Model Checking Timed Recursive CTL

Florian Bruse Martin Lange

University of Kassel, Germany

GI Meeting Deduction and Logic
08/04/2022

Temporal Logic with Recursion

proposal to enhance expressiveness of temporal logics: add recursion operator

e.g. CTL:

$$\varphi \quad ::= \quad q \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{EX}\varphi \mid \mathtt{E}(\varphi \,\mathtt{U}\, \varphi) \mid \mathtt{EG}\varphi$$

Temporal Logic with Recursion

proposal to enhance expressiveness of temporal logics: add recursion operator

e.g. Recursive CTL (RecCTL):

$$\begin{aligned} \varphi &::= q \mid x \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{EX}\varphi \mid \mathtt{E}(\varphi\,\mathtt{U}\,\varphi) \mid \mathtt{EG}\varphi \mid \Phi(\varphi_1, \ldots, \varphi_k) \\ \Phi &::= \mathcal{F} \mid \mathtt{rec}\,\mathcal{F}(x_1, \ldots, x_k).\varphi \end{aligned}$$

two types of formulas:

- propositional (φ): as usual interpreted in $2^{\mathcal{S}}$
- first-order (Φ): interpreted as function of type $2^{\mathcal{S}} \times \ldots \times 2^{\mathcal{S}} \to 2^{\mathcal{S}}$

Temporal Logic with Recursion

proposal to enhance expressiveness of temporal logics: add recursion operator

e.g. Recursive CTL (RecCTL):

$$\begin{aligned} \varphi &::= q \mid x \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathtt{EX}\varphi \mid \mathtt{E}(\varphi\,\mathtt{U}\,\varphi) \mid \mathtt{EG}\varphi \mid \Phi(\varphi_1, \ldots, \varphi_k) \\ \Phi &::= \mathcal{F} \mid \mathtt{rec}\,\mathcal{F}(x_1, \ldots, x_k).\varphi \end{aligned}$$

two types of formulas:

- propositional (φ): as usual interpreted in $2^{\mathcal{S}}$
- first-order (Φ): interpreted as function of type $2^{\mathcal{S}} \times \ldots \times 2^{\mathcal{S}} \to 2^{\mathcal{S}}$

Proposition 1 ([B./Lange'20])

a) *RecCTL is more expressive than the modal μ-calculus.*

b) *Model checking RecCTL is* EXPTIME*-complete.*

Timed Recursive CTL

aim: extend high expressiveness with decidable model checking to verification of real-time systems

Timed Recursive CTL

aim: extend high expressiveness with decidable model checking to verification of real-time systems

Timed Recursive CTL (TRCTL) as merger of Timed CTL and RecCTL:

$$\begin{aligned} \varphi &::= q \mid x \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathtt{E}(\varphi \,\mathtt{U}^{J}\, \varphi) \mid \Phi(\varphi_1, \ldots, \varphi_k) \\ \Phi &::= \mathcal{F} \mid \mathtt{rec}\, \mathcal{F}(x_1, \ldots, x_k).\varphi \end{aligned}$$

with J an interval over $\mathbb{Q}$

Timed Recursive CTL

aim: extend high expressiveness with decidable model checking to verification of real-time systems

Timed Recursive CTL (TRCTL) as merger of Timed CTL and RecCTL:

$$\begin{aligned} \varphi &::= q \mid x \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathtt{E}(\varphi \,\mathtt{U}^{J}\, \varphi) \mid \Phi(\varphi_1, \ldots, \varphi_k) \\ \Phi &::= \mathcal{F} \mid \mathtt{rec}\, \mathcal{F}(x_1, \ldots, x_k).\varphi \end{aligned}$$

with J an interval over $\mathbb{Q}$

Ex.: let $\Phi := \mathtt{rec}\, \mathcal{F}(x, y).(x \to y) \wedge \mathcal{F}(\mathtt{EF}^{\leq 3} x, \mathtt{EF}^{\leq 2} y)$, then

