Reachability in Updatable Timed Automata made Faster and More Effective

Paul Gastin
LSV, ENS Paris-Saclay
France

Sayan Mukherjee, B. Srivathsan
Chennai Mathematical Institute,
India

ANR TickTac Meeting
November 20, 2020

Overview

-1. Updatable Timed automata, reachability

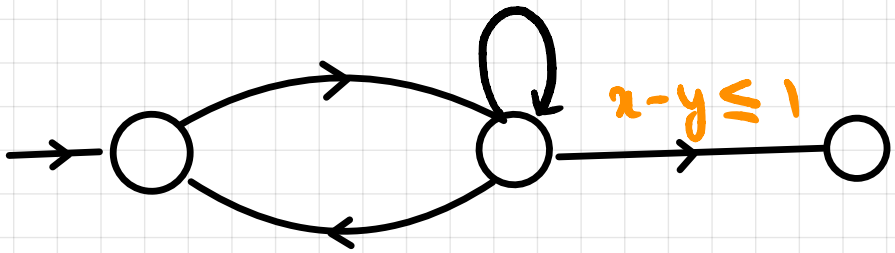
-2. Existing zone-based algorithm

-3. Our modification

-4. Impact

Updatable Timed Automata

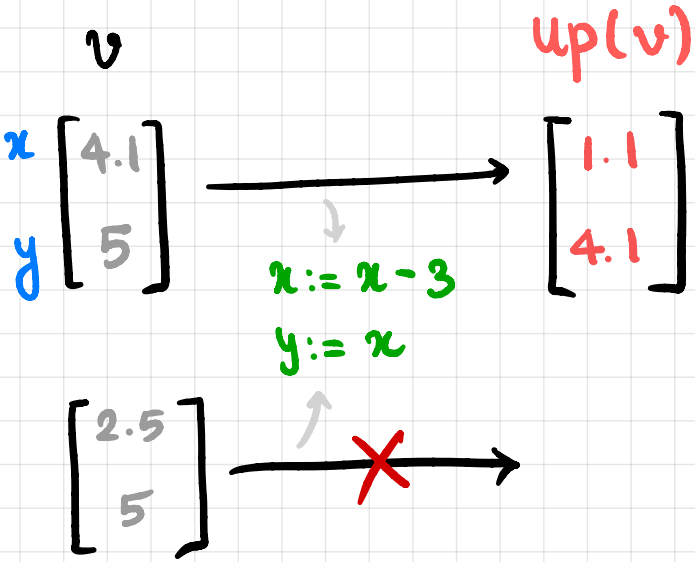
Bouyer, Dufourd, Fleury, Petit (TCS '04)



Guards

$$\varphi := x \lhd c \mid c \lhd x \mid x-y \lhd c \mid c \lhd x-y \mid \varphi \wedge \varphi$$

$c \in \mathbb{N}$ $\quad$ $\lhd \in \{\leq, <\}$

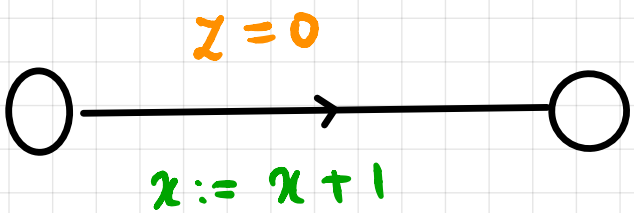


Updates

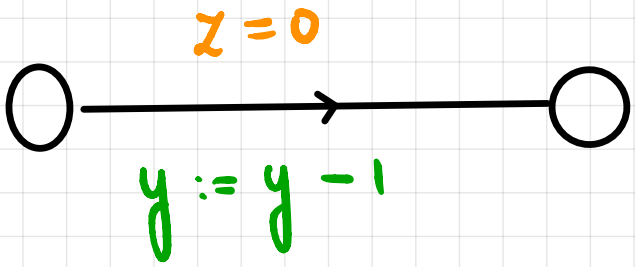
$$x := c \mid x := y + c \mid x := y - c$$

Reachability is undecidable for UTA

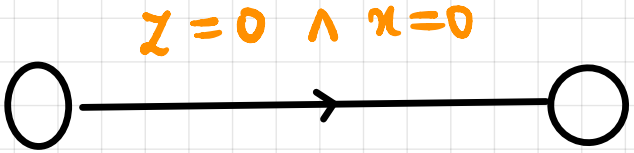
Since Counters can be simulated using updates



increment (c_1)

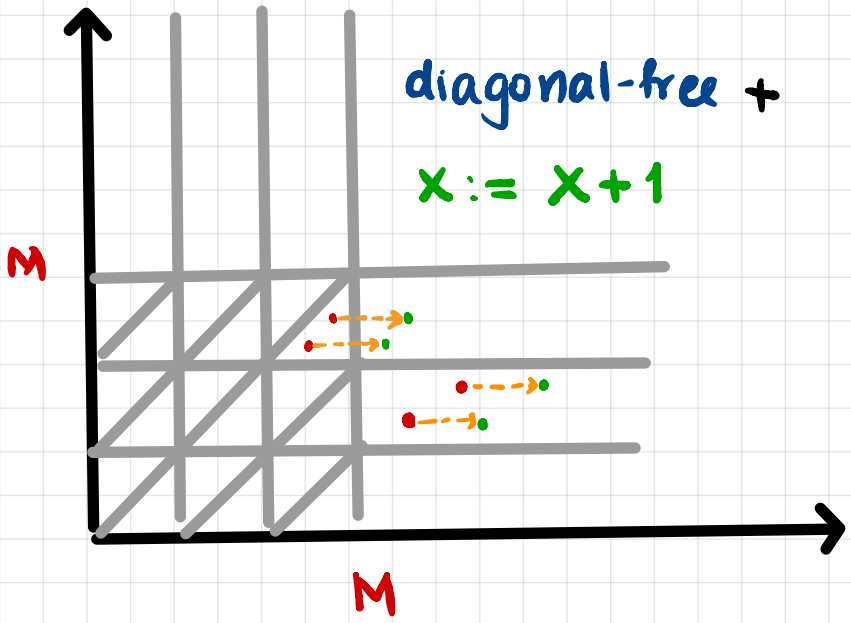


decrement (c_2)

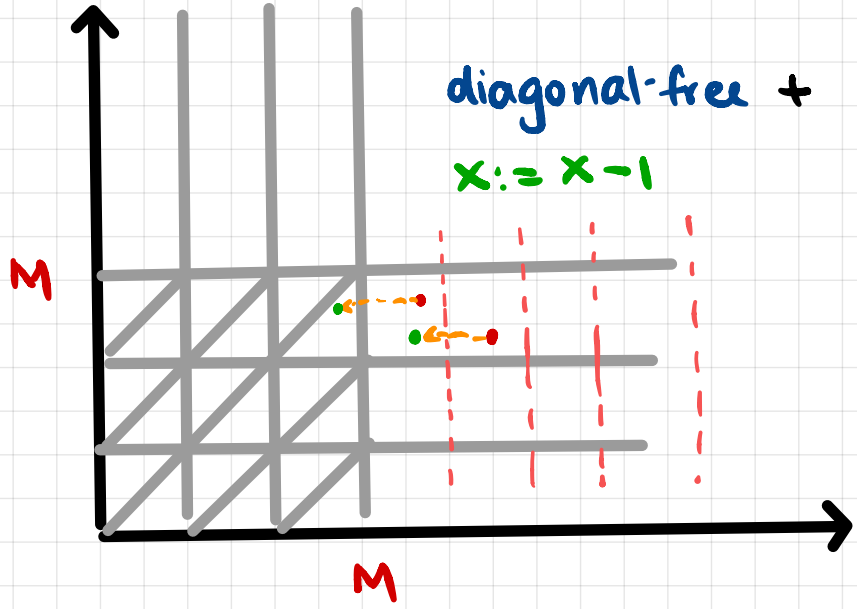


is Counter (c_1) = 0?

