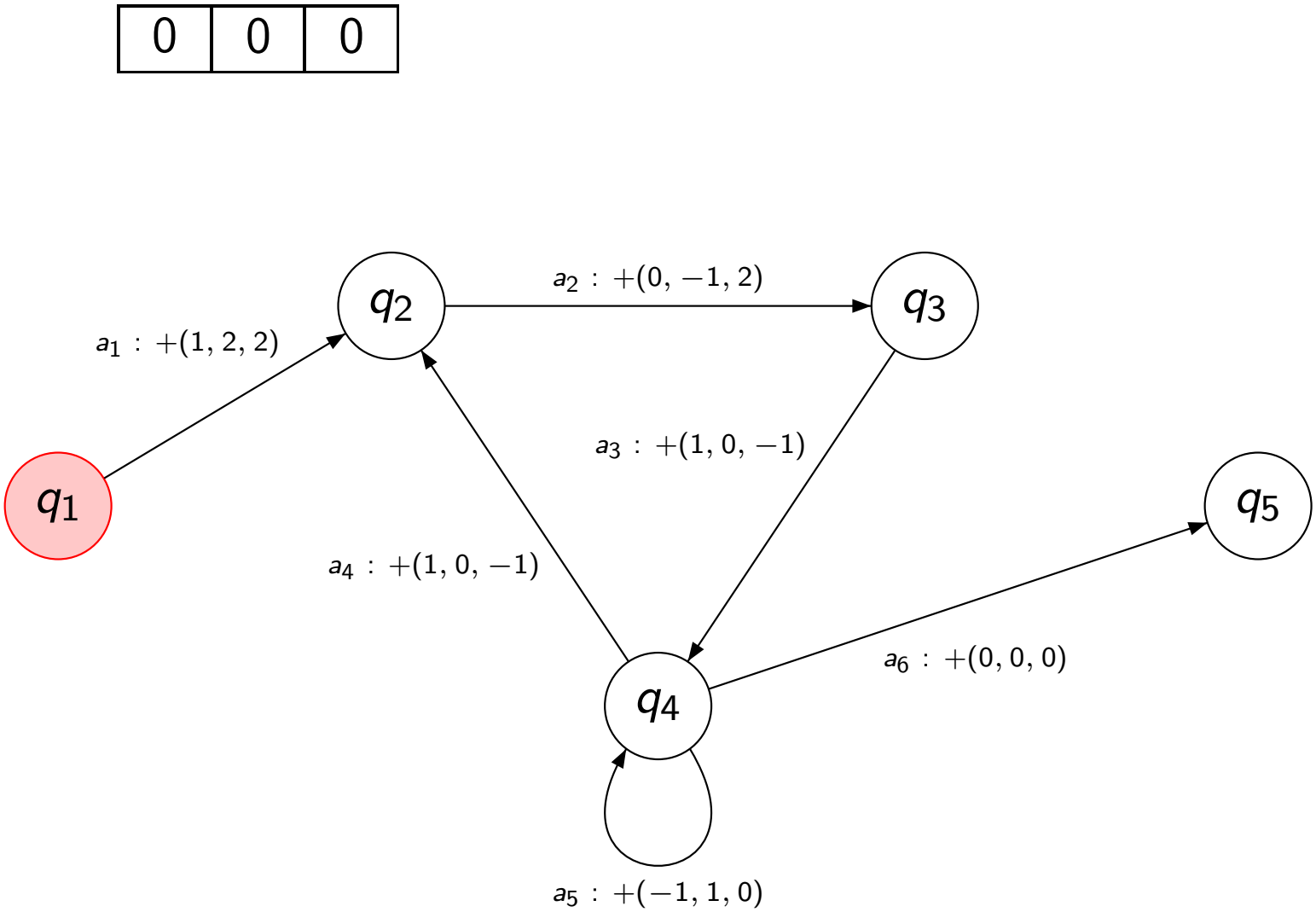
LTL Model-checking for Vector Addition Systems with one zero-test

Rémi Bonnet

LSV, CNRS, ENS Cachan

September 29, 2011

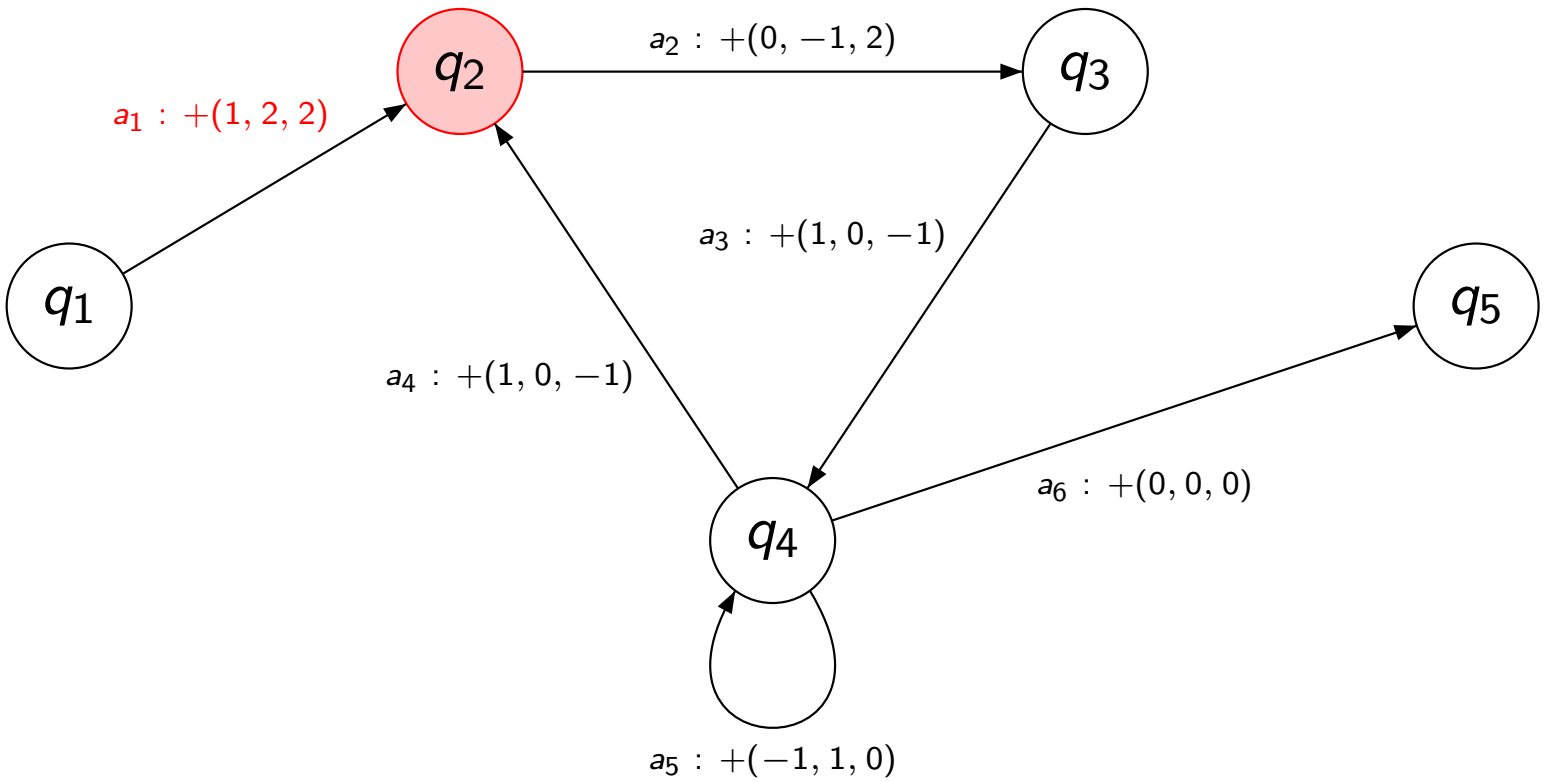
Counter Machines and Vector Addition Systems