$\Phi(p, p)$

Timed Recursive CTL

aim: extend high expressiveness with decidable model checking to verification of real-time systems

Timed Recursive CTL (TRCTL) as merger of Timed CTL and RecCTL:

$$\begin{aligned} \varphi &::= q \mid x \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathrm{E}(\varphi \,\mathrm{U}^{J}\, \varphi) \mid \Phi(\varphi_1, \ldots, \varphi_k) \\ \Phi &::= \mathcal{F} \mid \mathtt{rec}\, \mathcal{F}(x_1, \ldots, x_k).\varphi \end{aligned}$$

with J an interval over $\mathbb{Q}$

Ex.: let $\Phi := \mathtt{rec}\, \mathcal{F}(x, y).(x \to y) \wedge \mathcal{F}(\mathrm{EF}^{\leq 3} x, \mathrm{EF}^{\leq 2} y)$, then

$$\Phi(p, p) \equiv \big((x \to y) \wedge \Phi(\mathrm{EF}^{\leq 3} x, \mathrm{EF}^{\leq 2} y)\big)(p, p)$$

Timed Recursive CTL

aim: extend high expressiveness with decidable model checking to verification of real-time systems

Timed Recursive CTL (TRCTL) as merger of Timed CTL and RecCTL:

$$\begin{aligned} \varphi &::= q \mid x \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathrm{E}(\varphi \,\mathrm{U}^{J}\, \varphi) \mid \Phi(\varphi_1, \ldots, \varphi_k) \\ \Phi &::= \mathcal{F} \mid \mathtt{rec}\, \mathcal{F}(x_1, \ldots, x_k).\varphi \end{aligned}$$

with J an interval over $\mathbb{Q}$

Ex.: let $\Phi := \mathtt{rec}\, \mathcal{F}(x, y).(x \to y) \wedge \mathcal{F}(\mathrm{EF}^{\leq 3} x, \mathrm{EF}^{\leq 2} y)$, then

$$\Phi(p,p) \equiv \big((x \to y) \wedge \Phi(\mathrm{EF}^{\leq 3} x, \mathrm{EF}^{\leq 2} y)\big)(p,p) \equiv (p \to p) \wedge \Phi(\mathrm{EF}^{\leq 3} p, \mathrm{EF}^{\leq 2} p) \equiv \Phi(\mathrm{EF}^{\leq 3} p, \mathrm{EF}^{\leq 2} p)$$

Timed Recursive CTL

aim: extend high expressiveness with decidable model checking to verification of real-time systems

Timed Recursive CTL (TRCTL) as merger of Timed CTL and RecCTL:

$$\begin{aligned} \varphi &::= q \mid x \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathrm{E}(\varphi \,\mathrm{U}^{J}\, \varphi) \mid \Phi(\varphi_1, \ldots, \varphi_k) \\ \Phi &::= \mathcal{F} \mid \mathtt{rec}\, \mathcal{F}(x_1, \ldots, x_k).\varphi \end{aligned}$$

with J an interval over $\mathbb{Q}$

Ex.: let $\Phi := \mathtt{rec}\, \mathcal{F}(x, y).(x \to y) \wedge \mathcal{F}(\mathrm{EF}^{\leq 3} x, \mathrm{EF}^{\leq 2} y)$, then

$$\begin{aligned} \Phi(p,p) &\equiv \big((x \to y) \wedge \Phi(\mathrm{EF}^{\leq 3} x, \mathrm{EF}^{\leq 2} y)\big)(p,p) \equiv (p \to p) \wedge \Phi(\mathrm{EF}^{\leq 3} p, \mathrm{EF}^{\leq 2} p) \equiv \Phi(\mathrm{EF}^{\leq 3} p, \mathrm{EF}^{\leq 2} p) \\ &\equiv (\mathrm{EF}^{\leq 3} p \to \mathrm{EF}^{\leq 2} p) \wedge \Phi(\mathrm{EF}^{\leq 3}\mathrm{EF}^{\leq 3} p, \mathrm{EF}^{\leq 2}\mathrm{EF}^{\leq 2} p) \end{aligned}$$

