Efficient Emptiness Check for Timed Büchi Automata

F. Herbreteau, B. Srivathsan and I. Walukiewicz

Université de Bordeaux, LaBRI - CNRS

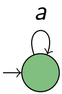
August 2010

Timed Büchi Automata [AD94]

Timed Büchi Automata [AD94]

Finite words

$L_{finite} = a^*$

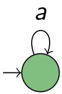


Finite automata

Timed Büchi Automata [AD94]

Finite words

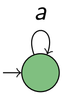
$$L_{finite} = a^*$$



Finite automata

Infinite words

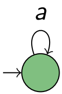
$$L_{infinite} = a^\omega$$



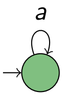
Büchi automata

Timed Büchi Automata [AD94]

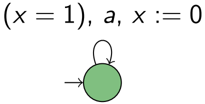
Finite words	Infinite words	Timed words
$L_{finite} = a^*$	$L_{infinite} = a^\omega$	$L_t = (a,1)(a,2)\ldots$



Finite automata



Büchi automata

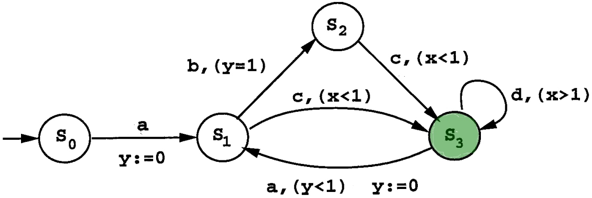


Timed automata

Clocks: can be

- **compared** with integers, diagonal-free constraints
- **reset** to 0

Timed Büchi Automata [AD94]



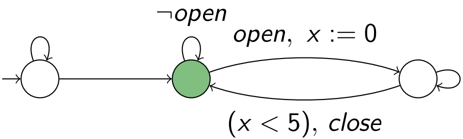
Run: infinite sequence of transitions

$$(s_0, \overbrace{0}^{x}, \overbrace{0}^{y}) \xrightarrow{0.4,a} (s_1, 0.4, 0) \xrightarrow{0.5,c} (s_3, 0.9, 0.5) \xrightarrow{0.3,d} (s_3, 1.2, 0.8) \xrightarrow{15,d} \cdots$$

- **accepting** if infinitely often green
- **non-Zeno** if time diverges ($\sum_{i \geq 0} \delta_i \to \infty$)

Model-Checking Real-Time Systems

Correctness: Safety + **Liveness** + **Fairness**



"Infinitely often, the gate is open for at least 5 s."

Realistic counter-examples: infinite **non-Zeno** runs

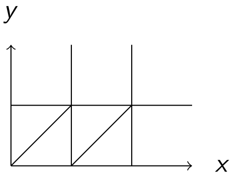
The Problem That We Consider

Given a TBA A, does it **have** a **non-Zeno** accepting run?

Theorem [AD94]

Deciding if a TBA has a non-Zeno accepting run is **PSPACE-complete**

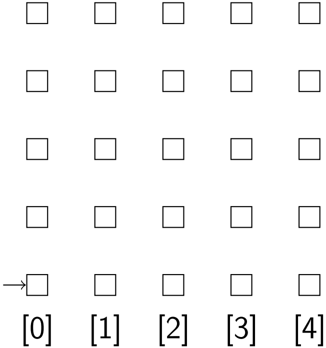
Regions [AD94]



- 6 Corner points, e.g $[(0, 1)]$
- 14 Open line segments, e.g $[0 < x = y < 1]$
- 8 Open regions, e.g $[0 < x < y < 1]$

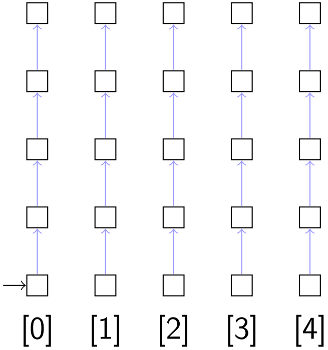
- Region: set of valuations that satisfy the **same guards** w.r.t. time

Region Graph



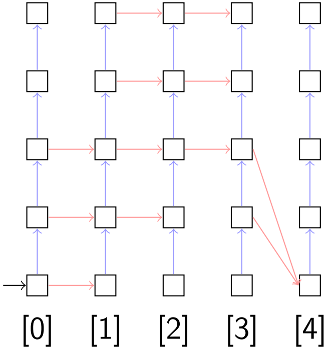
- Region: set of valuations that satisfy the **same guards** w.r.t. time