Counter Machines and Vector Addition Systems

1	2	1

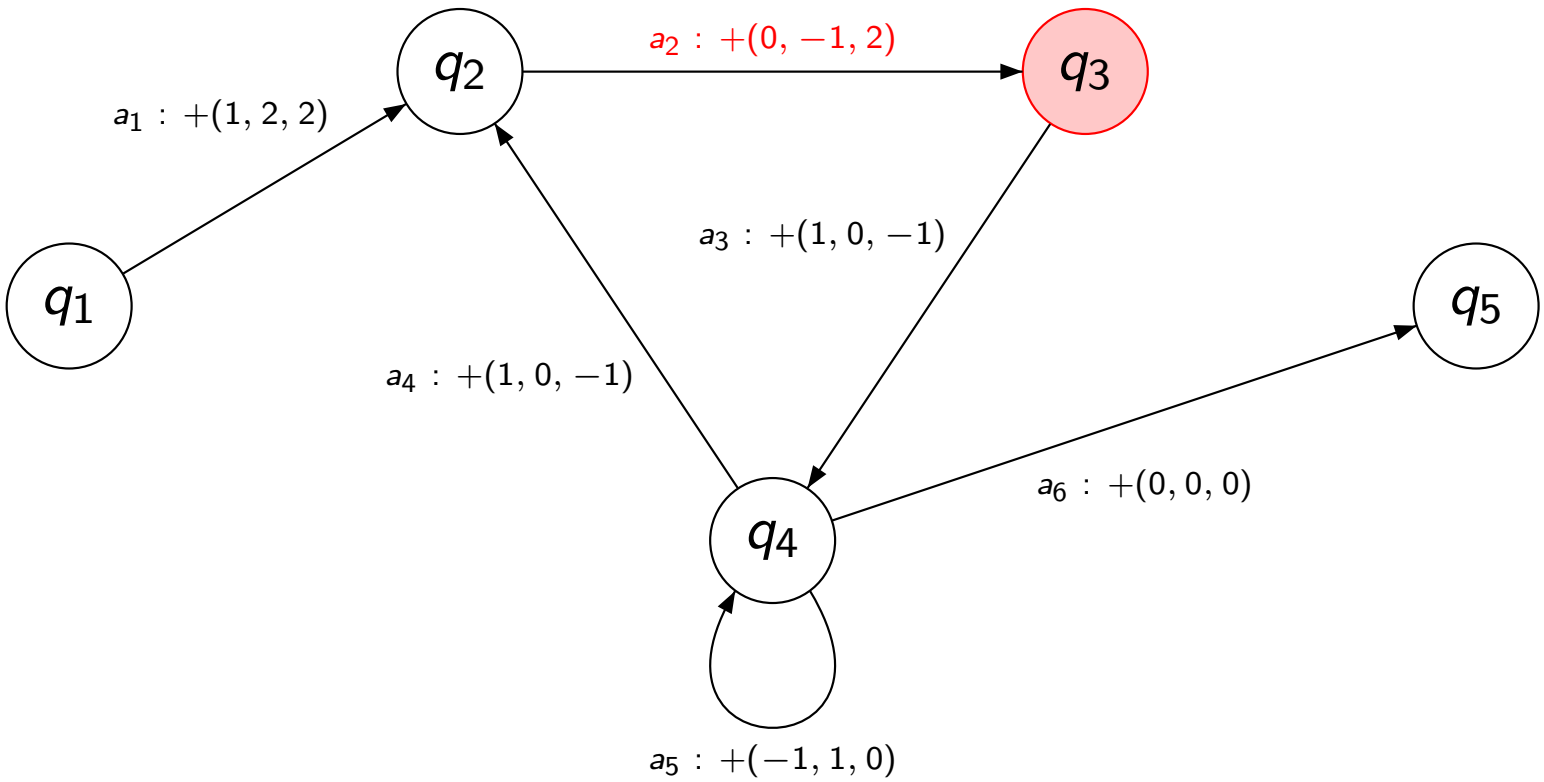
a_1



Counter Machines and Vector Addition Systems

1	1	3

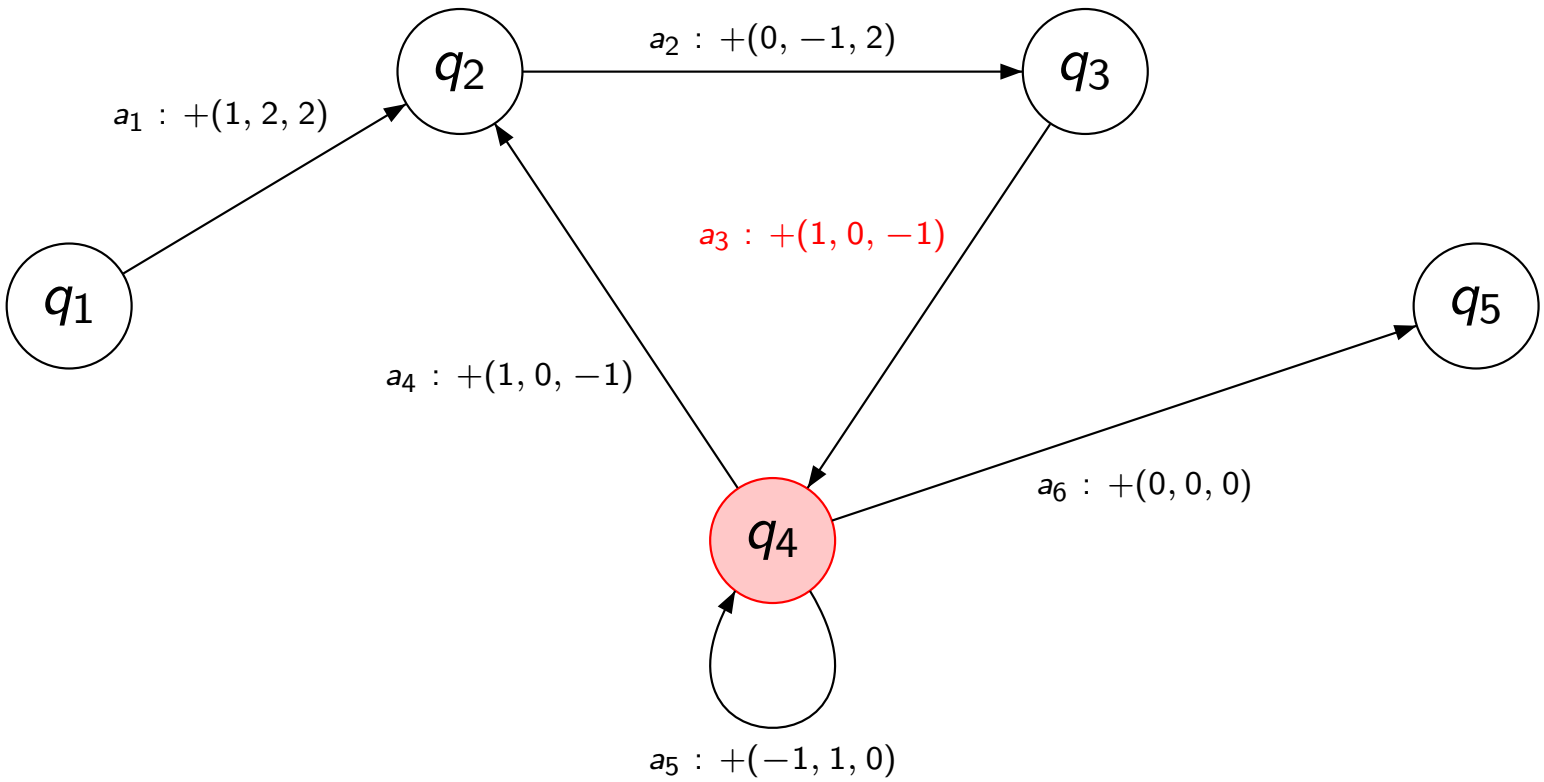
$a_1 a_2$



Counter Machines and Vector Addition Systems

2	1	2

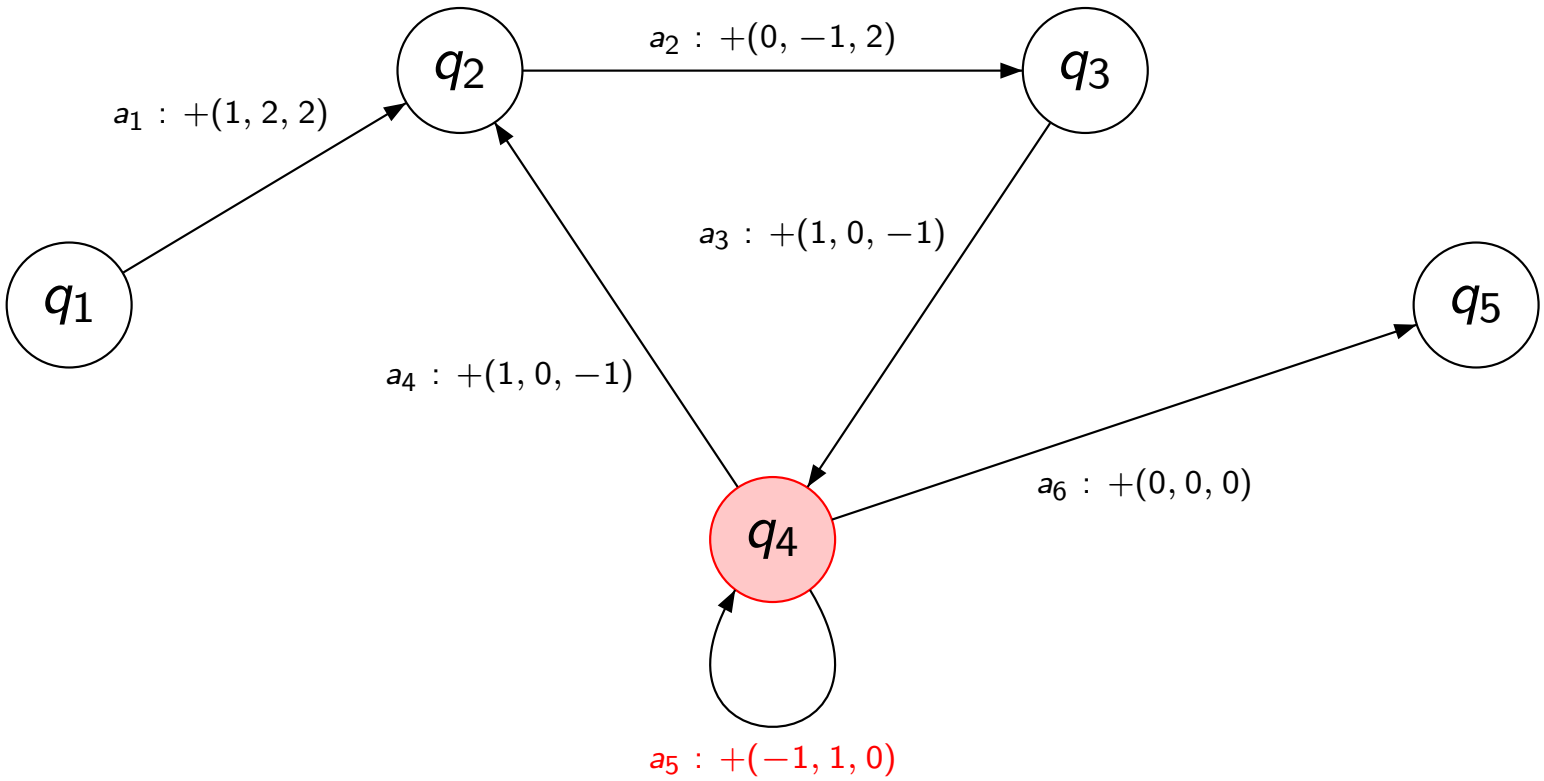
$a_1 a_2 a_3$



Counter Machines and Vector Addition Systems

1	2	2

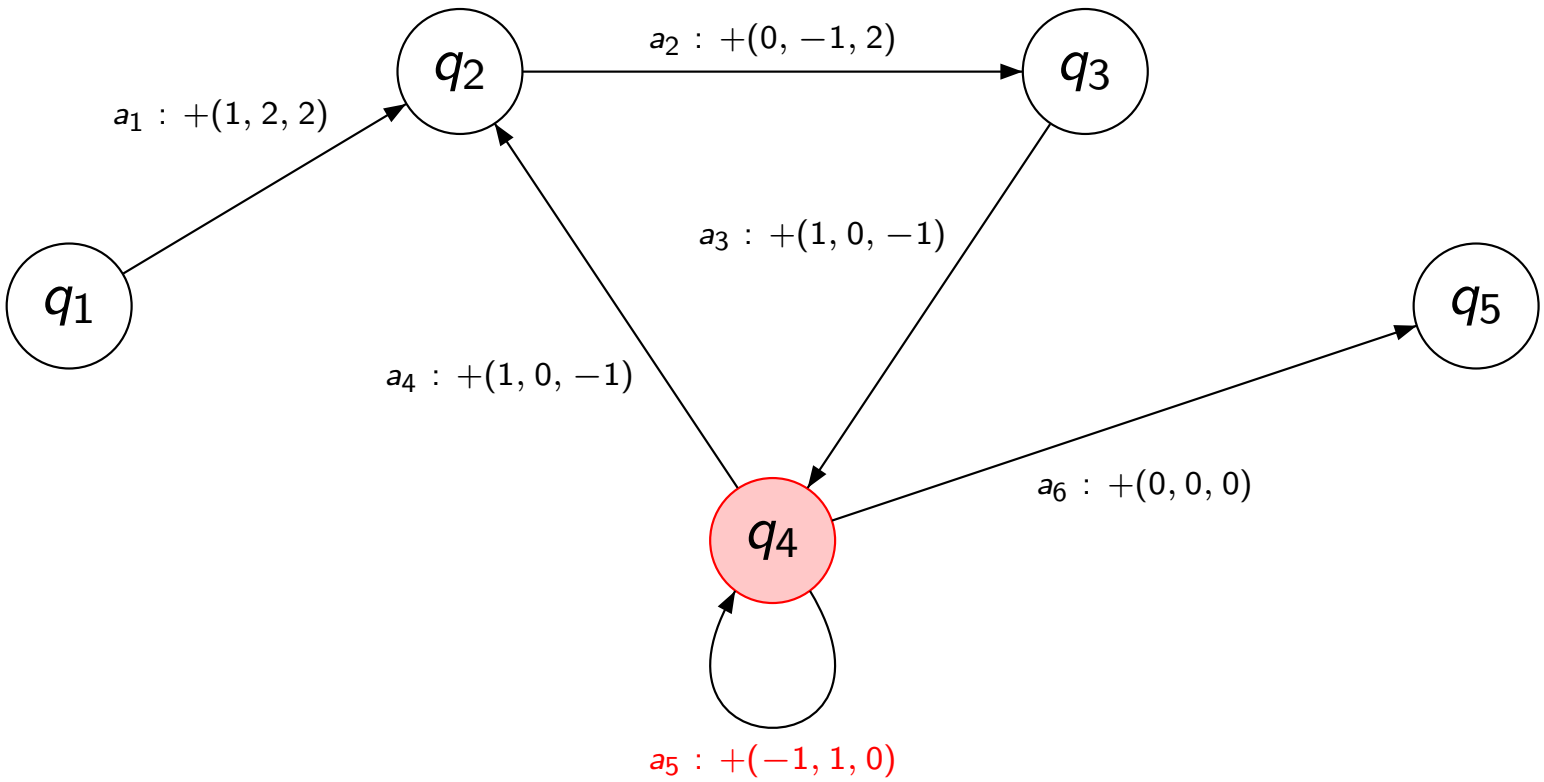
$a_1 a_2 a_3 a_5$