Timed Recursive CTL

aim: extend high expressiveness with decidable model checking to verification of real-time systems

Timed Recursive CTL (TRCTL) as merger of Timed CTL and RecCTL:

$$\begin{aligned} \varphi &::= q \mid x \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathrm{E}(\varphi \,\mathrm{U}^{J}\, \varphi) \mid \Phi(\varphi_1, \ldots, \varphi_k) \\ \Phi &::= \mathcal{F} \mid \mathtt{rec}\, \mathcal{F}(x_1, \ldots, x_k).\varphi \end{aligned}$$

with J an interval over $\mathbb{Q}$

Ex.: let $\Phi := \mathtt{rec}\, \mathcal{F}(x,y).(x \to y) \wedge \mathcal{F}(\mathrm{EF}^{\leq 3}x, \mathrm{EF}^{\leq 2}y)$, then

$$\begin{aligned} \Phi(p,p) &\equiv \big((x \to y) \wedge \Phi(\mathrm{EF}^{\leq 3}x, \mathrm{EF}^{\leq 2}y)\big)(p,p) \equiv (p \to p) \wedge \Phi(\mathrm{EF}^{\leq 3}p, \mathrm{EF}^{\leq 2}p) \equiv \Phi(\mathrm{EF}^{\leq 3}p, \mathrm{EF}^{\leq 2}p) \\ &\equiv (\mathrm{EF}^{\leq 3}p \to \mathrm{EF}^{\leq 2}p) \wedge \Phi(\mathrm{EF}^{\leq 3}\mathrm{EF}^{\leq 3}p, \mathrm{EF}^{\leq 2}\mathrm{EF}^{\leq 2}p) \equiv (\mathrm{EF}^{\leq 3}p \to \mathrm{EF}^{\leq 2}p) \wedge \Phi(\mathrm{EF}^{\leq 6}p, \mathrm{EF}^{\leq 4}p) \end{aligned}$$

Timed Recursive CTL

aim: extend high expressiveness with decidable model checking to verification of real-time systems

Timed Recursive CTL (TRCTL) as merger of Timed CTL and RecCTL:

$$\begin{aligned} \varphi &::= q \mid x \mid \varphi \vee \varphi \mid \neg\varphi \mid \mathrm{E}(\varphi \,\mathrm{U}^{J}\, \varphi) \mid \Phi(\varphi_1, \ldots, \varphi_k) \\ \Phi &::= \mathcal{F} \mid \mathtt{rec}\, \mathcal{F}(x_1, \ldots, x_k).\varphi \end{aligned}$$

with J an interval over $\mathbb{Q}$

Ex.: let $\Phi := \mathtt{rec}\, \mathcal{F}(x,y).(x \to y) \wedge \mathcal{F}(\mathrm{EF}^{\leq 3}x, \mathrm{EF}^{\leq 2}y)$, then

$$\begin{aligned} \Phi(p,p) &\equiv \big((x \to y) \wedge \Phi(\mathrm{EF}^{\leq 3}x, \mathrm{EF}^{\leq 2}y)\big)(p,p) \equiv (p \to p) \wedge \Phi(\mathrm{EF}^{\leq 3}p, \mathrm{EF}^{\leq 2}p) \equiv \Phi(\mathrm{EF}^{\leq 3}p, \mathrm{EF}^{\leq 2}p) \\ &\equiv (\mathrm{EF}^{\leq 3}p \to \mathrm{EF}^{\leq 2}p) \wedge \Phi(\mathrm{EF}^{\leq 3}\mathrm{EF}^{\leq 3}p, \mathrm{EF}^{\leq 2}\mathrm{EF}^{\leq 2}p) \equiv (\mathrm{EF}^{\leq 3}p \to \mathrm{EF}^{\leq 2}p) \wedge \Phi(\mathrm{EF}^{\leq 6}p, \mathrm{EF}^{\leq 4}p) \\ &\equiv \ldots \equiv \bigwedge_{n \geq 1} (\mathrm{EF}^{\leq 3n}p \to \mathrm{EF}^{\leq 2n}p) \end{aligned}$$