Region Graph



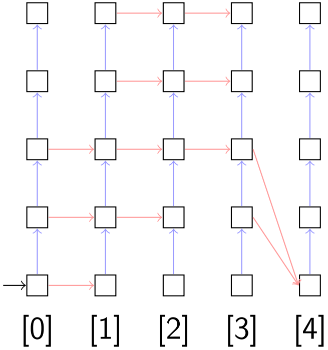
- Region: set of valuations that satisfy the **same guards** w.r.t. time

Region Graph



- Region: set of valuations that satisfy the **same guards** w.r.t. time

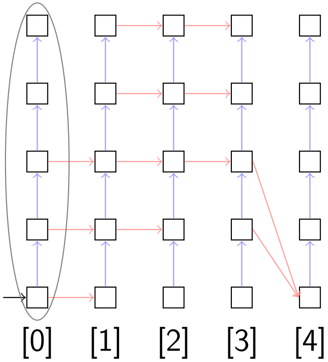
Region Graph



- Region: set of valuations that satisfy the **same guards** w.r.t. time

$\mathcal{O}(|X|!.M^{|X|})$ many regions!

Region Graph

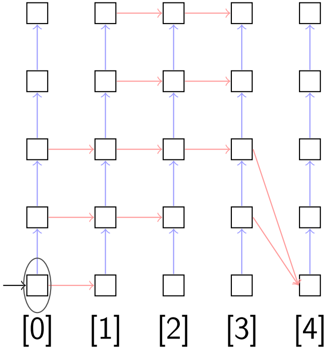


- Region: set of valuations that satisfy the **same guards** w.r.t. time

$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

Region Graph & Zone Graph

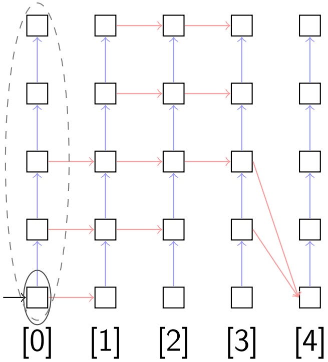


- Region: set of valuations that satisfy the **same guards** w.r.t. time

$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

Region Graph & Zone Graph

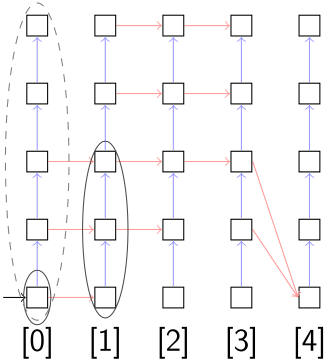


- Region: set of valuations that satisfy the **same guards** w.r.t. time

$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

Region Graph & Zone Graph

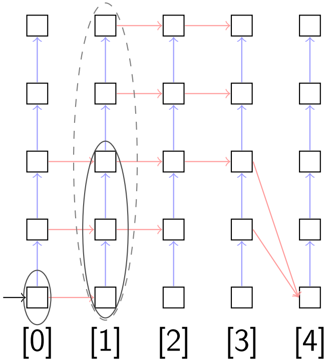


- Region: set of valuations that satisfy the **same guards** w.r.t. time

$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

Region Graph & Zone Graph

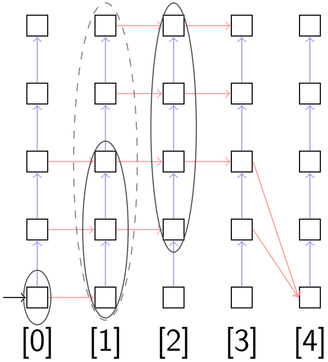


- Region: set of valuations that satisfy the **same guards** w.r.t. time

$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

Region Graph & Zone Graph

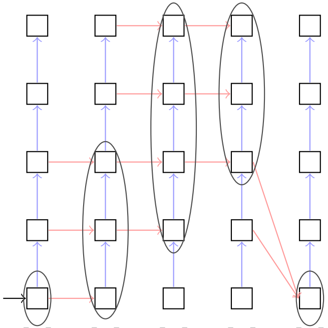


- Region: set of valuations that satisfy the **same guards** w.r.t. time

$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

Region Graph & Zone Graph

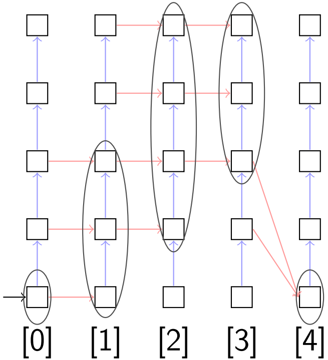


- Region: set of valuations that satisfy the **same guards** w.r.t. time

$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

Region Graph & Zone Graph



- Region: set of valuations that satisfy the **same guards** w.r.t. time

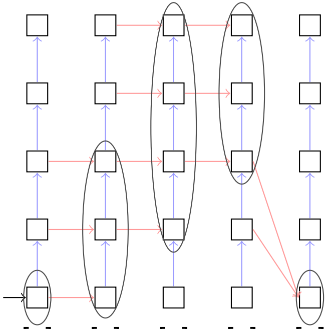
$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