Counter Machines and Vector Addition Systems

0	3	1

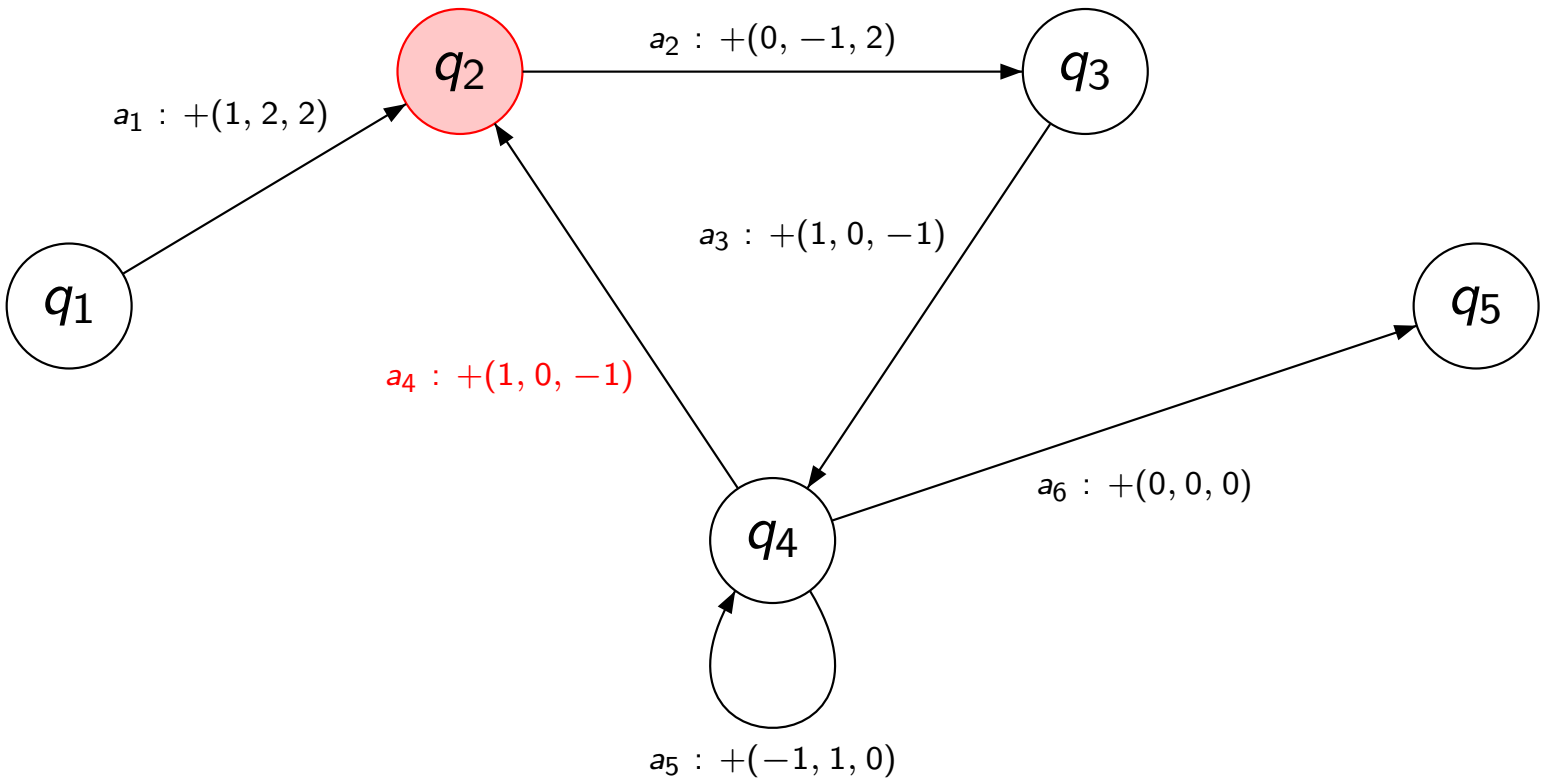
$a_1 a_2 a_3 a_5 a_5$



Counter Machines and Vector Addition Systems

1	3	0

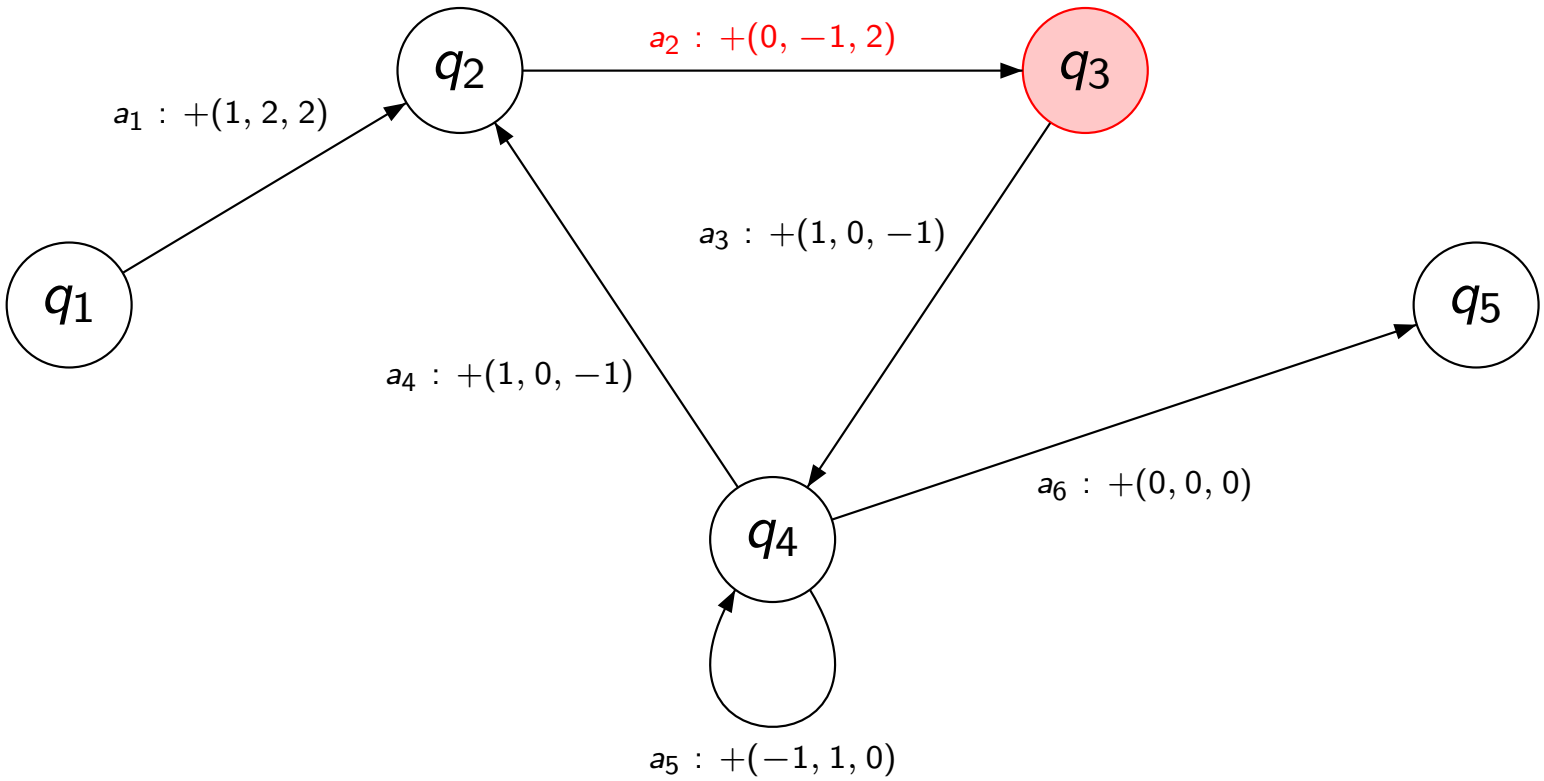
$a_1 a_2 a_3 a_5 a_5 a_4$



Counter Machines and Vector Addition Systems

1	2	2

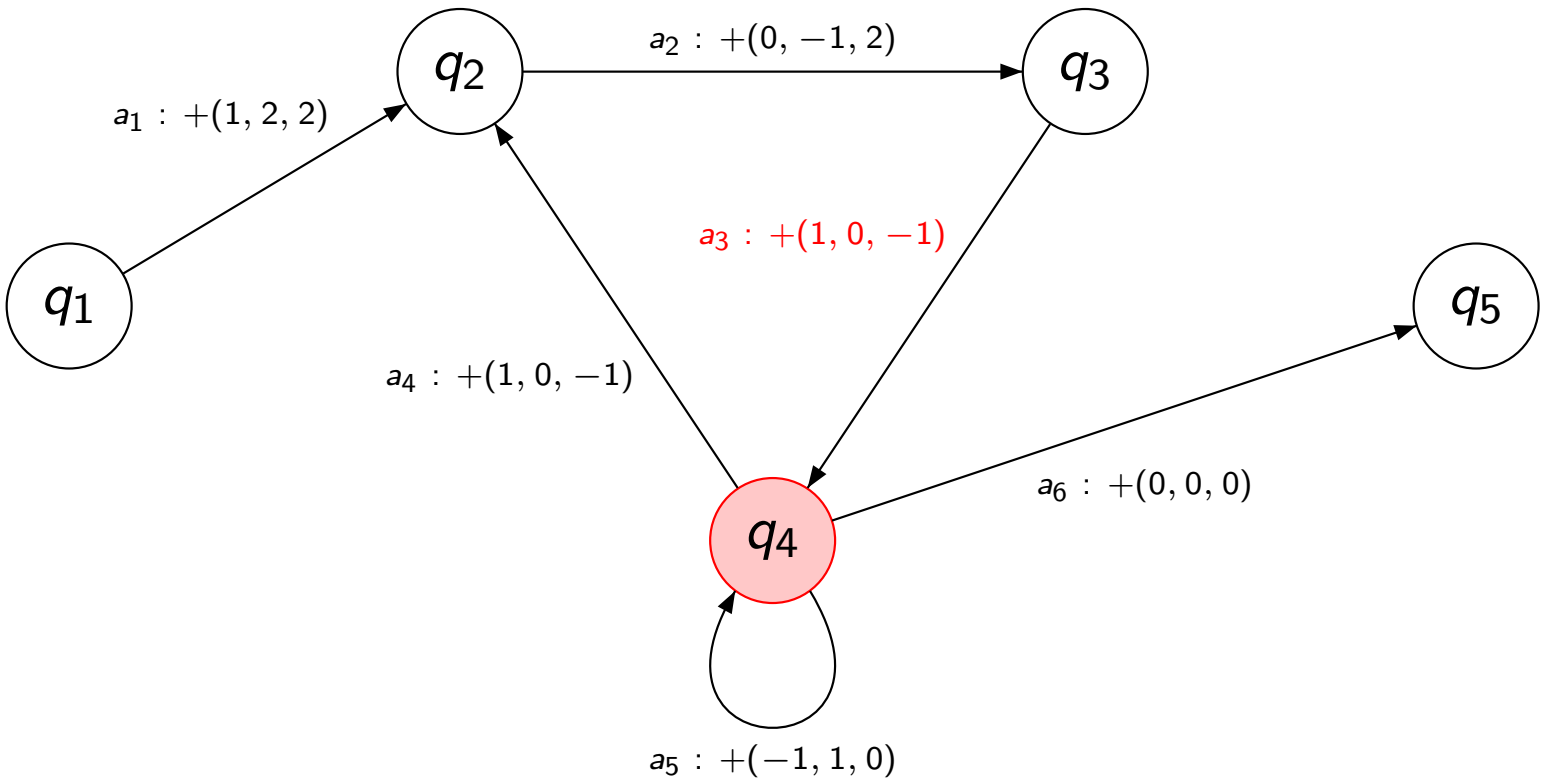
$a_1 a_2 a_3 a_5 a_5 a_4 a_2$



Counter Machines and Vector Addition Systems

2	1	1

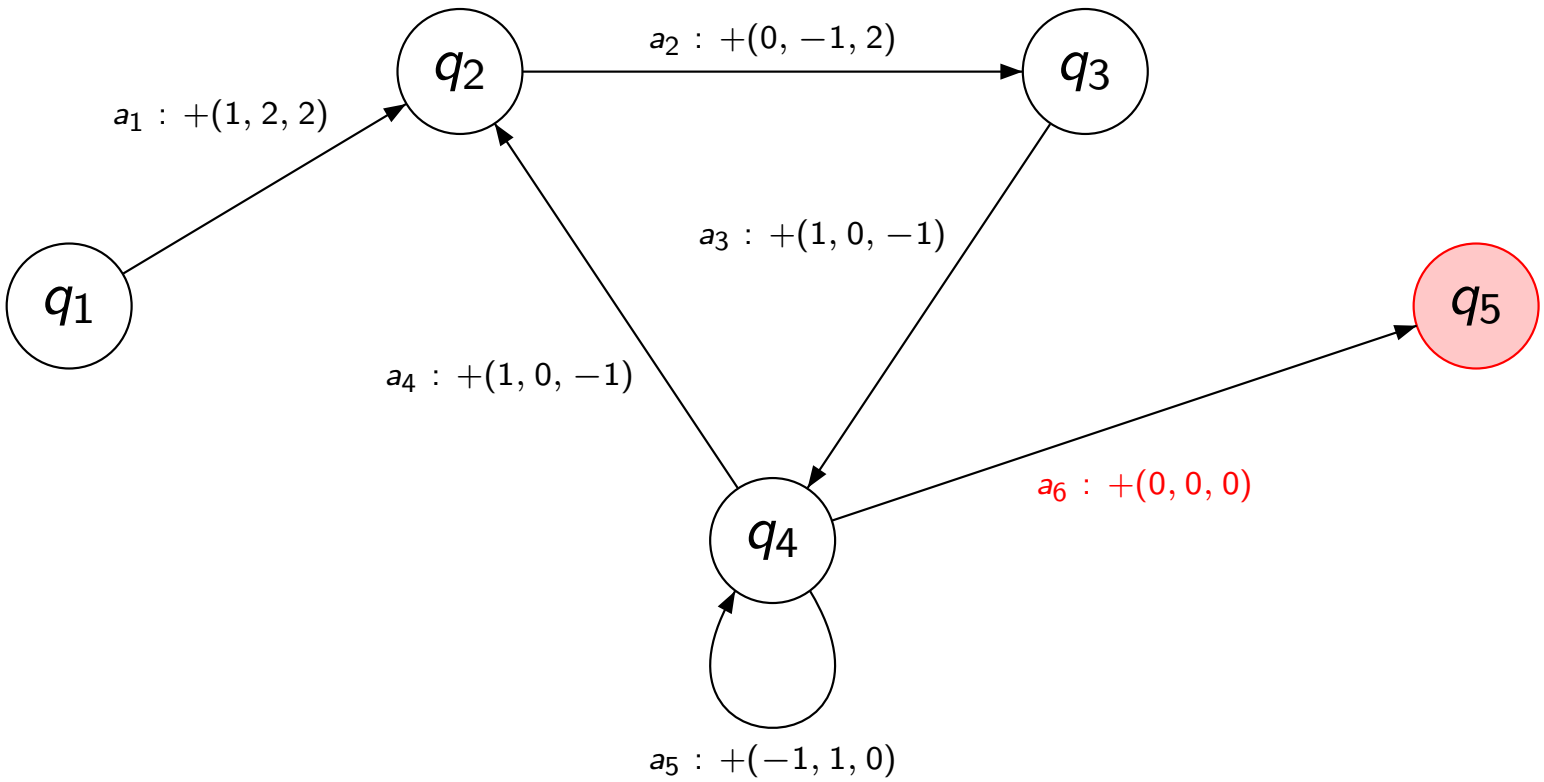
$a_1 a_2 a_3 a_5 a_5 a_4 a_2 a_3$



Counter Machines and Vector Addition Systems

2	1	1

$a_1 a_2 a_3 a_5 a_5 a_4 a_2 a_3 a_6$



Extensions of Vector Addition Systems

Vector Addition Systems : only additions of constant vectors of $\mathbb{Z}^d$.

Possible extensions :

- $x(i) := 0$ ("resets")
- $x(i) := x(i) + x(j)$ ("transfers")
- $x(i) = 0?$ ("zero-tests")

In this talk : we allow transitions labelled by $x(1) = 0?$.