Decidable subclasses are based on region equivalence



Region-compatible



Not region-compatible

	Diagonal-free	Diagonals
$x := c$, $x := y$	Decidable	Decidable
$x := x + c$	Decidable	Undecidable
$x := x - 1$	Undecidable	Undecidable

Bouyer, Dufourd, Fleury, Petit (TCS '04)

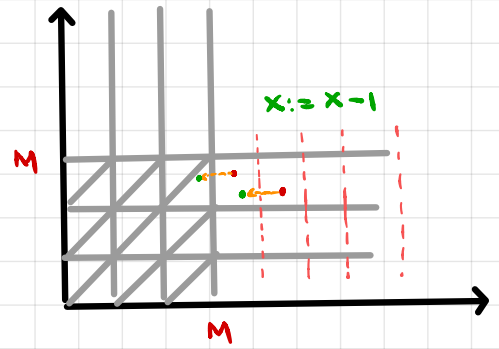
Timed automata with bounded subtraction

Fersman, Krcál, Pettersson, Yi (Inf. Comp.'07)

Updates: $x := c$ and $x := x - c$

Every transition with $x := x - c$ should have some guard $x \leq d$

$\therefore$ avoids problem of



Not region-compatible

UPPAAL-TIMES: Tool for Schedulability analysis using this class

Overview

-1. Updatable Timed automata, reachability ✓

-2. Existing zone-based algorithm

-3. Our modification

-4. Impact

Zones

location

Zone

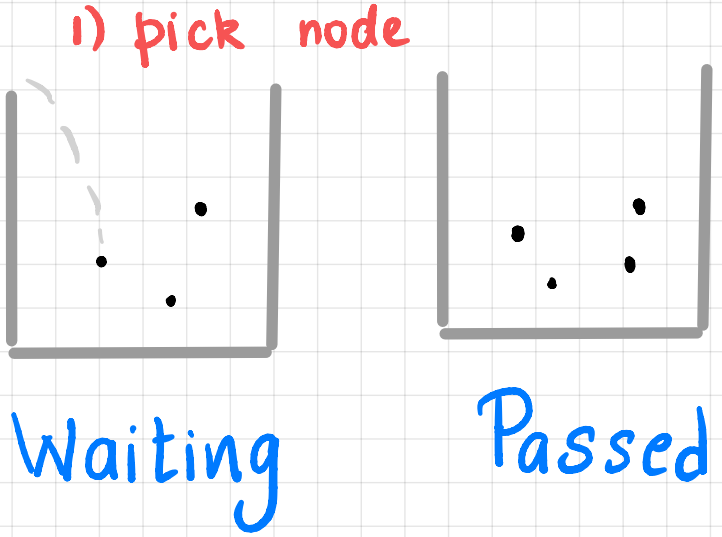
(p, Z_p) : a set of configurations

$\downarrow$

(q, Z_q) : successor via $p \xrightarrow[up]{g} q$

$$Z_q = \overrightarrow{up(Z_p \cap g)}$$

Zone Enumeration



(p, Z_p)

↓

2) compute successors (q, Z_q)

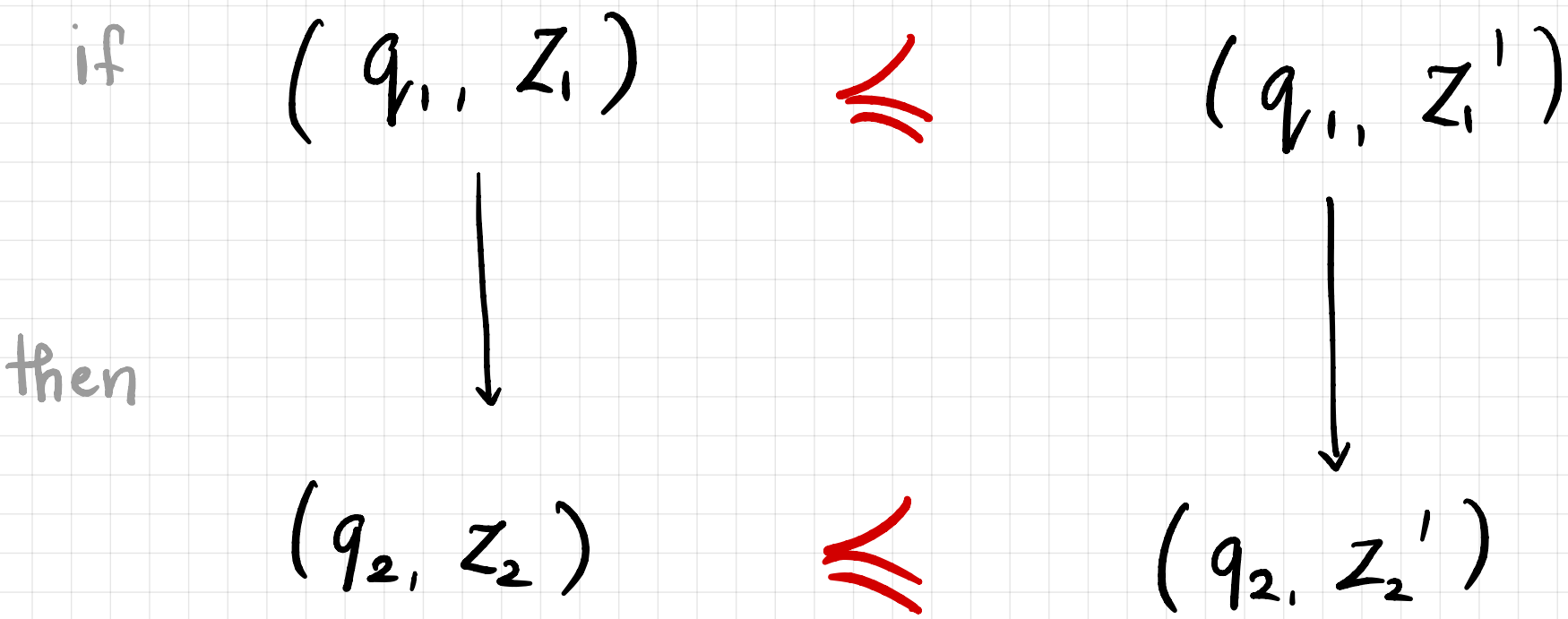
(q, Z_q) can be **discarded** if $\exists$ $(q, Z_q') \in$ Waiting $\cup$ Passed

s.t.

$$Z_q \leq Z_q'$$

3) check simulation

Simulations



Simulations preserve paths

GOAL

Find simulations that are

finite and efficient

Coming next: $\mathcal{G}$-simulation [Gastin, Mukherjee, S. (CAV'19)]

$$(q, Z) \preccurlyeq (q, Z')$$

if

$$\forall v \in Z \qquad \exists v' \in Z'$$

s.t.

$$(q, \underline{v}) \preccurlyeq (q, \underline{v'})$$

G-preorder

G : a set of atomic constraints

$$v \preccurlyeq_G v' \quad \text{if}$$

$$\forall \varphi \in G.\ \forall \delta \geq 0 \quad v + \delta \models \varphi \Rightarrow v' + \delta \models \varphi$$

$$\forall \varphi \in G.\ \forall \delta \geq 0 \quad v + \delta \models \varphi \Rightarrow v' + \delta \models \varphi$$

Examples:

$x = 4.2 \quad \preceq_{\{x \geq 5\}} \quad x = 4.9$

$x = 4.2 \quad \not\preceq_{\{x \geq 5\}} \quad x = 4.1$

$x = 4.2 \quad \not\preceq_{\{x \leq 5\}} \quad x = 4.9$

$x = 4.2 \quad \preceq_{\{x \leq 5\}} \quad x = 4.1$

$\langle x = 5,\ y = 10 \rangle \quad \preceq_{\{x - y \leq 2\}} \quad \langle x = 2,\ y = 0 \rangle$