Finite accepting conditions [AD94, Bou04]

Both **regions** and **zones** preserve state **reachability**

Region Graph & Zone Graph



- Region: set of valuations that satisfy the **same guards** w.r.t. time

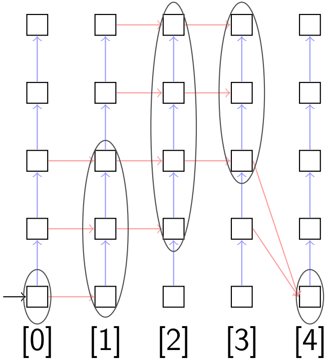
$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

Büchi accepting conditions [AD94, Tri09]

Both **regions** and **zones** preserve **repeated state** reachability

Region Graph & Zone Graph



- Region: set of valuations that satisfy the **same guards** w.r.t. time

$\mathcal{O}(|X|!.M^{|X|})$ many regions!

- Zone: convex **union of regions**

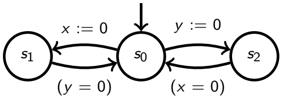
non-Zenoness

- Region: an extra **time progress criterion** on paths [AD94]
- Zone: **???**

Time Progress in the Zone Graph

Time Progress Criterion [AD94]

$$\bigwedge_{x \in X} \text{unbounded}(x) \vee \text{fluctuating}(x)$$



- Path in RG(A):

$$(s_1, 0 = x < y) \qquad (s_2, 0 = y < x)$$

$$(s_0, 0 = x = y) \to (s_1, 0 = x = y) \to (s_0, 0 = x = y) \to (s_2, 0 = y = x) \dashrightarrow$$

- Path in ZG(A):

$$(s_0, 0 = x = y) \to (s_1, 0 = x \leq y) \to (s_0, 0 = x = y) \to (s_2, 0 = y \leq x) \dashrightarrow$$

The **time progress** criterion is **not sound** on ZG(A)

Outline

Outline

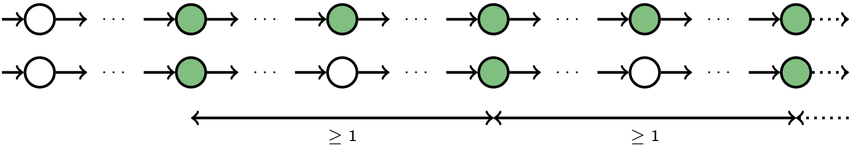
Standard Reduction: Combinatorial Explosion

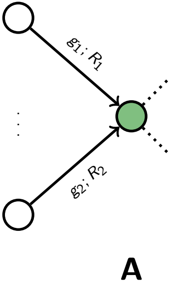
A New Construction

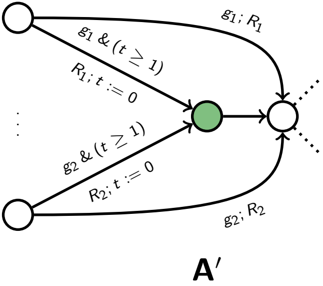
Conclusion

From TBA to Strongly non-Zeno TBA [TYB05]

Key Idea : reduce non-Zenoness to Büchi acceptation







Strongly non-Zeno TBA [Tri99, TYB05]

Definition

Strongly non-Zeno TBA: **all** accepting runs are **non-Zeno**

Theorem [TYB05]

For every TBA A, there exists a Strongly non-Zeno TBA A' that has an **accepting** run iff A has a **non-Zeno accepting** run

(size of A': $|X| + 1$ clocks and at most $2|Q|$ states)

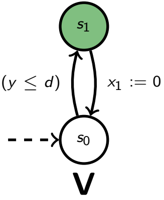
Theorem [Tri09]

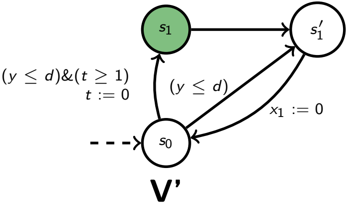
A has a non-Zeno accepting run iff $\mathrm{ZG}(A')$ has an accepting run

Coming Next on Strongly non-Zeno Construction

Adding one clock leads to an **exponential blowup** in the Zone Graph!

Guard $t \geq 1$ Allows to Count...





