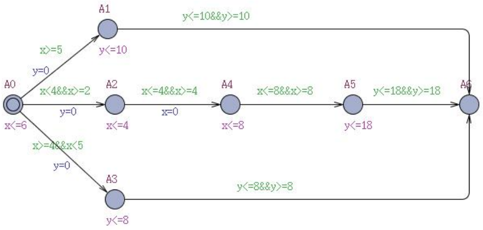


Figure 9: The TA of the schedule

A small case study

- The anomalous behaviour may occur when the execution time of J_2 choose a value in the interval $(2, 6)$.

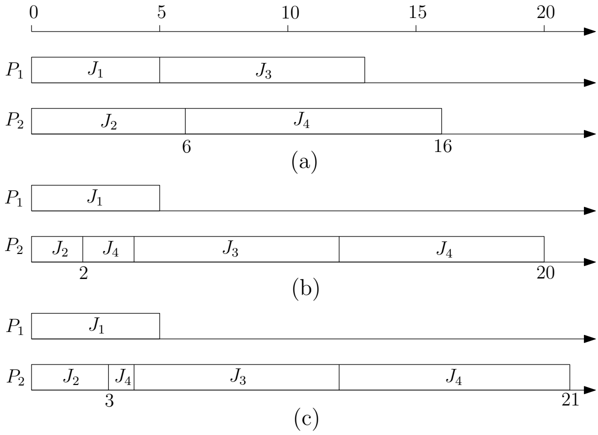


Figure 10: The anomalous behaviour

Overview

Benchmark

- We conduct some experiments on a laptop with Inter Core i3-5005U at 2.0GHz and 4GB DDR3L-1600MHz RAM.
- The benchmark is shown in the below table.

NO.	Location numbers	Clock numbers	QLRA numbers	time (s)
1	10	3	175	6.1
2	12	1	506	1.5
3	16	1	794	2.2
4	20	1	1135	3.4
5	24	1	1530	5.1
6	23	2	356	4.1
7	7	2	13	0.02
8	58	2	7237	112.6
9	58	2	372167	7560

Conclusion and discussion

Conclusion and discussion

- Bounded and Continuous time: reduction to QLRA validity problem.
- The complexity of our approach is 3-fold exponential in the size of TA $\mathcal{A}$ and 2-fold exponential in the number of nested chops in ELDI formula ϕ.
- Although the theoretical complexity of our approach is quite high, in practice, the worst cases happen with quite low possibility.

References

- [ZHR91]: A calculus of durations. *Inf. Proc. Let. 40, 5 (1991), 269–276.*
- [ZZYL94]: Linear duration invariants. *In FTRTFT 1994. 86–109.*
- [LD96]: Checking linear duration invariants by linear programming. *In ASIAN 1996. 321–332.*
- [ACM02]: Timed Regular Expressions. *Journal of the ACM. 49, 2 (2002), 172–206.*
- [ZH04]: Duration Calculus: A Formal Approach to Real-Time Systems. *Springer, 2004.*
- [TH04]: Verifying linear duration constraints of timed automata. *In ICTAC 2004. 295–309.*
- [SPC05]: Bounded Validity Checking of Interval Duration Logic. *In TACAS 2005. 301–316.*
- [FH08]: Efficient model checking for duration calculus based on branching-time approximations. *In SEFM 2008. 63–72.*
- [ZLZ09]: Model checking linear duration invariants of networks of automata. *In FSEN 2009. 244–259.*
- [ZZZZ13]: Bounded model-checking of discrete duration calculus. *In HSCC 2013. 213–222.*
- [Liu00]: Real-Time Systems. *Prentice Hall, 2000.*

Thanks