Next Task:

Get a simulation on the UTA configurations based on G- preorder

Associate a set of constraints $G(q)$ to every q

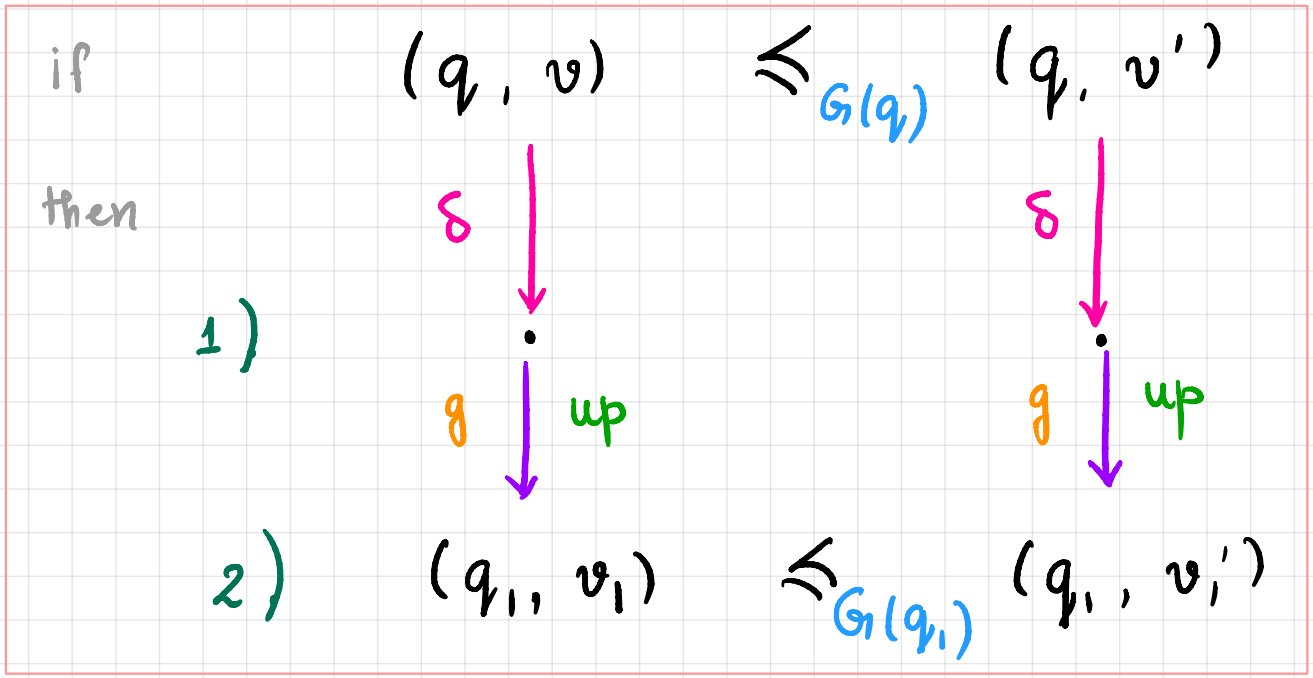
if $(q, v) \preceq_{G(q)} (q, v')$

then

1) $(q, v) \xrightarrow{\delta} \cdot \xrightarrow{g\ up} (q_1, v_1)$

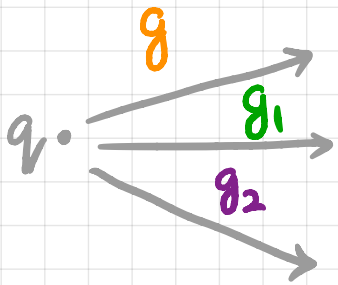
2) $(q, v') \xrightarrow{\delta} \cdot \xrightarrow{g\ up} (q_1, v_1')$

$(q_1, v_1) \preceq_{G(q_1)} (q_1, v_1')$

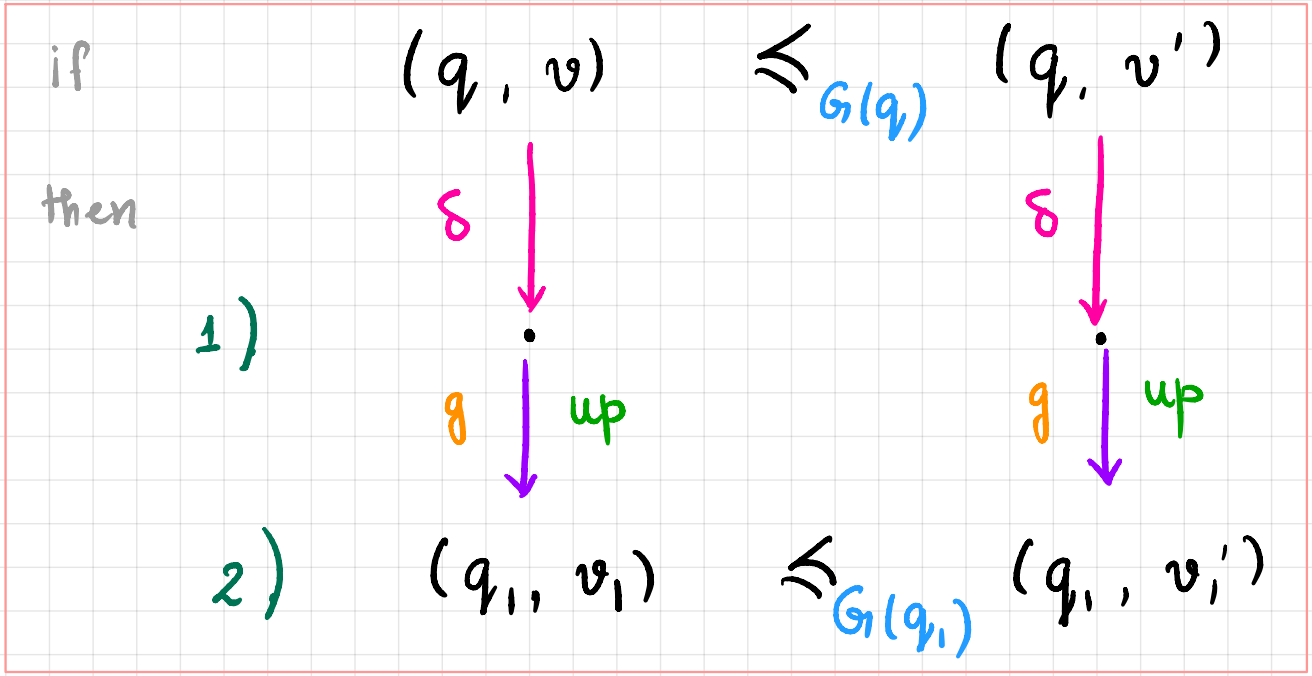


1): Every guard that v satisfies should be satisfied by v'

Add guards in outgoing transitions of q to $G(q)$



$\{ g, g_1, g_2 \} \subseteq G(q)$



2) Simulation is preserved after an update

How to ensure this ?