Two categories of problems

- State-based problems: Is a state reachable? What states are reachable? How does the set of reachable states look like?
- Action-based problems: How does the set of traces look like? Is there a trace that satisifies a formulae of a logic?

State-based problems

A state is (q, x) where :

- $q \in Q$ is a control state.
- $x \in \mathbb{N}^d$ is the value of the counters.

$$\begin{array}{rcl} \mathit{Reach}(q_0, x_0) & = & \{(q, x) \in Q \times \mathbb{N}^d \mid (q_0, x_0) \xrightarrow{*} (q, x)\} \\ \mathit{Cover}(q_0, x_0) & = & \downarrow \mathit{Reach}(q_0, x_0) \end{array}$$

Problems regarding these sets:

- Coverability ($(q, x) \in \mathit{Cover}(q_0, x_0)$?) is decidable [Abdulla and Mayr '09]
- Boundedness ($\mathit{Reach}(q_0, x_0)$ *finite?*) is decidable [Finkel and Sangnier '10]
- Reachability ($(q, x) \in \mathit{Reach}(q_0, x_0)$?) is decidable [Reinhardt '08, B. '11]

An important tool for the analysis of VAS

Theorem [Baker '73, Hack '76]

The reachability set of a VAS cannot be effectively represented.

An important tool for the analysis of VAS

Theorem [Baker '73, Hack '76]

The reachability set of a VAS cannot be effectively represented.

However, the cover can be finitely represented:

Proposition

Given a downward closed set $D \subseteq \mathbb{N}^d$, there exists a finite set $B \subseteq \mathbb{N}_\omega^d$ such that $\downarrow B \cap \mathbb{N}^d = D$.

An important tool for the analysis of VAS

Theorem [Baker '73, Hack '76]

The reachability set of a VAS cannot be effectively represented.

However, the cover can be finitely represented:

Proposition

Given a downward closed set $D \subseteq \mathbb{N}^d$, there exists a finite set $B \subseteq \mathbb{N}_\omega^d$ such that $\downarrow B \cap \mathbb{N}^d = D$.

and this representation can be effectively computed:

Theorem [Karp and Miller '69]

Given a VAS and an initial state (q_0, x_0), one can compute the finite set of the maximal elements of $Cover(q_0, x_0)$.

An important tool for the analysis of VAS_0

Theorem [B., Finkel, Leroux and Zeitoun '10]

Given a VAS_0 and an initial state (q_0, x_0), one can compute the finite set of the maximal elements of $Cover(q_0, x_0)$.

Action-based problems

$$
\begin{aligned}
Traces(q_0, x_0) &= \{u \in A^* \mid \exists (q, x) \in Q \times \mathbb{N}^d,\ (q_0, x_0) \xrightarrow{u} (q, x)\} \\
Traces^{\omega}(q_0, x_0) &= \{u \in A^{\omega} \mid (q_0, x) \xrightarrow{u} ...\} \\
L_r(q_0, x_0, q_f, x_f) &= \{u \in A^* \mid (q_0, x_0) \xrightarrow{u} (q, x)\}
\end{aligned}
$$

- Regularity (is $Traces(q_0, x_0)$ a regular language?)
- Model Checking (does $Traces^{\omega}(q_0, x_0)$ satisfies a formula?)
- ... and many others

Most results known for VAS. Very few results in this area for VAS_0.

LTL Model-Checking

Definition : LTL Formulas

$$\varphi ::= a \text{ (with } a \in A) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathcal{X}\varphi \mid \Box\varphi \mid \Diamond\varphi$$

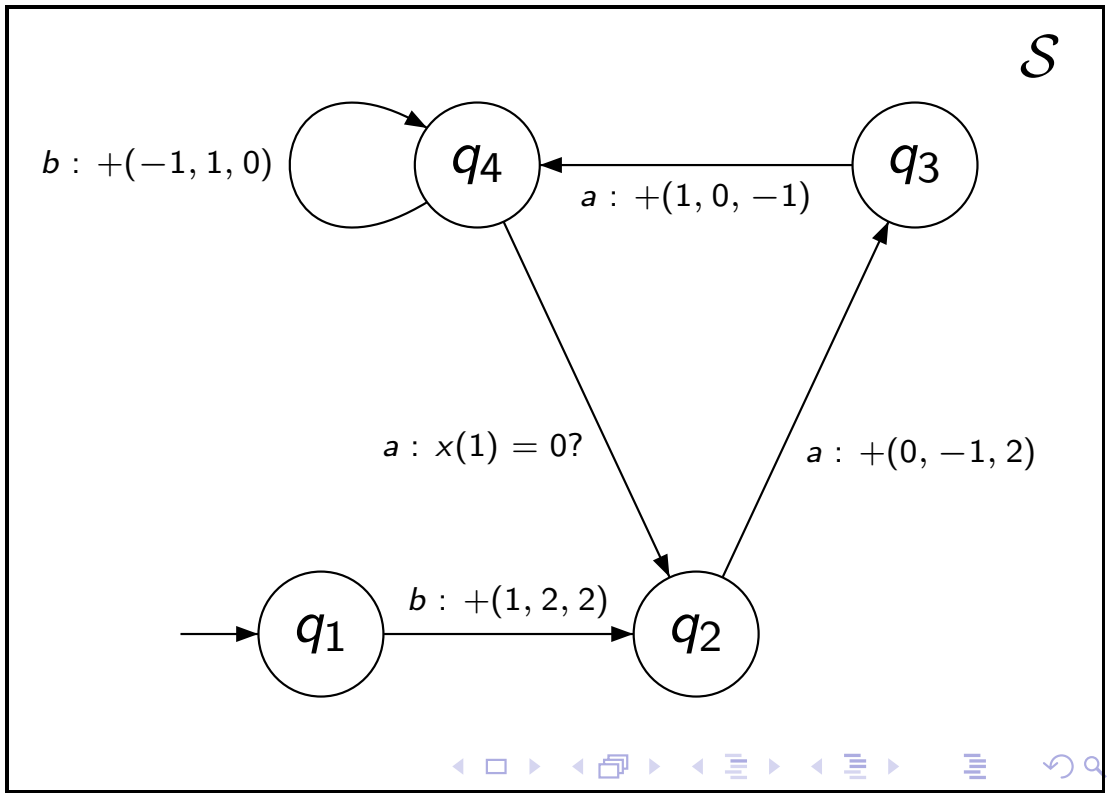
A transition system satisfies φ if there exists an infinite trace w such that $w \models \varphi$.

Decidability of Temporal Logics

	VAS	VAS_0
Coverability	decidable (EXPSPACE) [Karp and Miller '69, Rackoff '78]	decidable [Abdulla and Mayr '09]
Reachability	decidable [Mayr '81, Kosaraju '82, Leroux '11]	decidable [Reinhardt '08, B. '11]
Cover	effective [Karp and Miller '69]	effective [B., Finkel, Leroux, Zeitoun '10]
LTL on actions	decidable (EXPSPACE) [Esperza '94, Habermehl '97]	this work
LTL on states	undecidable [Esperza '94]	undecidable
CTL	undecidable [Esperza '94]	undecidable

LTL and Buchi Automatas

$\Box\Diamond ab$

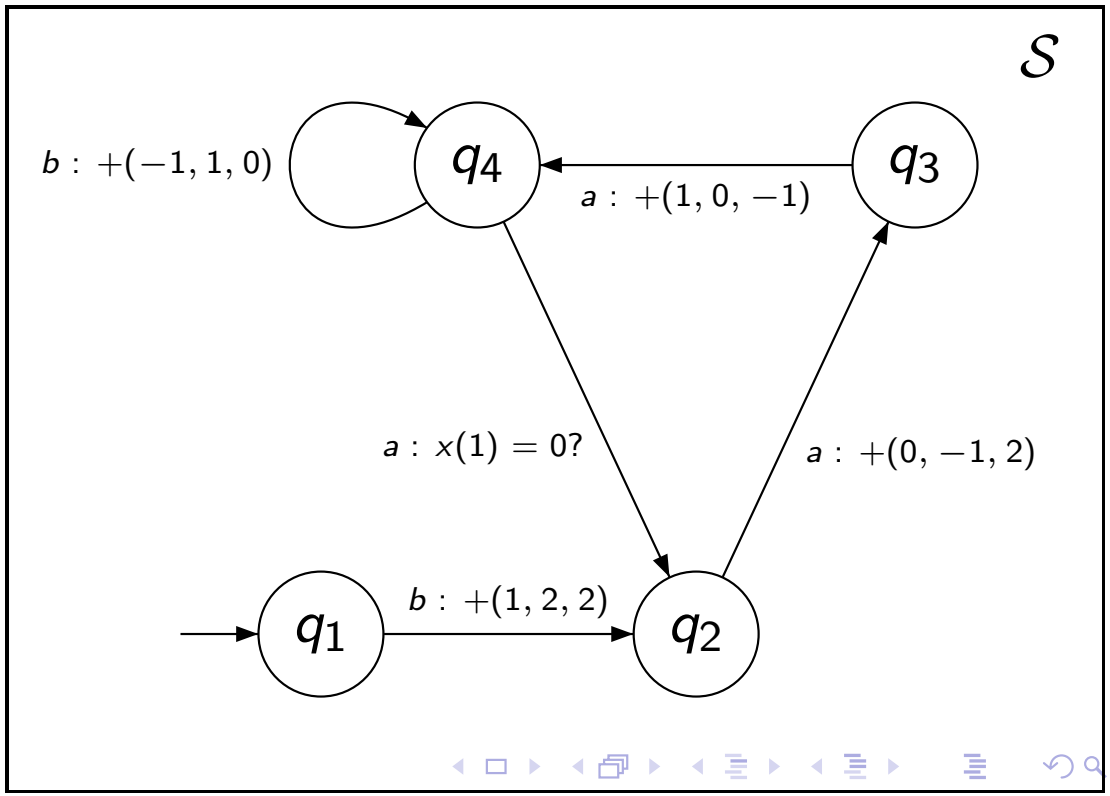


LTL and Buchi Automatas

- For any formula φ, one can build a Buchi automata $\mathcal{A}_\varphi$ with:

$$w \models \varphi \iff w \text{ is recognized by } \mathcal{A}_\varphi$$

$\Box\Diamond ab$

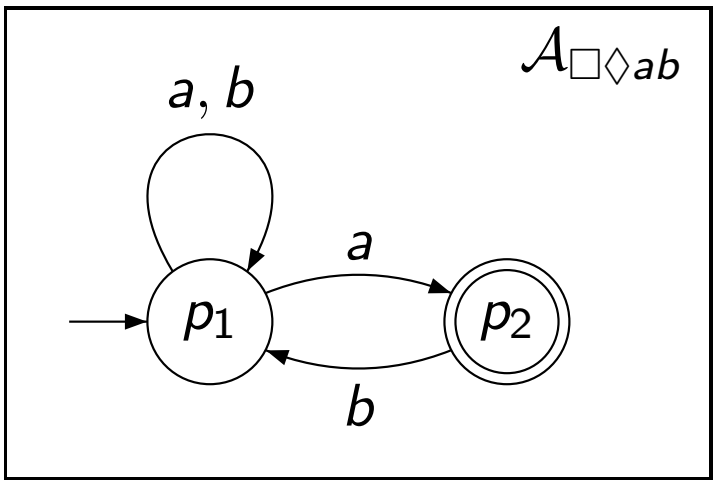


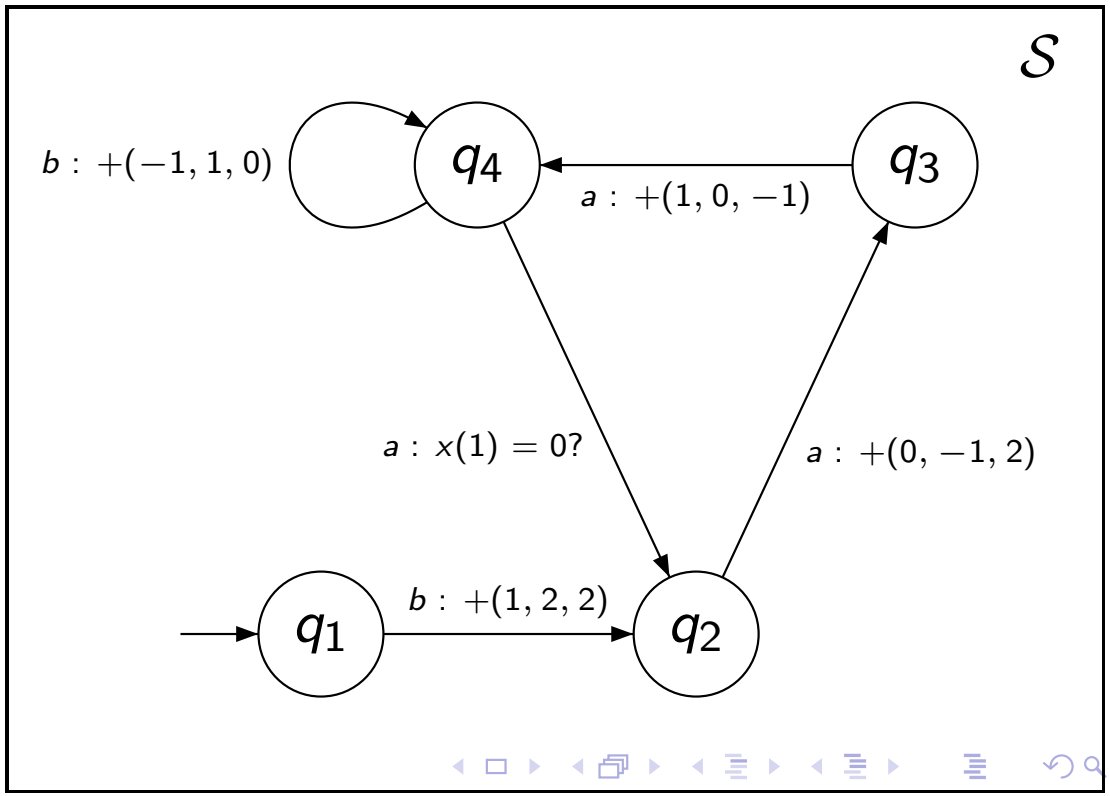
LTL and Buchi Automatas

- For any formula φ, one can build a Buchi automata $\mathcal{A}_\varphi$ with:

$$w \models \varphi \iff w \text{ is recognized by } \mathcal{A}_\varphi$$

$\Box\Diamond ab$



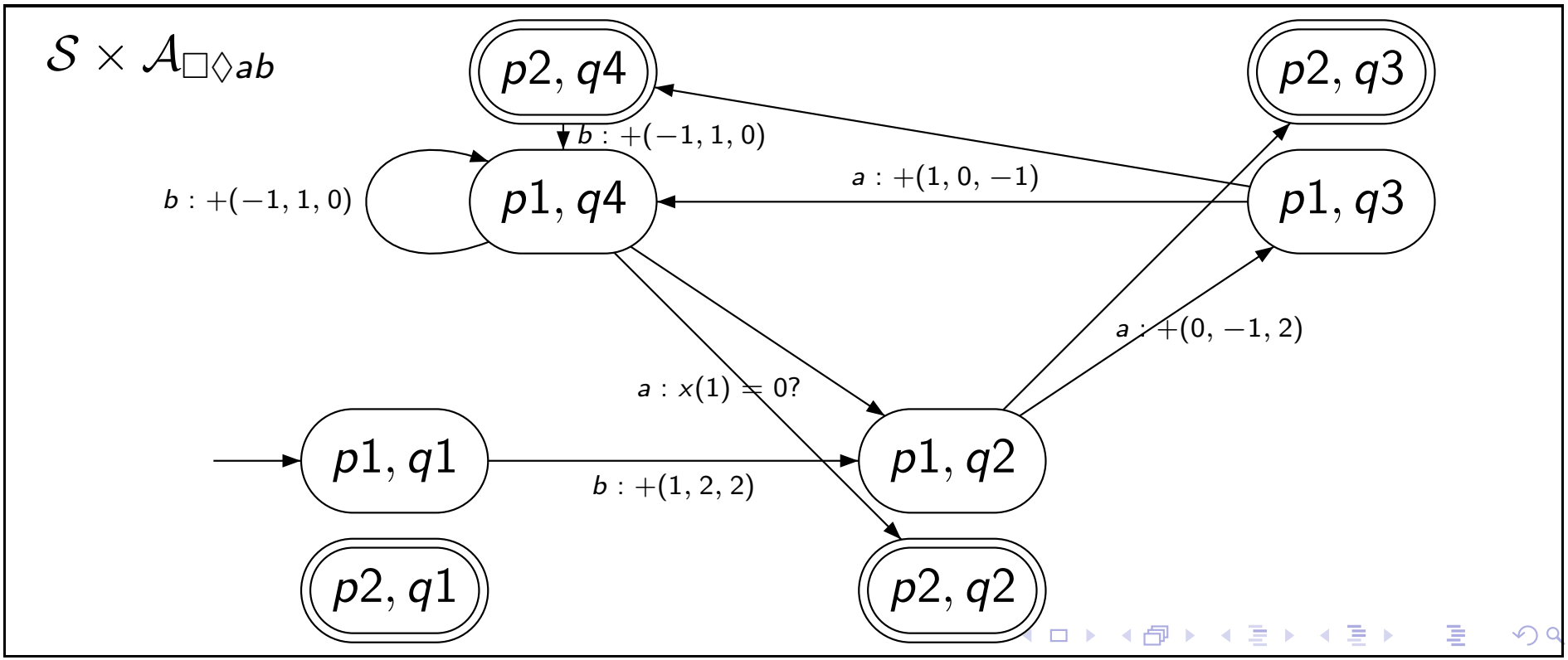


LTL and Buchi Automatas

- For any formula φ, one can build a Buchi automata $\mathcal{A}_\varphi$ with:

$$w \models \varphi \iff w \text{ is recognized by } \mathcal{A}_\varphi$$

- LTL satisfiability of φ by a VAS_0 $\mathcal{S}$ is equivalent to the repeated reachability of the final control states in $\mathcal{S} \times \mathcal{A}_\varphi$.



Our aim

Our aim

Is Repeated Control State Reachability decidable?

On Vector Addition Systems

Definition : increasing loop

(x, y) is an increasing loop on q_f if we have $x \leq y$ and $(q_f, x) \xrightarrow{*} (q_f, y)$.

Proposition

A control state q can be reached infinitely often if and only there is an increasing loop (x, y) on q_f such that $(q_f, x) \in \mathit{Cover}(q_0, x_0)$.

Increasing loops can be detected (Yen's path logic ...)

And on VAS_0?

And on VAS_0?

Problem

VAS_0 are not monotonic! An increasing loop doesn't imply repeated reachability

And on VAS_0?

Problem

VAS_0 are not monotonic! An increasing loop doesn't imply repeated reachability

But, they have some form of limited monotonicity...

And on VAS_0?

Problem

VAS_0 are not monotonic! An increasing loop doesn't imply repeated reachability

But, they have some form of limited monotonicity...

Proposition

If a sequence of transitive is fireable from $(q, (0, x))$, it is fireable from $(q, (0, y))$ for $x \leq y$.

And on VAS_0?

Problem

VAS_0 are not monotonic! An increasing loop doesn't imply repeated reachability

But, they have some form of limited monotonicity...

Proposition

If a sequence of transitive is fireable from $(q, (0, x))$, it is fireable from $(q, (0, y))$ for $x \leq y$.

We will look at increasing loops with the first counter value fixed to 0!