TRCTL and Timed Automata

TRCTL interpreted over timed LTS, typically represented by timed automata (standard)

recursion formulas interpreted as least fixpoints in pointwise-ordered function space

TRCTL and Timed Automata

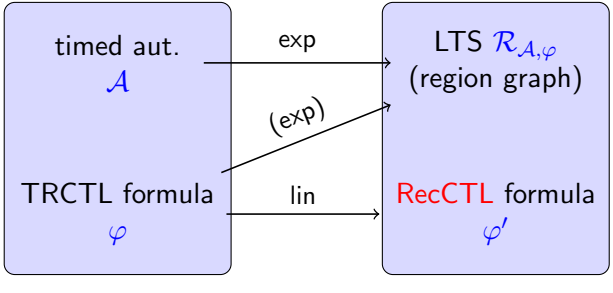
TRCTL interpreted over timed LTS, typically represented by timed automata (standard)

recursion formulas interpreted as least fixpoints in pointwise-ordered function space

Theorem 1

Model checking TRCTL over timed automata is in 2EXPTIME.

PROOF: uses standard untiming region abstraction



TRCTL and Timed Automata

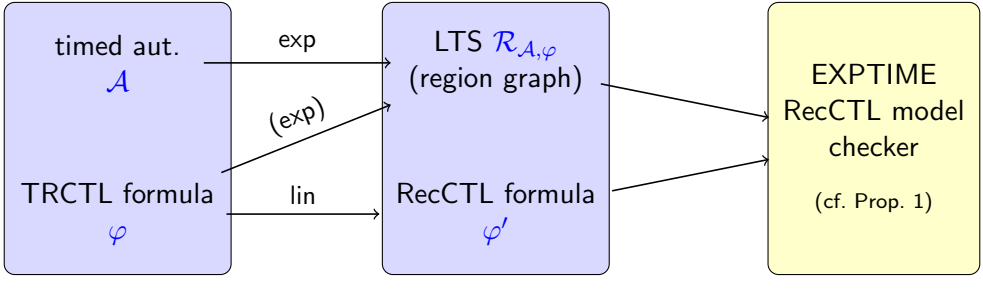
TRCTL interpreted over timed LTS, typically represented by timed automata (standard)

recursion formulas interpreted as least fixpoints in pointwise-ordered function space

Theorem 1

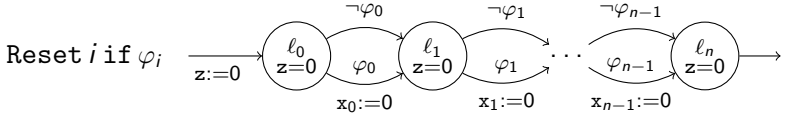
Model checking TRCTL over timed automata is in 2EXPTIME.

PROOF: uses standard untiming region abstraction



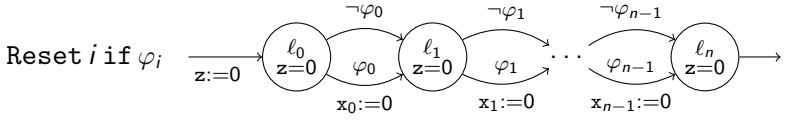
Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton

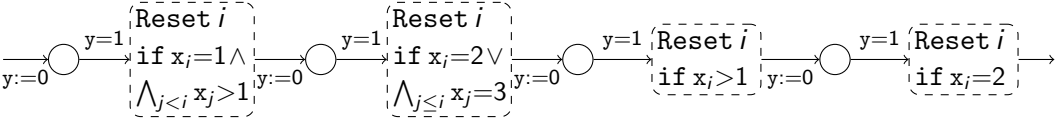


Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



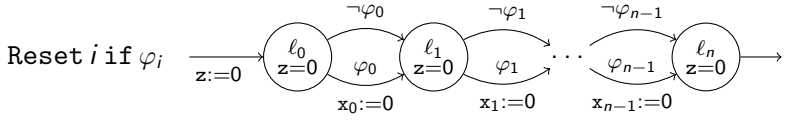
now what does



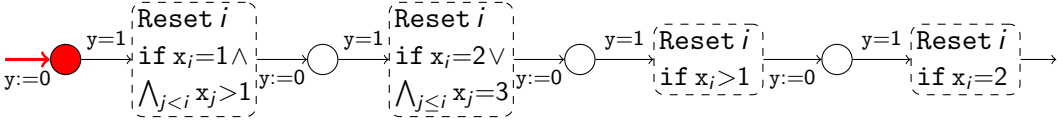
do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



now what does



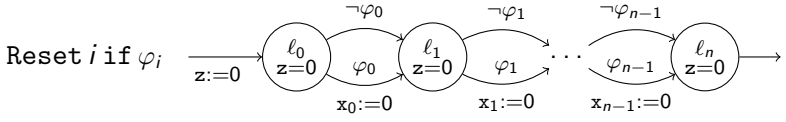
do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

take for instance

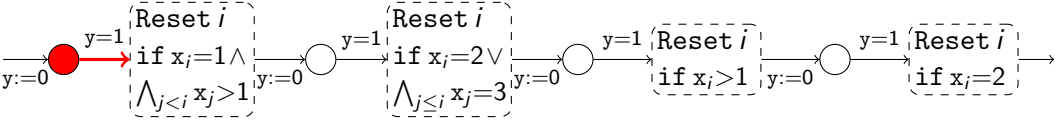
x_0	x_1	x_2	x_3	x_4	representing
1	1	0	0	1	19

Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



now what does



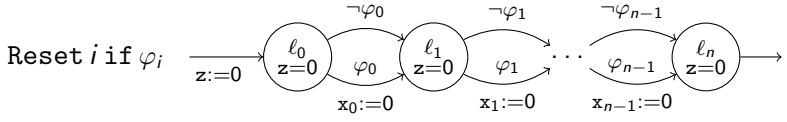
do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

take for instance

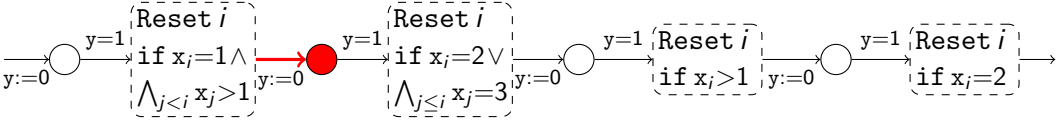
x_0	x_1	x_2	x_3	x_4	representing
2	2	1	1	2	

Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



now what does



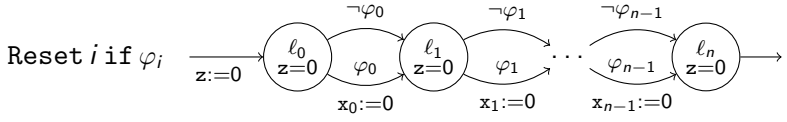
do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

take for instance

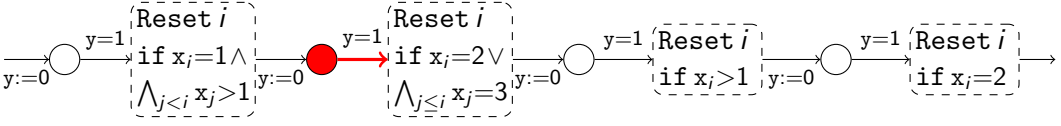
x_0	x_1	x_2	x_3	x_4	representing
2	2	0	1	2	

Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



now what does



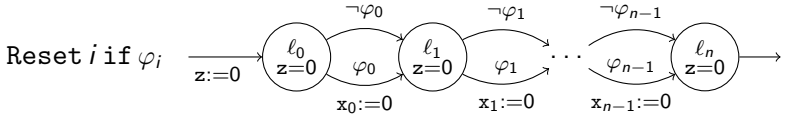
do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

take for instance

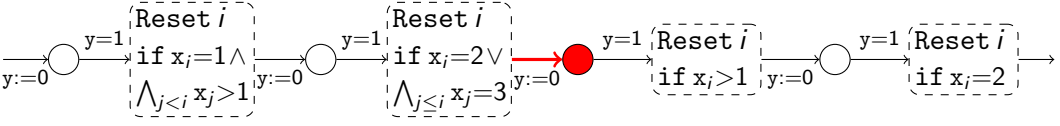
x_0	x_1	x_2	x_3	x_4	representing
3	3	1	2	3	

Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



now what does



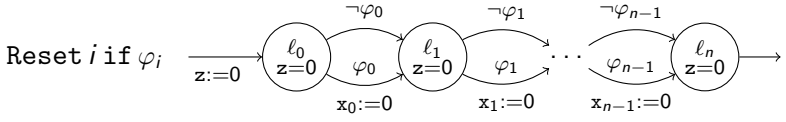
do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

take for instance

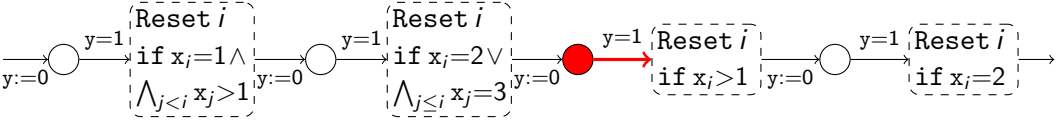
x_0	x_1	x_2	x_3	x_4	representing
0	0	1	0	3	

Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



now what does



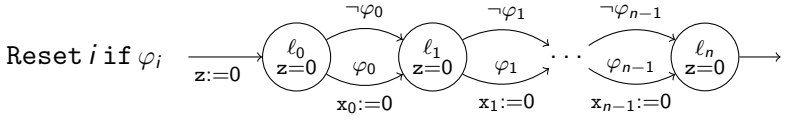
do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

take for instance

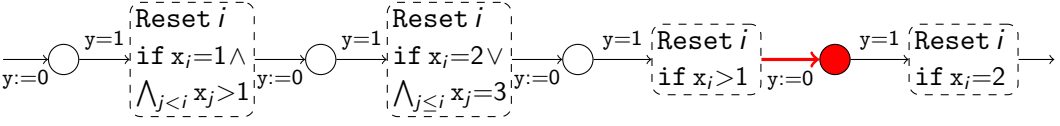
x_0	x_1	x_2	x_3	x_4	representing
1	1	2	1	4	

Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



now what does



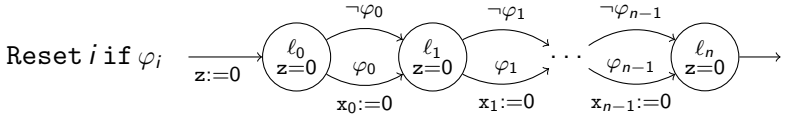
do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

take for instance

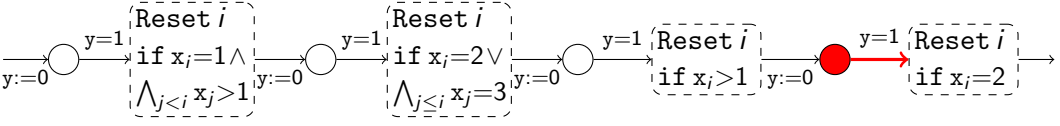
x_0	x_1	x_2	x_3	x_4	representing
1	1	0	1	0	

Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



now what does



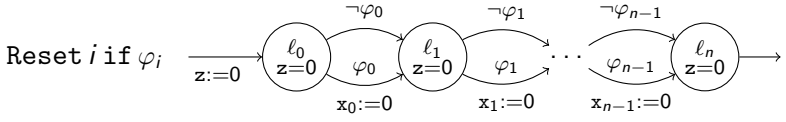
do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

take for instance

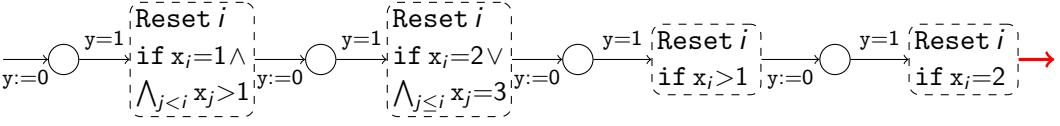
x_0	x_1	x_2	x_3	x_4	representing
2	2	1	2	1	

Manipulating Small Binary Counters in Timed Automata

Ex.: helpful "macro" timed automaton



now what does



do with clock evaluation $\{x_0, \ldots, x_{n-1}\} \to \{0, 1\}$?

take for instance

x_0	x_1	x_2	x_3	x_4	representing
0	0	1	0	1	20

Encoding Large Binary Counters in TRCTL Formulas

note:

- particular clock evaluations $\rightsquigarrow$ (small) counter for values in $[2^n]$
- set of small counter values naturally encodes (large) counter value in $[2^{2^n}]$

Encoding Large Binary Counters in TRCTL Formulas

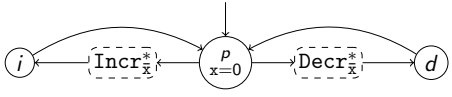
note:

- particular clock evaluations $\rightsquigarrow$ (small) counter for values in $[2^n]$
- set of small counter values naturally encodes (large) counter value in $[2^{2^n}]$
- TRCTL first-order formulas can be used to manipulate arguments, representing large counter values

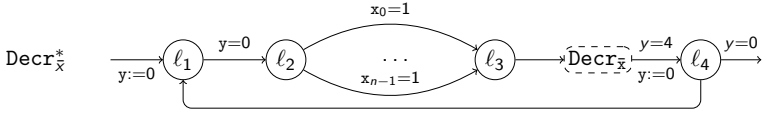
Lemma: if x encodes a large counter value, then

$$inc(x) = (x \wedge \mathtt{E}(\neg(p \vee i)\ \mathtt{U}^{>0}\ (p \wedge \neg x))) \vee (\neg x \wedge \mathtt{A}((\neg p \wedge \neg i)\,\mathtt{U}^{>0}\,(p \wedge x)))$$

encodes $x + 1$ in



with



The Lower Bound

Theorem 2

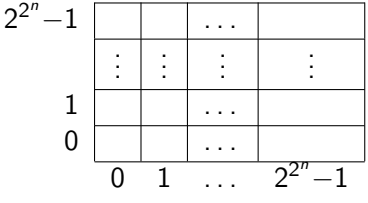
Model checking TRCTL over timed automata is 2EXPTIME*-hard.*

PROOF IDEA: given TM $\mathcal{M}$, $n \in \mathbb{N}$, construct TCTL formula of the form

$$\begin{aligned}(\,&\mathtt{rec}\; Chk(t,s).\ldots.Chk(dec(t),dec(s))\\ &\wedge Chk(dec(t),s)\\ &\wedge Chk(dec(t),inc(s))\ldots)(\mathtt{tt},\mathtt{ff})\end{aligned}$$

that checks for the existence of a tableau encoding an accepting run of $\mathcal{M}$

(uses characterisation of 2EXPTIME from [Chandra/Kozen/Stockmeyer'81])



Conclusion

we have ...

- introduced a highly expressive specification logic for real-time systems
- shown that its model checking problem is decidable

Conclusion

we have ...

- introduced a highly expressive specification logic for real-time systems
- shown that its model checking problem is decidable

comments:

- lower bound shown for TA with non-constant number of clocks
- recent result: can be improved to one clock by relying more on higher-order mechanics (cf. [Laroussinie/Markey/Schnoebelen'04])
- lower bound holds for restricted sets of clock constraints (e.g. only $>$, only $=$), and already for expression complexity
- does model checking remain decidable over extensions of TA?