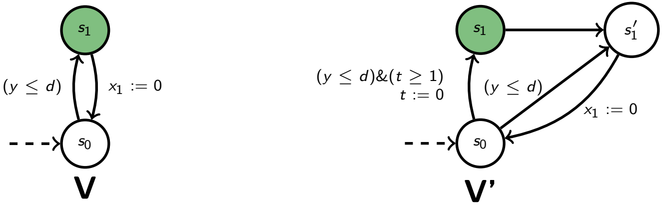
Run of **V**: 2 different zones in s_0

$$\cdots (s_0, y \leq x_1 \leq x_2) \xrightarrow{y \leq d} ({\color{green}s_1}, y \leq x_1 \leq x_2 \,\&\, y \leq d) \xrightarrow{x_1 := 0}$$

$$(s_0, 0 = x_1 \leq y \leq x_2) \xrightarrow{y \leq d} ({\color{green}s_1}, x_1 \leq y \leq x_2 \,\&\, y \leq d) \xrightarrow{x_1 := 0}$$

$$(s_0, 0 = x_1 \leq y \leq x_2) \cdots$$

Guard $t \geq 1$ Allows to Count...



Run of **V'**: $d + 2$ different zones in s_0

$$\cdots (s_0, y \leq x_1 \leq x_2 \leq t) \xrightarrow{(y \leq d)\&(t \geq 1),\, t:=0} \longrightarrow \xrightarrow{x_1 := 0}$$

$$(s_0, 0 = x_1 \leq t \leq y \leq x_2 \& \mathbf{y} - \mathbf{t} \geq \mathbf{0}) \xrightarrow{(y \leq d)\&(t \geq 1),\, t:=0} \longrightarrow \xrightarrow{x_1 := 0}$$

$$(s_0, 0 = x_1 \leq t \leq y \leq x_2 \& \mathbf{y} - \mathbf{t} \geq \mathbf{1}) \xrightarrow{(y \leq d)\&(t \geq 1),\, t:=0} \longrightarrow \xrightarrow{x_1 := 0}$$

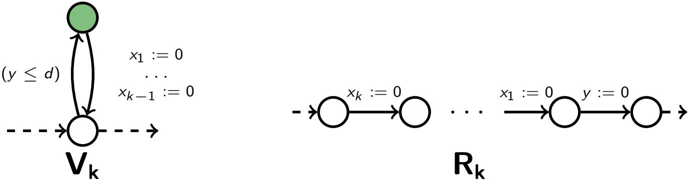
$$(s_0, 0 = x_1 \leq t \leq y \leq x_2 \& \mathbf{y} - \mathbf{t} \geq \mathbf{2}) \xrightarrow{(y \leq d)\&(t \geq 1),\, t:=0} \longrightarrow \xrightarrow{x_1 := 0}$$

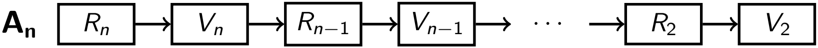
$$\ldots$$

$$(s_0, 0 = x_1 \leq t \leq y \leq x_2 \& \mathbf{y} - \mathbf{t} \geq \mathbf{d})$$

Remark: $y - t \geq c$ implies $x_2 - x_1 \geq c$

...and Leads to a Combinatorial Explosion



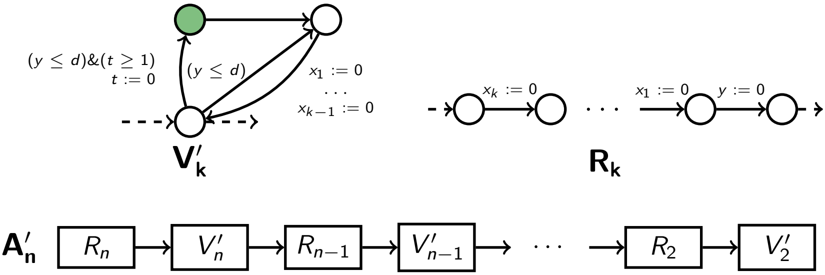


Lemma

$ZG(A_n)$ has linear size in n

Key Idea: at V_k only two possible zones that **collapse** to the same zone after R_{k-1}.