if $(q, v) \preceq_{G(q)} (q, v')$

$g \downarrow$ $\quad$ $g \downarrow$ (No update)

then $(q_1, v_1) \preceq_{\{x-y \le 1\}} (q_1, v_1')$

$\therefore \quad v_1 = v$

$v_1' = v'$

To ensure $v_1 \preceq_{\{x-y \le 1\}} v_1'$, add $x - y \le 1$ to $G(q)$

---

if $(q, v) \preceq_{G(q)} (q, v')$

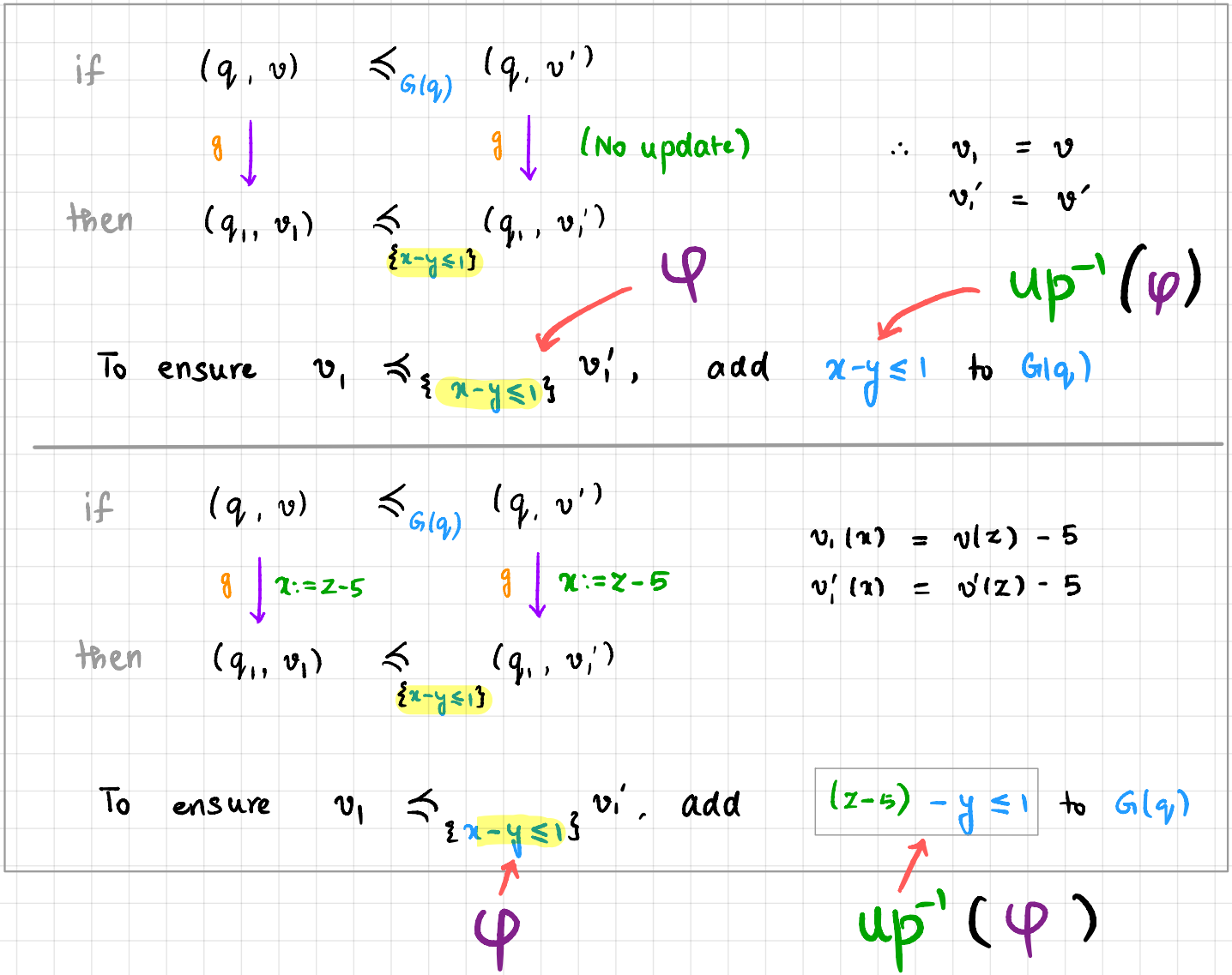
$g \downarrow x := z-5$ $\quad$ $g \downarrow x := z-5$

then $(q_1, v_1) \preceq_{\{x-y \le 1\}} (q_1, v_1')$

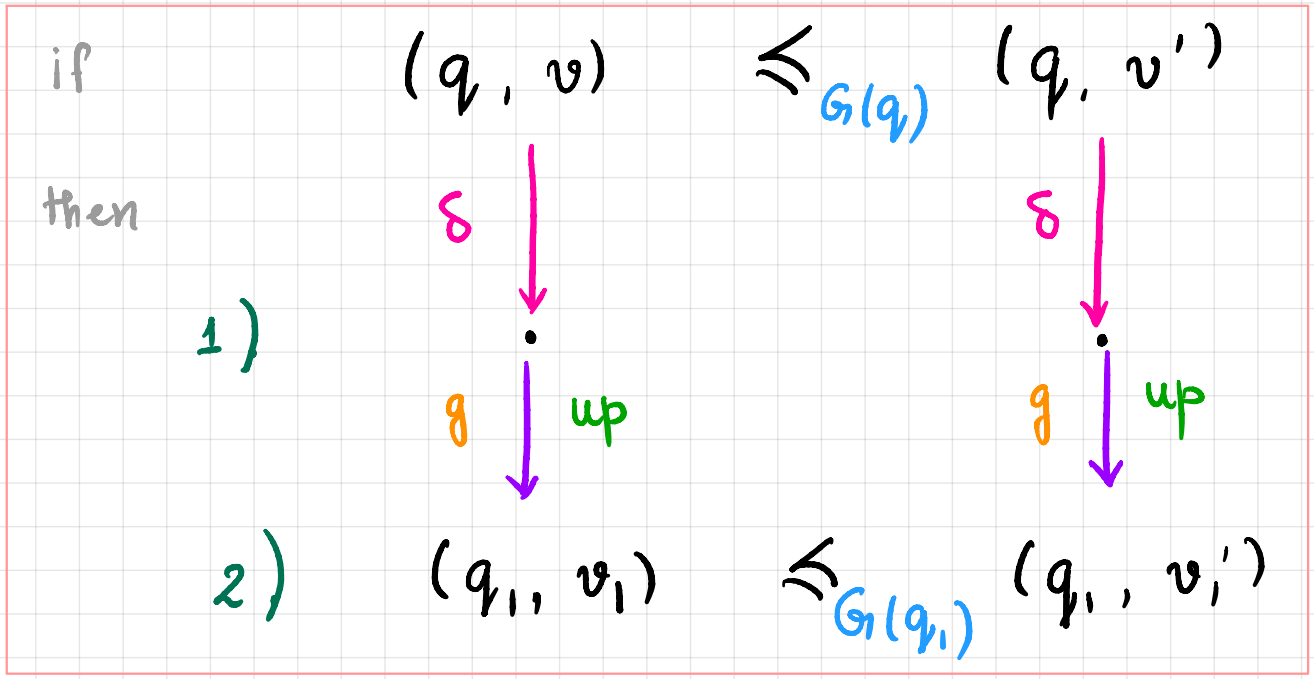
$v_1(x) = v(z) - 5$

$v_1'(x) = v'(z) - 5$

To ensure $v_1 \preceq_{\{x-y \le 1\}} v_1'$, add $(z-5) - y \le 1$ to $G(q)$



$$up^{-1}(\varphi) = \varphi[x \mapsto up(x)]$$



2) Simulation is preserved after an update

$$\forall \varphi \in G(q_1), \text{ add } up^{-1}(\varphi) \text{ to } G(q)$$

$$up^{-1}(\varphi) = \varphi[x \mapsto up(x)]$$

STATIC ANALYSIS

1) $\forall q$: Add guards in outgoing transitions of q to $G(q)$

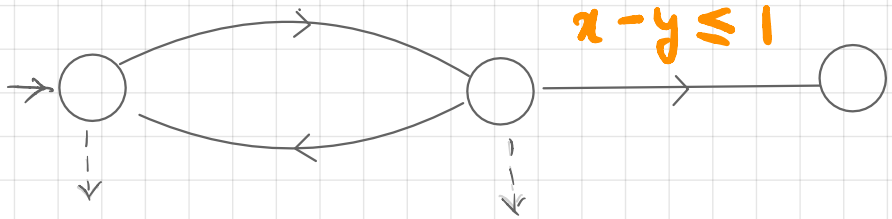
2) Repeat : for every $q \xrightarrow[up]{g} q_1$