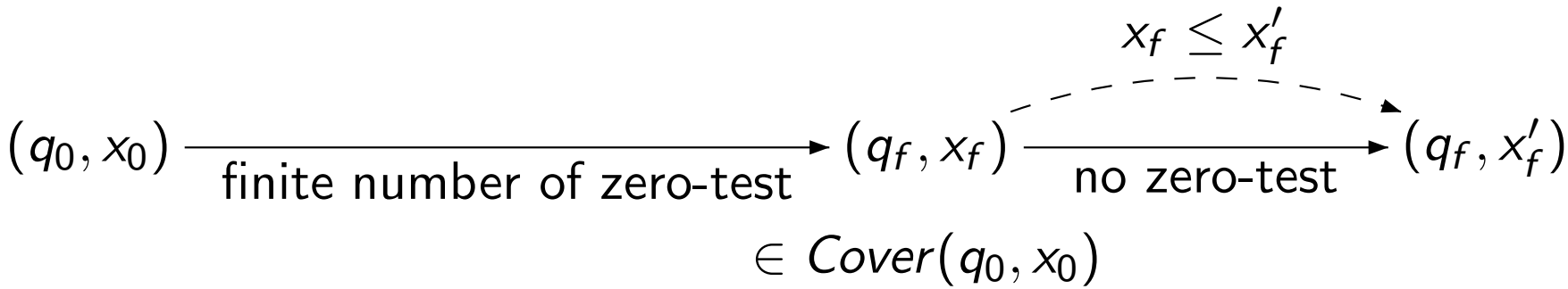
An easy case

Proposition

One can decide whether a control state is reached infinitely often on a run where the zero-test is fired only a finite number of cases.

Proof:

Reduction to repeated control state reachability.



Reduction to the detection of an increasing loop

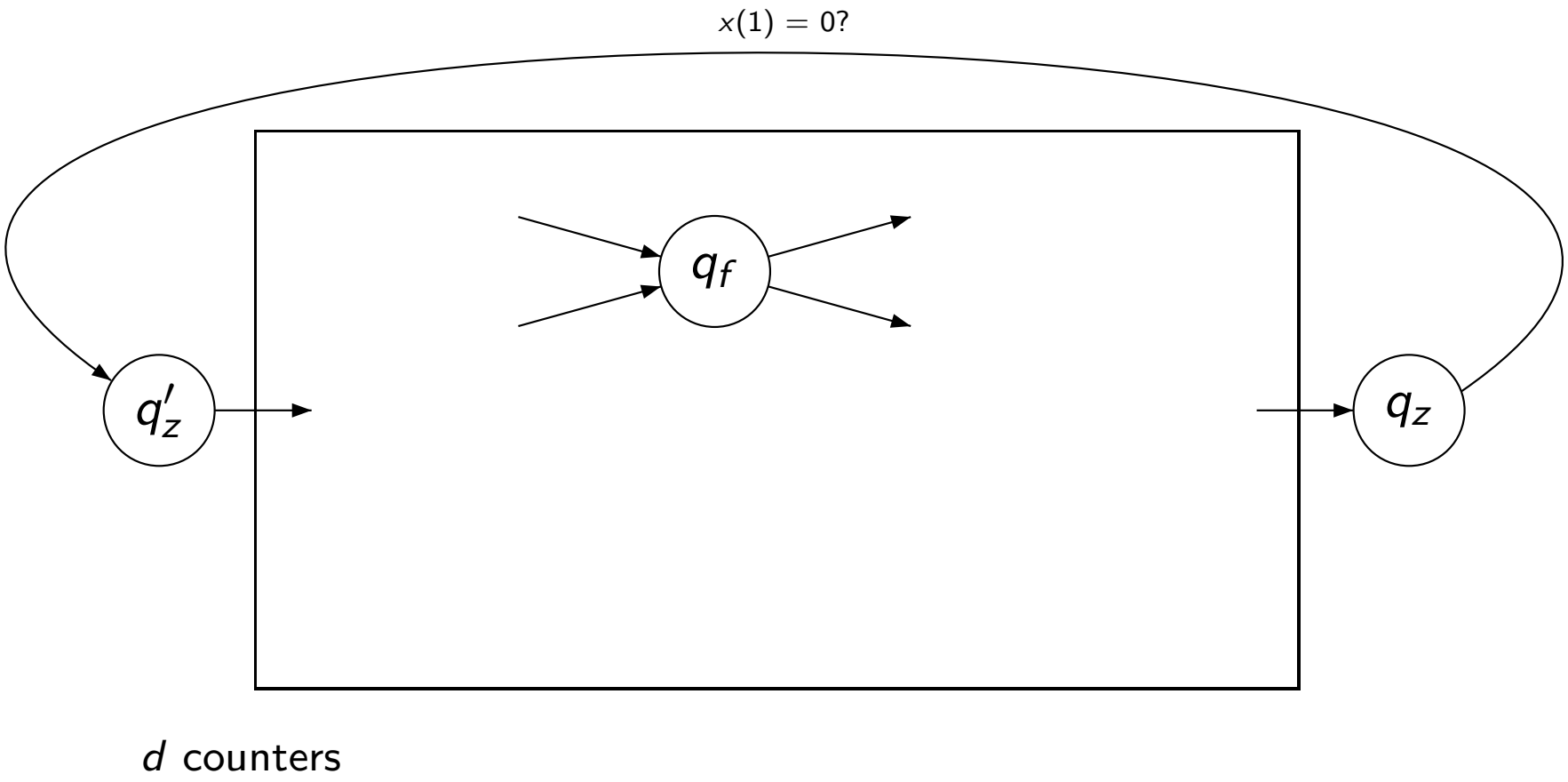
Proposition

We can require that the value of the first counter is zero on the control state that must be reached infinitely often.

Reduction to the detection of an increasing loop

Proposition

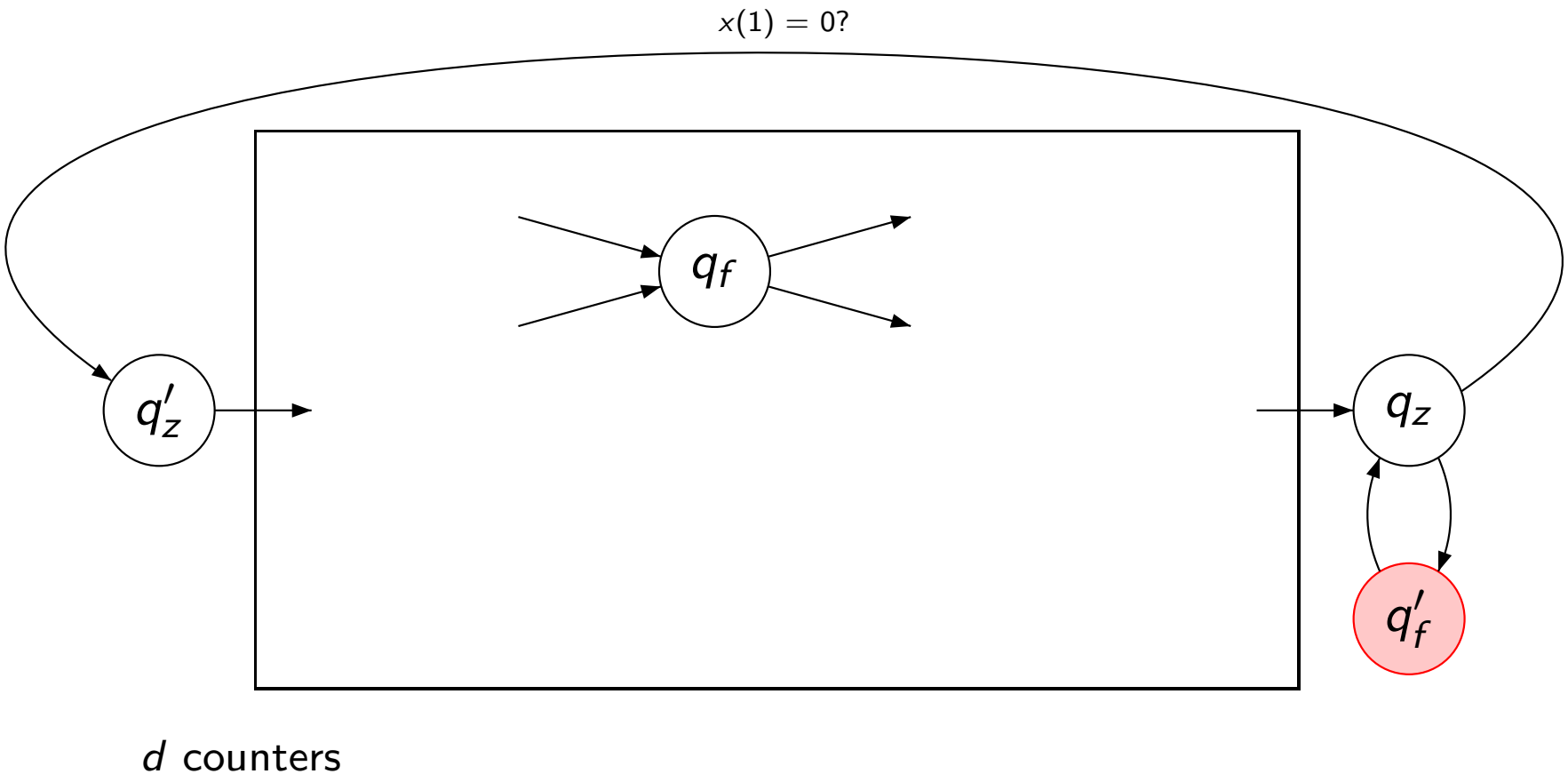
We can require that the value of the first counter is zero on the control state that must be reached infinitely often.