...and Leads to a Combinatorial Explosion



Lemma

$ZG(A'_n)$ has size exponential in n

Key Idea: at V'_k, $\bigwedge_{i \in [k;n]} x_i - x_{i-1} \geq c_i$ with $c_i \in [0; d]$

Outline

Standard Reduction: Combinatorial Explosion

A New Construction

Conclusion

Our Approach

- Remark: from the time progress criterion in [AD94]:

$$\bigwedge_{x \in X} \text{unbounded}(x) \vee \text{fluctuating}(x)$$

A run is **Zeno** iff:

1. some $x \in X$ is **blocking**, i.e. bounded and never reset
2. or **time cannot elapse**: $\cdots \bullet \xrightarrow{x:=0} \bullet \rightarrow \bullet \xrightarrow{(x=0)} \bullet \cdots$

- Ideas:
  - constraining **all** accepting runs to be non-Zeno is **expensive**: only **one** of them is required
  - from (1) and (2), define **conditions** on **SCC** in $\mathrm{ZG}(A)$

Coming Next: A New Algorithm

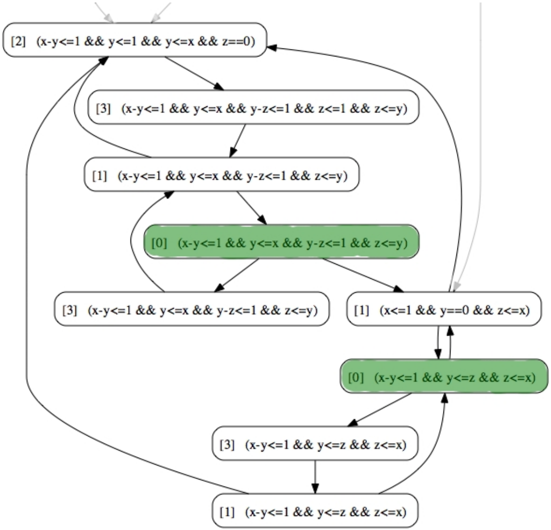
What we saw:

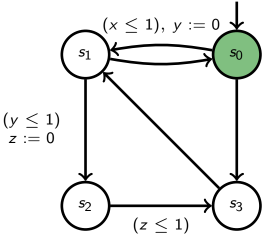
- $\mathrm{ZG}(A_n)$ has size $\mathcal{O}(n)$
- $\mathrm{ZG}(A'_n)$ has size $\mathcal{O}(2^n)$

What we propose:

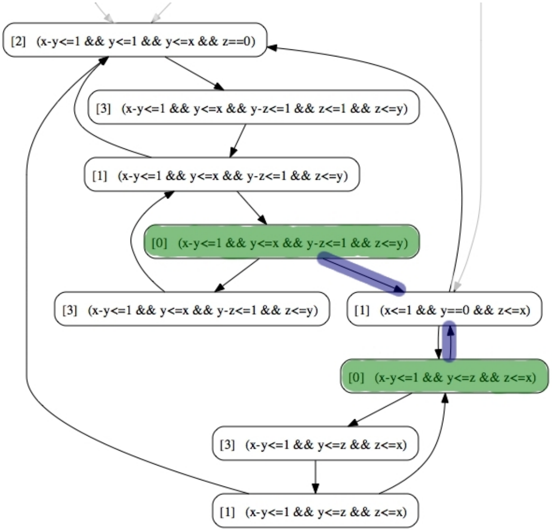
A $|\mathrm{ZG}(A_n)|.\mathcal{O}(n^2)$ algorithm

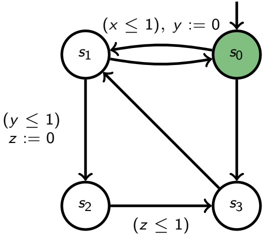
The Case of Blocking Clocks (no $x = 0$)



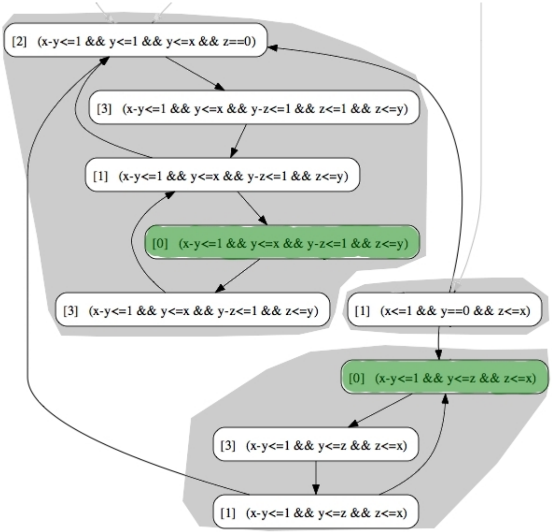


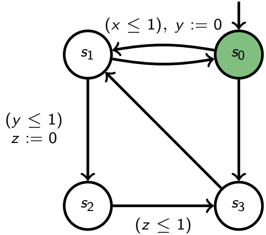
The Case of Blocking Clocks (no $x = 0$)