$\forall \varphi \in G(q_1)$, add $up^{-1}(\varphi)$ to $G(q)$

Until fixpoint

$1 \le x \le 3$

$x := x - 1$



G^0 $\quad 1 \le x,\ x \le 3$ $\qquad x - y \le 1$

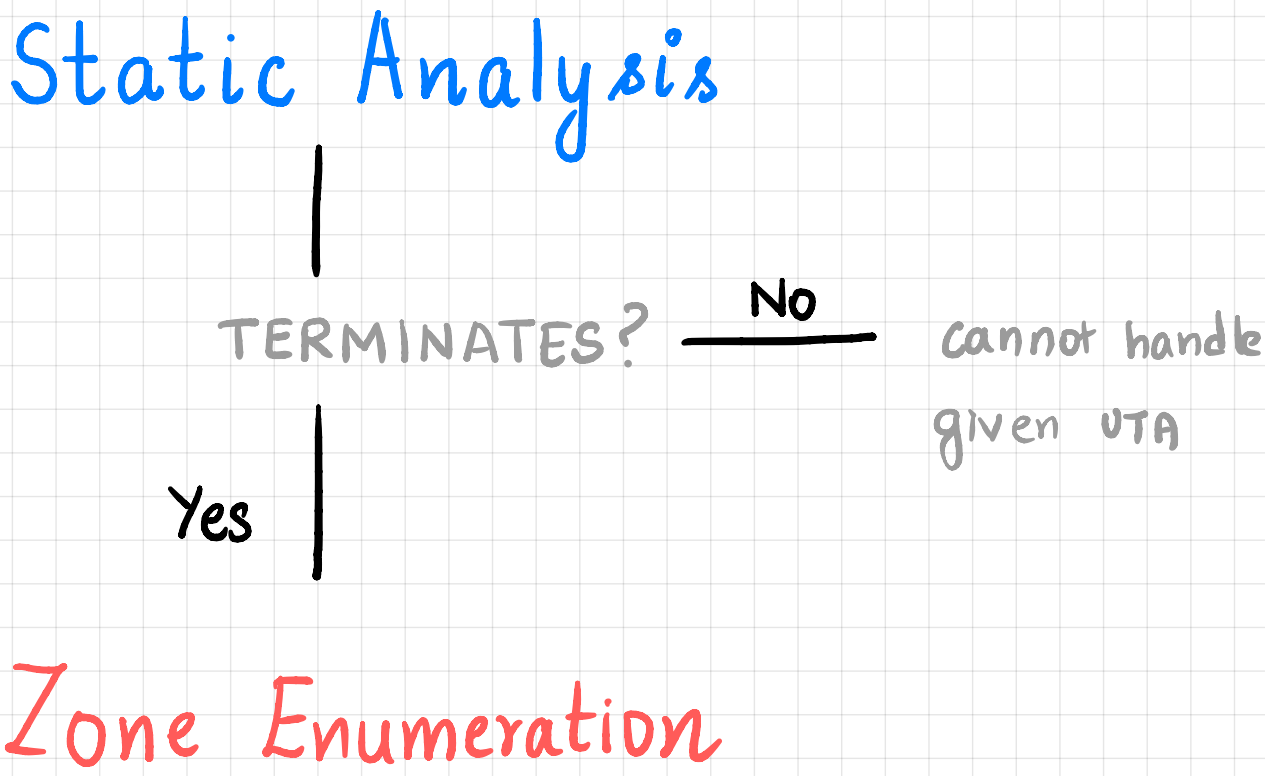
G^1 $\quad x - y \le 2$ $\qquad 1 \le x,\ x \le 3$

G^2 $\quad 2 \le x,\ x \le 4$ $\qquad x - y \le 2$

G^3 $\quad x - y \le 3$ $\qquad 2 \le x,\ x \le 4$

⋮ $\qquad$ ⋮

REACHABILITY ALGORITHM FOR UTA



Static Analysis

TERMINATES? — No — cannot handle given UTA

Yes

Zone Enumeration

CAV '19:

- Algo to detect whether static analysis terminates $O(|Q| \cdot |X|^2)$
- Algo for $Z \leq_G Z'$ $O(2^d |X|^2)$ d: # diagonals in G

- Works for subclasses of Bouyer et al.
- does not work for T.A. with bounded subtraction

Overview

-1. Updatable Timed automata, reachability ✓

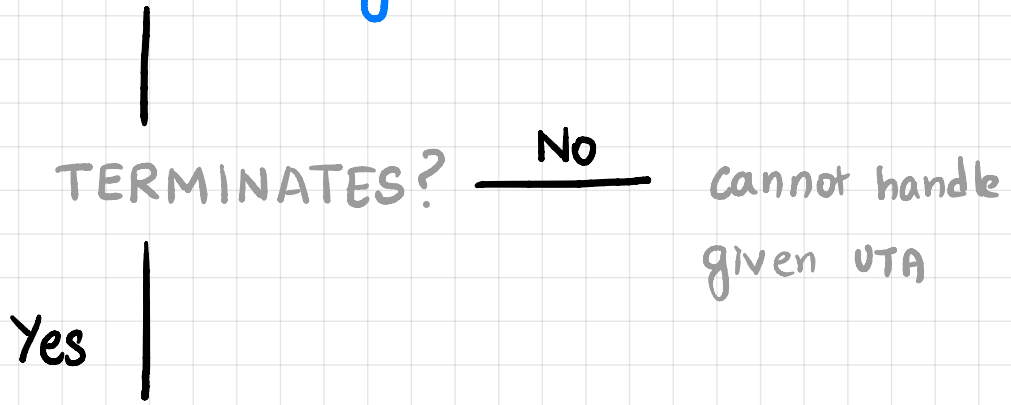
-2. Existing zone-based algorithm ✓

-3. Our modification

-4. Impact

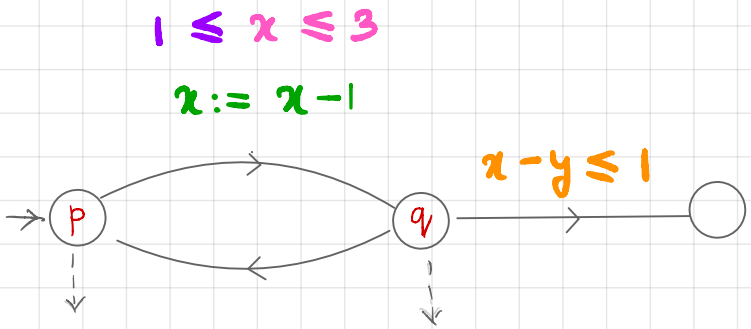
RESULTS

New Static Analysis



Zone Enumeration

New: Reduced propagation which terminates for more UTA, including T.A. with bounded subtraction



G^0 $1 \le x,\ x \le 3$ $\quad$ $x - y \le 1$

G^1 $x - y \le 2$ $\quad$ $1 \le x,\ x \le 3$

G^2 $2 \le x,\ x \le 4$ $\quad$ $x - y \le 2$

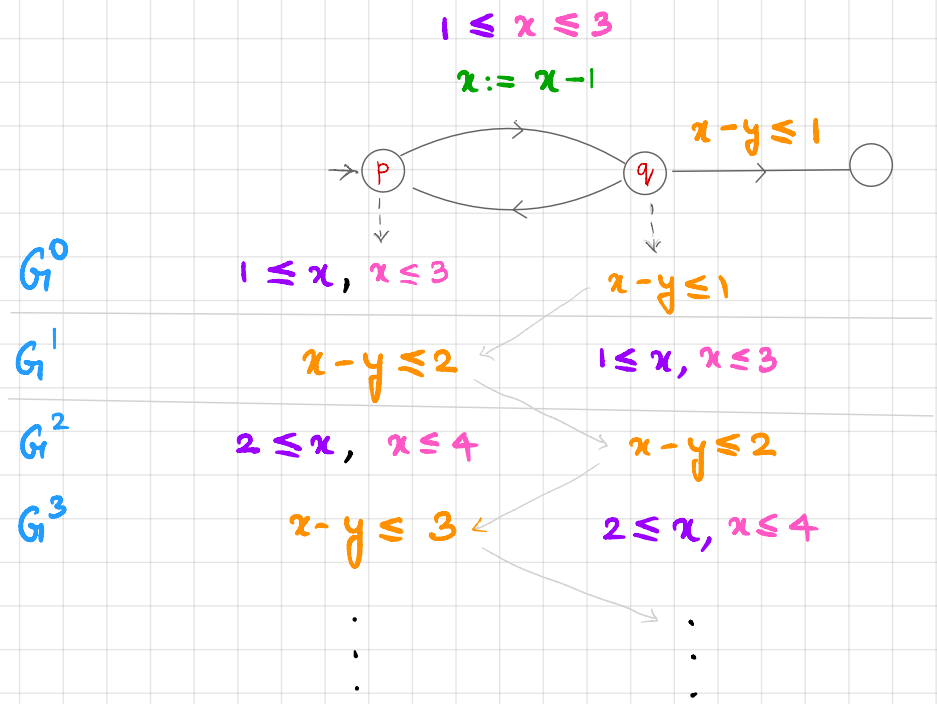
G^3 $x - y \le 3$ $\quad$ $2 \le x,\ x \le 4$

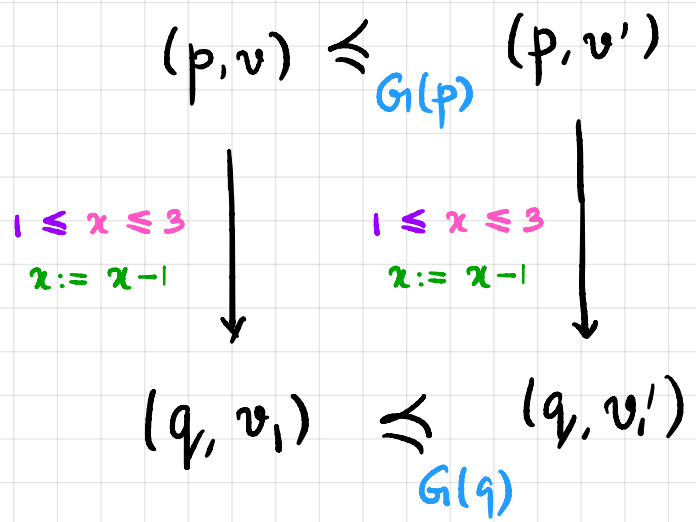
$\vdots$ $\quad$ $\vdots$

$(p, v) \preceq_{G(p)} (p, v')$

$1 \le x \le 3$, $x := x - 1$ ↓ $\quad$ $1 \le x \le 3$, $x := x - 1$ ↓

$(q, v_1) \preceq_{G(q)} (q, v_1')$



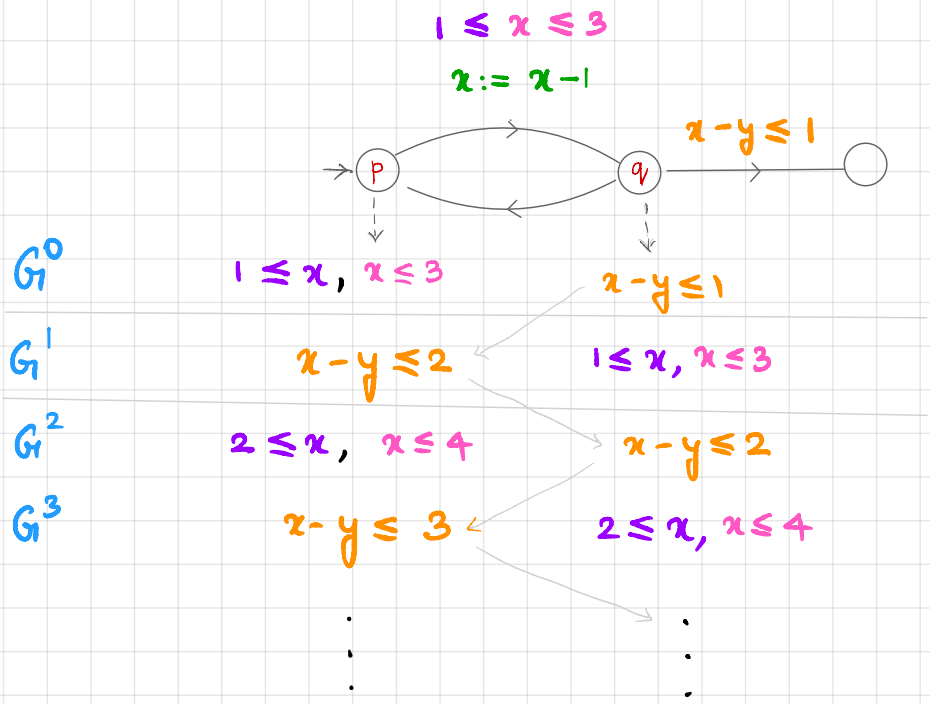


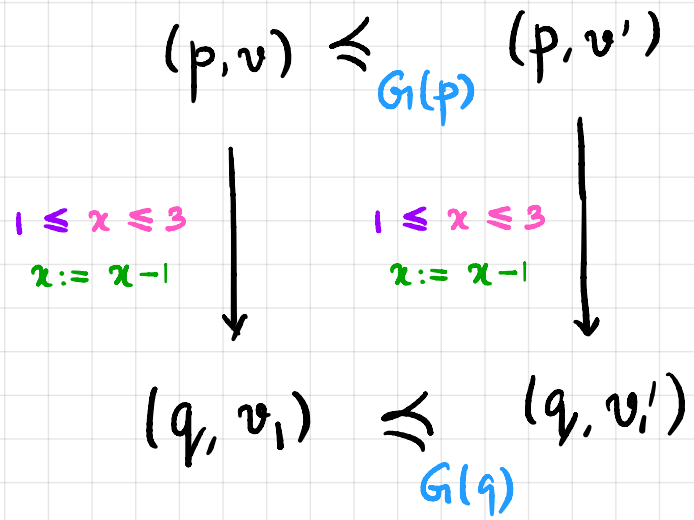
NOTICE: Relevant v, v' satisfy $x \leq 3$