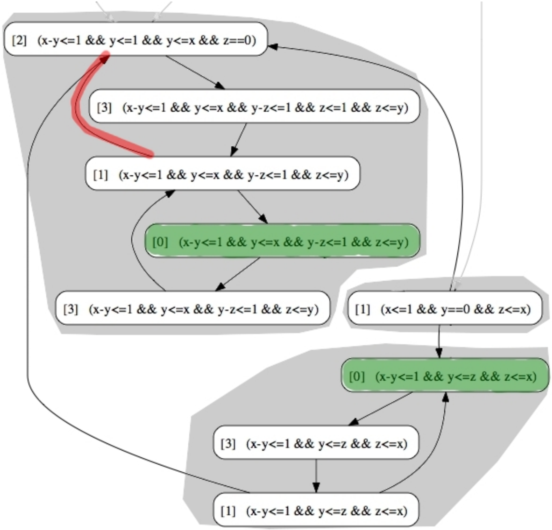


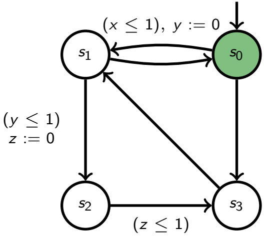
The Case of Blocking Clocks (no $x = 0$)



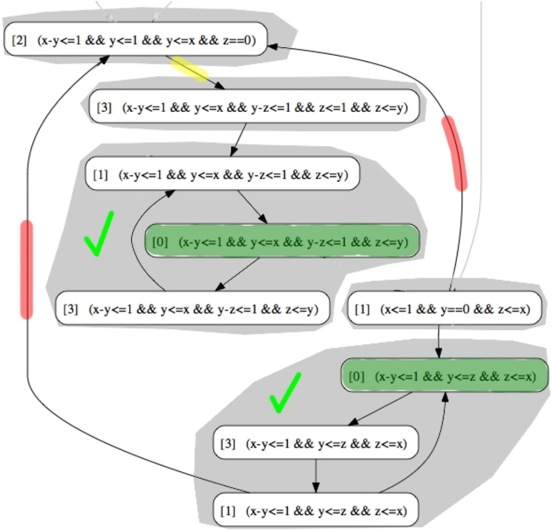


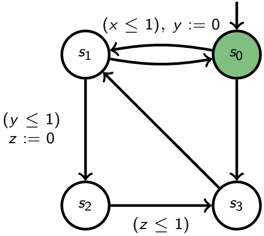
The Case of Blocking Clocks (no $x = 0$)



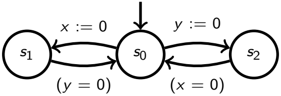


The Case of Blocking Clocks (no $x = 0$)



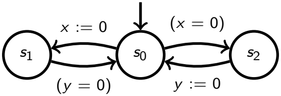


The Case of Zero Checks



$$s_0 \xrightarrow{x:=0} s_1 \xrightarrow{(y=0)} s_0 \xrightarrow{y:=0} s_2 \xrightarrow{(x=0)} s_0$$

All states are in the scope of a zero check!



$$s_0 \xrightarrow{x:=0} s_1 \xrightarrow{(y=0)} s_0 \xrightarrow{(x=0)} s_2 \xrightarrow{y:=0} s_0$$

State $\mathbf{s_2}$ is **clear**: all zero-checks are **preceded** by resets!

Given an SCC of ZG(A) does there exist a **clear node** ?

The Case of Zero Checks

Idea: extend nodes in $\mathsf{ZG}(A)$ with a set of clocks that we **guess** will be **checked for** 0

For each node in $\mathsf{ZG}(A)$, $2^{|X|}$ extended nodes!

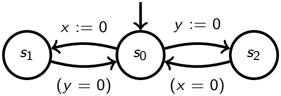
Lemma

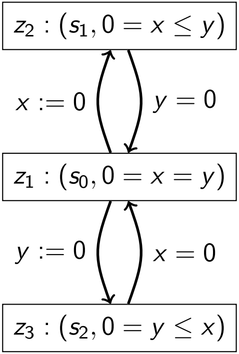
In every reachable node (q, Z) in $\mathsf{ZG}(A)$, clocks are **totally ordered**

Corollary

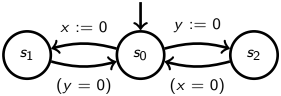
For every **reachable** (q, Z), it is **sufficient** to consider only $|\mathbf{X}| + \mathbf{1}$ guess sets

The Case of Zero Checks (1st example)





The Case of Zero Checks (1st example)