HENCE: $v \preccurlyeq_{\{x \leq 3\}} v' \Rightarrow v'(x) \leq v(x)$

$\Rightarrow v \preccurlyeq_{\{x \leq c\}} v' \quad \forall c$

Adding $x \leq 4, x \leq 5, \ldots$ to $G(p)$ is **unnecessary**



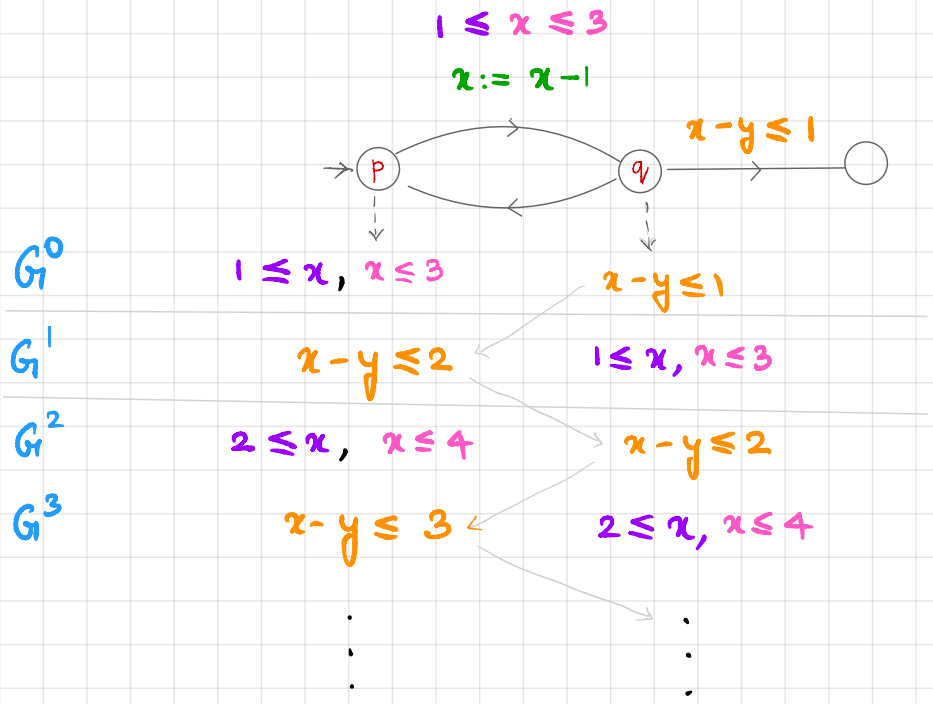


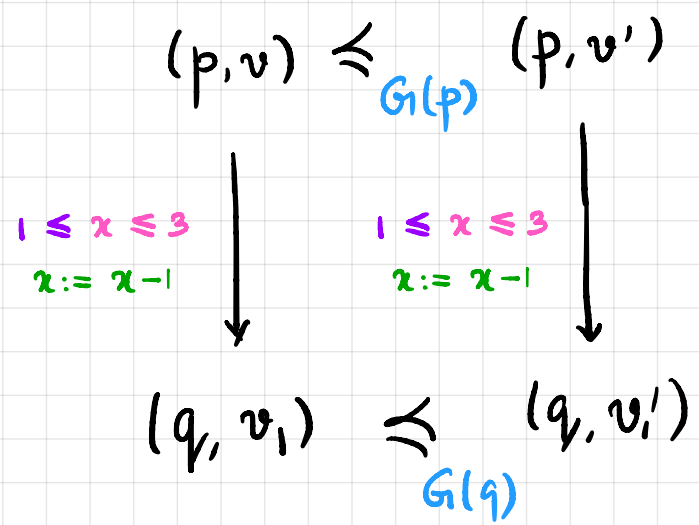
NOTICE: Relevant v, v' satisfy $x \leq 3$

HENCE: $v \preccurlyeq_{\{x \geq 3\}} v' \Rightarrow v(x) \leq v'(x)$

$\Rightarrow v \preccurlyeq_{\{x \geq c\}} v'$ $\forall c$

Adding $x \geq 4$, $x \geq 5$, ... to $G(p)$ is **unnecessary**



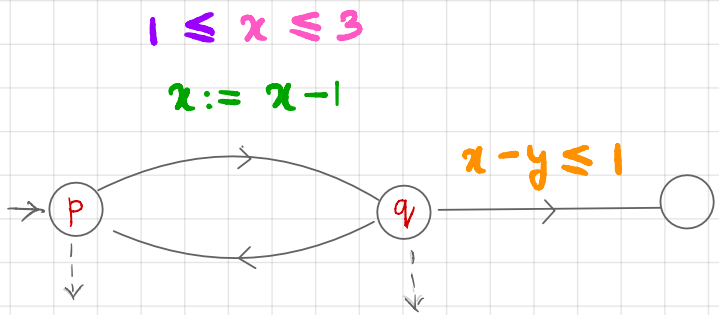


NOTICE: Relevant v, v' satisfy $x \leq 3$

HENCE: Both v, v' satisfy $x-y \leq 3,\ x-y \leq 4, \ldots$

$$v \preceq_{\{x-y \leq c\}} v' \qquad \forall c \geq 3 \text{ is already true}$$

Adding $x-y \leq 3, x-y \leq 4, \ldots$ to $G(p)$ is **unnecessary**



G^0 $1 \le x,\ x \le 3$ $\quad x - y \le 1$

G^1 $x - y \le 2$ $\quad 1 \le x,\ x \le 3$

G^2 $2 \le x,\ x \le 4$ $\quad x - y \le 2$

G^3 $\quad 2 \le x,\ x \le 4$

G^4 $3 \le x$

G^5 $\quad 3 \le x$

$(p, v) \preceq_{G(p)} (p, v')$

$1 \le x \le 3$, $x := x - 1$ ↓ ↓ $1 \le x \le 3$, $x := x - 1$

$(q, v_1) \preceq_{G(q)} (q, v_1')$

Finite!

NEW STATIC ANALYSIS

1) $\forall q$: Add guards in outgoing transitions of q to $G(q)$

2) Repeat: for every $q \xrightarrow[up]{g} q_1$

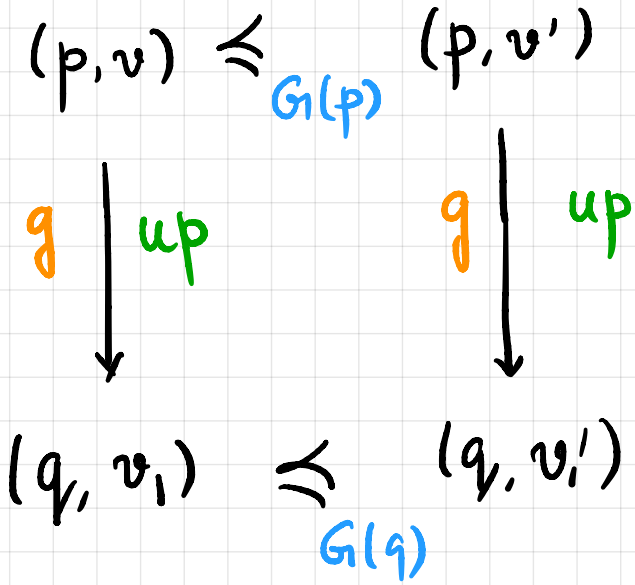
$\forall \varphi \in G(q_1)$, add ~~$up^{-1}(\varphi)$~~ $pre(g, up, \varphi)$ to $G(q)$

Until fixpoint

$up^{-1}(\varphi)$	g contains	$pre(g, up, \varphi)$
$x \lhd d$	$x \lhd c$	$\top$
$d \lhd x$	$x \lhd c$ $c < d$	$c \leq x$
$x - y \lhd d$	$x \lhd c$ $c < d$	$\top$

few more cases in the paper

MAIN IDEA



Choose $G(p)$ and $G(q)$ s.t.

Previously:

$$v \preccurlyeq_{G(p)} v' \implies v_1 \preccurlyeq_{G(q)} v_1'$$

Now:

$$v \preccurlyeq_{G(p)} v' \wedge v \vdash g \implies v_1 \preccurlyeq_{G(q)} v_1'$$

$up^{-1}(\varphi)$	g contains	$pre(g, up, \varphi)$
$x \lhd d$	$x \lhd c$	$\top$
$d \lhd x$	$x \lhd c$ $c < d$	$c \leqslant x$
$x - y \lhd d$	$x \lhd c$ $c < d$	$\top$

Remark: Similar optimization appears in the context of M-extrapolations in the first static analysis paper of:

Behrmann, Bouyer, Fleury, Larsen (TACAS'03)

- diagonal constraints are not considered
- Updates $x := y + c$, and no $x := y - c$
- Going from M to LU or G is not direct

Detecting Termination

New static analysis does not terminate iff

a constraint with a constant $\geq N$ is added

$N = \text{poly}(M, L, |Q|, |X|)$

M: largest guard constant

L: largest update constant

Complexity of static analysis:

poly ($|\mathcal{A}|$) : if constants encoded in unary

PSPACE-complete : if constants encoded in binary

For PSPACE-hardness:

emptiness of bounded 1-counter automata $\longrightarrow$ termination of new static analysis

Fearnley, Jurdzinski (ICALP'15)

Overview

-1. Updatable Timed automata, reachability ✓

-2. Existing zone-based algorithm ✓

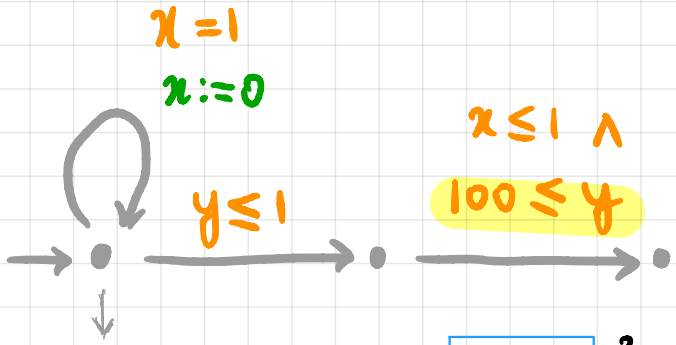
-3. Our modification ✓

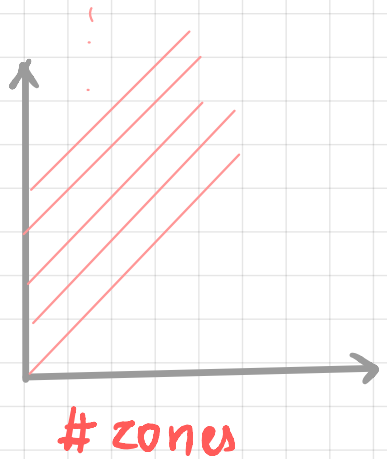
-4. Impact

FASTER

New analysis generates fewer constraints

⇒ more simulations during zone enumeration





New G: { x=1, y≤1, x≤1, 1≤y } → 4

Old G: { x=1, y≤1, x≤1, 100≤y } → 104

[& TCHECKER & UPPAAL]

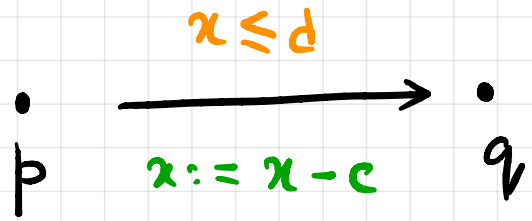
More impact in the presence of diagonal constraints since fewer diagonals implies faster $Z \preceq_G Z'$

More Effective

Termination guaranteed for Timed automata with bounded subtraction

Updates: $x := c$ and $x := x - c$

Every transition with $x := x - c$ should have some guard $x \leq d$



Constants in $G(p)$ are at most d

UTA with bounded clocks:

- decidable, since only finitely many zones
- How to make use of simulations?

Adding guard $\xrightarrow{\bigwedge_x x \leq B}$ guarantees termination of new static analysis

EDF SCHEDULING

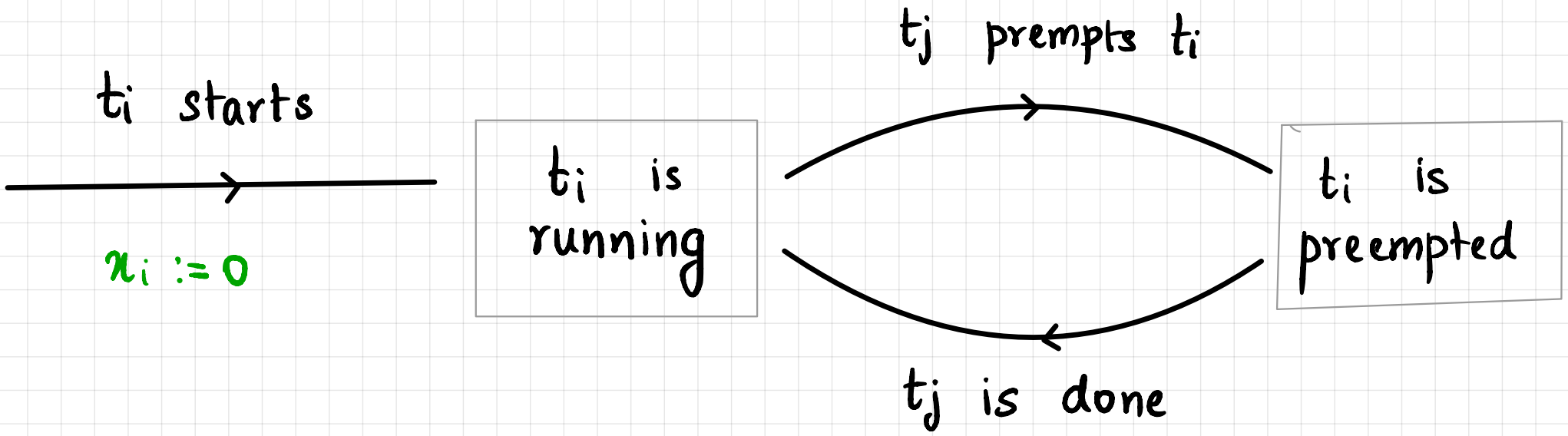
Task : (computation time C , deadline D)

Task release : periodic or by a T.A. specification

Preemption allowed

Problem: can all tasks be scheduled within deadline?

Modeling preemption: (Fersman et al.'07)



$$x_i = x_i - C_j$$

$$x_i \leq D_j$$

Modeling Earliest - Deadline - First (EDF) scheduling:

> Whenever choice between t_i and t_j, choose the one closest to its deadline

When task `t` starts, reset a clock `d`

t_i prioritized over t_j if: $D_i - d_i < D_j - d_j$

EXPERIMENTS

- New static analysis implemented in TCHECKER

- Benchmarks: EDF scheduling with preemption for different task release strategies

This application requires both diagonal constraints and subtractions

CAV'19 ↓

		New static analysis		Static analysis of [16]	
Model	Schedulable?	# nodes	time	# nodes	time
SporadicPeriodic-5	Yes	677	1.710s	-	-
SporadicPeriodic-20	No	852	1.742s	-	-
Mine-Pump	Yes	31352	7m 23.509s	-	-
Flower task triggering automaton: (computation time, deadline)				CANNOT HANDLE	
(1,2), (1,2), (1,2)	No	212	0.057s	-	-
(1,10), (1,10), (1,10), (1,4)	Yes	105242	8m 57.256s	-	-
Worst-case task triggering automaton: (computation time, deadline)					
(1,2), (1,2), (1,2)	No	20	0.050s	-	-
(1,10), (1,10), (1,10), (1,4)	Yes	429	0.454s	-	-
12 copies of (1,20)	Yes	786	12m 5.250s	-	-
$\mathcal{A}_{gain} \times 3$	N/A	24389	7.611s	24389	12.402s
$\mathcal{A}_{gain} \times 4$	N/A	707281	14m 12.369s	707281	27m 13.540s

Sporadic Periodic : TIMES tool

Mine - pump: Gerdsmeier et al. (CATS'01)

Others : Synthetic

CONCLUSION

-1. Updatable Timed automata, reachability ✓

Undecidable, decidable subclasses

-2. Existing zone-based algorithm ✓

Static analysis + zone enumeration with G-Simulation

-3. Our modification ✓

New static analysis with reduced constraint propagation

-4. Impact ✓

Covers T.A. with bounded subtraction; preemptive scheduling

PERSPECTIVES

- 1. G-simulation :

  easier to understand than LV

  needed for diagonal constraints

- 2. New static analysis :

  more impact in the presence of diagonal constraint & update

  can be easily incorporated in implementations of classical T.A.

- 3. Implementation :

  No existing tool allows general updates + diagonal constraints

  Work in progress to make our implementation public

PERSPECTIVES - II

- Diagonals still a problem

  Lazy abstractions for UTA à la :

  Herbreteau, S., Walukiewicz (CAV'13)
  Roussanaly, Markey, Sankur (CAV'19) ?

- Stop-watches using updates ?

- Links to verification of counter systems ?