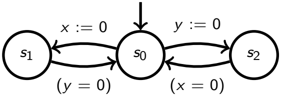


$\boxed{z_2 : (s_1, 0 = x \leq y), \emptyset}$ $\boxed{z_2, \{x\}}$ $\boxed{z_2, \{x, y\}}$

$\boxed{z_1 : (s_0, 0 = x = y), \emptyset}$ $\boxed{z_1, \{x, y\}}$

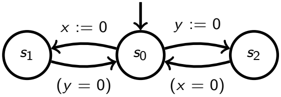
$\boxed{z_3 : (s_2, 0 = y \leq x), \emptyset}$ $\boxed{z_3, \{y\}}$ $\boxed{z_3, \{x, y\}}$

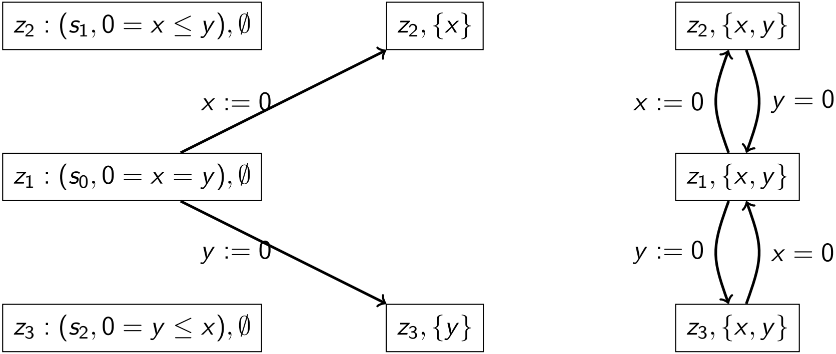
The Case of Zero Checks (1st example)



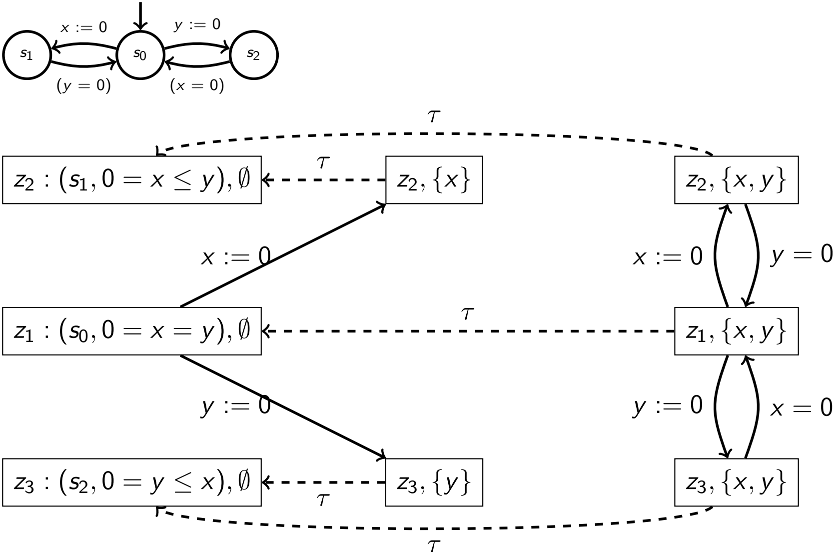
$$\begin{tikzcd}
\boxed{z_2 : (s_1, 0 = x \leq y), \emptyset} & \boxed{z_2, \{x\}} & \boxed{z_2, \{x, y\}} \arrow[d, bend left, "y = 0"] \\
\boxed{z_1 : (s_0, 0 = x = y), \emptyset} & & \boxed{z_1, \{x, y\}} \\
\boxed{z_3 : (s_2, 0 = y \leq x), \emptyset} & \boxed{z_3, \{y\}} & \boxed{z_3, \{x, y\}} \arrow[u, bend right, "x = 0"']
\end{tikzcd}$$

The Case of Zero Checks (1st example)

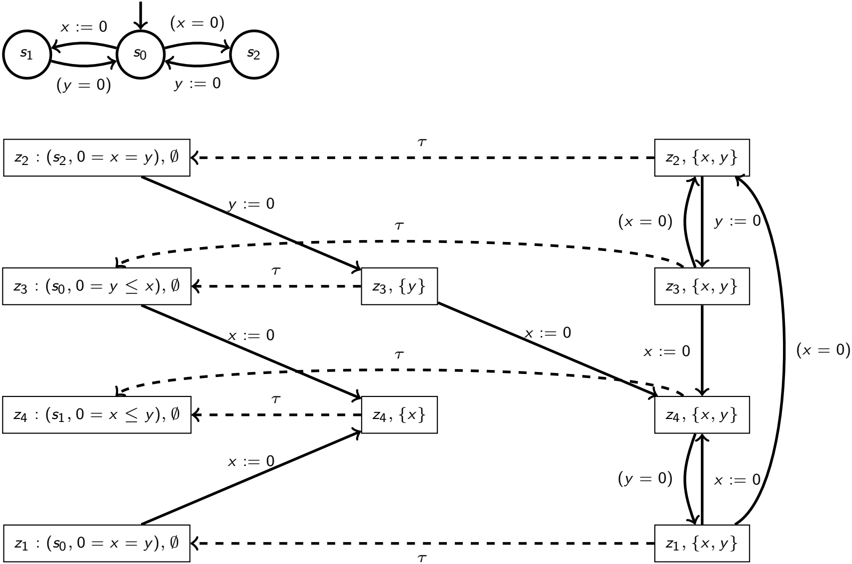




The Case of Zero Checks (1st example)



The Case of Zero Checks (2nd example)



A Global Algorithm

Lemma

A TBA A has a non-Zeno accepting run iff GZG(A) has an **SCC** that:

- contains an **accepting node** and,
- contains a **clear node** $(q, Z, \emptyset)$ and,
- has **no blocking** clock.

Theorem

The existence of such an SCC can be decided in time $|\mathrm{ZG}(A)|.\mathcal{O}(|X|^2)$

- A $|\mathrm{GZG}(A)|.\mathcal{O}(|X|)$ algorithm over graph GZG(A) of size $|\mathrm{ZG}(A)|.\mathcal{O}(|X|)$

Outline

Benchmarks

A	ZG(A)	ZG(A')		GZG(A)		
	size	size	otf	size	otf	opt
Train-Gate2 (mutex)	134	194	194	400	400	**134**
Train-Gate2 (bound. resp.)	988	**227482**	352	3840	1137	292
Train-Gate2 (liveness)	100	217	35	298	53	33
Fischer3 (mutex)	1837	3859	3859	7292	7292	**1837**
Fischer4 (mutex)	46129	96913	96913	229058	229058	**46129**
Fischer3 (liveness)	1315	4962	52	5222	64	40
Fischer4 (liveness)	33577	147167	223	166778	331	207
FDDI3 (liveness)	508	1305	44	3654	79	42
FDDI5 (liveness)	6006	15030	90	67819	169	88
FDDI3 (bound. resp.)	6252	41746	59	52242	114	60
CSMA/CD4 (collision)	4253	7588	7588	20146	20146	**4253**
CSMA/CD5 (collision)	45527	80776	80776	260026	260026	**45527**
CSMA/CD4 (liveness)	3038	9576	1480	14388	3075	832
CSMA/CD5 (liveness)	32751	120166	8437	186744	21038	4841

- Combinatorial explosion may **occur**
- **Optimized** use of GZG(A) (to appear at ATVA 2010)

Conclusions & Perspectives

- **Combinatorial explosion** occurs due to the **strongly non-Zeno** constructions from [AM04, TYB05]
- A $|\mathrm{ZG}(A)|.\mathcal{O}(|X|^2)$ algorithm for TBA emptiness that:
  - encodes **fluctuating** condition as a **Büchi condition**
  - and disables transitions with **blocking clocks**
- Application to the computation of **non-Zeno** strategies for Timed Games

Bibliography

R. Alur and D.L. Dill.
A theory of timed automata.
Theoretical Computer Science, 126(2):183–235, 1994.

R. Alur and P. Madhusudan.
Decision problems for timed automata: A survey.
In *SFM-RT'04*, volume 3185 of *LNCS*, pages 1–24, 2004.

H. Bowman and R. Gómez.
How to stop time stopping.
Formal Asp. Comput., 18(4):459–493, 2006.

P. Bouyer.
Forward analysis of updatable timed automata.
Formal Methods in System Design, 24(3):281–320, 2004.

R. Gómez and H. Bowman.
Efficient detection of zeno runs in timed automata.
In *Proc. 5th Int. Conf. on Formal Modeling and Analysis of Timed Systems, FORMATS 2007*, volume 4763 of *LNCS*, pages 195–210, 2007.

S. Tripakis.
Verifying progress in timed systems.
In *Proc. 5th Int. AMAST Workshop, ARTS'99*, volume 1601 of *LNCS*, pages 299–314. Springer, 1999.

S. Tripakis.
Checking timed büchi emptiness on simulation graphs.
ACM Transactions on Computational Logic, 10(3):??–??, 2009.

S. Tripakis, S. Yovine, and A. Bouajjani.
Checking timed büchi automata emptiness efficiently.
Formal Methods in System Design, 26(3):267–292, 2005.