Reduction to the detection of an increasing loop

Proposition

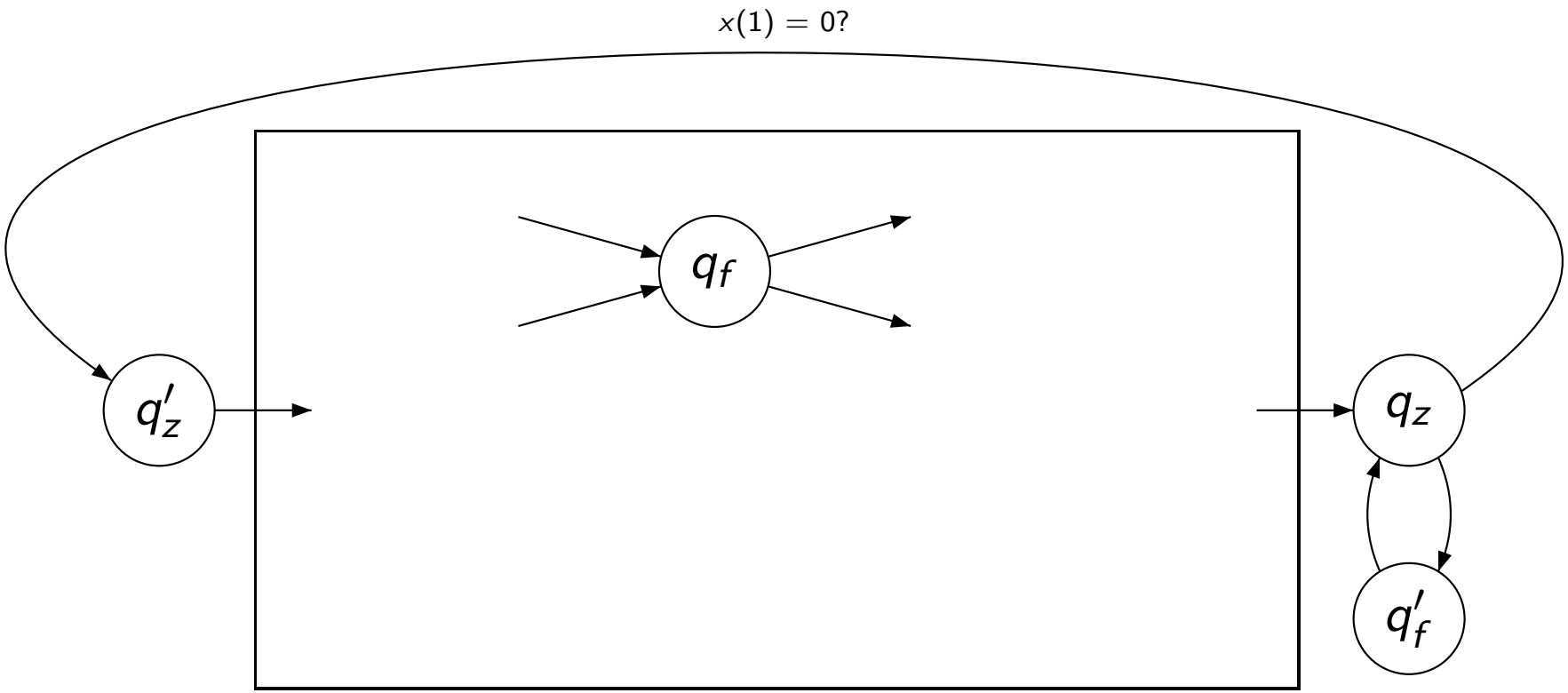
We can require that the value of the first counter is zero on the control state that must be reached infinitely often.



Reduction to the detection of an increasing loop

Proposition

We can require that the value of the first counter is zero on the control state that must be reached infinitely often.

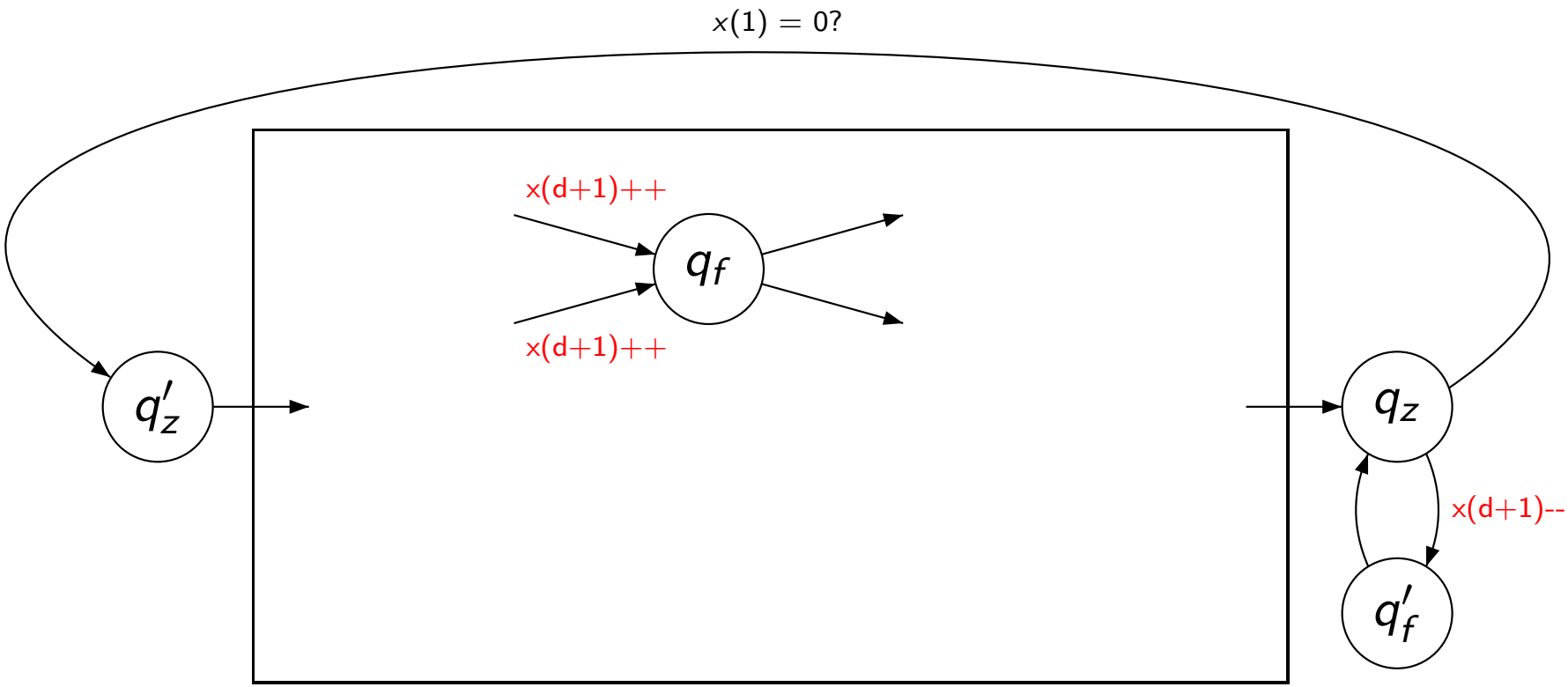


d counters + 1 counter ensuring q'_f can be visited as many times as q_f

Reduction to the detection of an increasing loop

Proposition

We can require that the value of the first counter is zero on the control state that must be reached infinitely often.



d counters + 1 counter ensuring q'_f can be visited as many times as q_f

Detecting an increasing loop

Question

Let $I \in \mathbb{N}_\omega^{d-1}$ and q control state. Is there $x \leq I$ and $y \geq x$ such that $(q,(0,x)) \xrightarrow{*} (q,(0,y))$?

Detecting an increasing loop

Question

Let $I \in \mathbb{N}_\omega^{d-1}$ and q control state. Is there $x \leq I$ and $y \geq x$ such that $(q,(0,x)) \xrightarrow{*} (q,(0,y))$?

- If $I \in \mathbb{N}^{d-1}$, this is a coverability problem.

Detecting an increasing loop

Question

Let $I \in \mathbb{N}_\omega^{d-1}$ and q control state. Is there $x \leq I$ and $y \geq x$ such that $(q,(0,x)) \xrightarrow{*} (q,(0,y))$?

- If $I \in \mathbb{N}^{d-1}$, this is a coverability problem.
- If ω appears in I, the associated counters can start as high as they want, as long as the final value is higher.

Detecting an increasing loop

Question

Let $I \in \mathbb{N}_\omega^{d-1}$ and q control state. Is there $x \leq I$ and $y \geq x$ such that $(q,(0,x)) \xrightarrow{*} (q,(0,y))$?

- If $I \in \mathbb{N}^{d-1}$, this is a coverability problem.
- If ω appears in I, the associated counters can start as high as they want, as long as the final value is higher.

this looks like a counter that can accept negative integers!

Detecting an increasing loop

Proposition

Let $I \in \mathbb{N}_\omega^{d-1}$ and q control state. There exists $x \leq I$ and $y \geq x$ such that $(q,(0,x)) \xrightarrow{*} (q,(0,y))$ iff we have a run where:

the first counter	starts at 0,	lives in $\mathbb{N}$	ends at 0.
a counter with $I(i) = \omega$	starts at 0,	lives in $\mathbb{Z}$,	ends at 0 or higher.
a counter with $I(i) \in \mathbb{N}$	starts at $I(i)$,	lives in $\mathbb{N}$,	ends at $I(i)$ or higher.

Can we simulate such counters by using only operations allowed in VAS_0?

Encoding a counter on $\mathbb{Z}$

- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.

$c : 0$

$c^+ : 0$
$c^- : 0$

Encoding a counter on $\mathbb{Z}$

- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.

$$c : 0 \xrightarrow{c{+}{+}} c : 1$$

$c^+ : 0$
$c^- : 0$

Encoding a counter on $\mathbb{Z}$

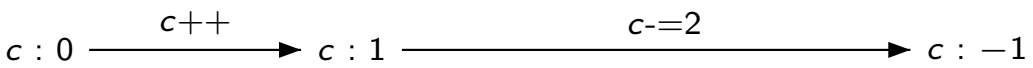
- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.

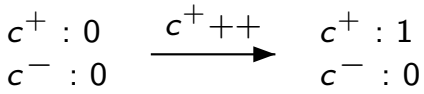
$$c : 0 \xrightarrow{c{+}{+}} c : 1$$

$$\begin{matrix} c^+ : 0 \\ c^- : 0 \end{matrix} \xrightarrow{c^+{+}{+}} \begin{matrix} c^+ : 1 \\ c^- : 0 \end{matrix}$$

Encoding a counter on $\mathbb{Z}$

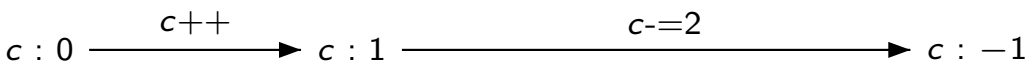
- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.

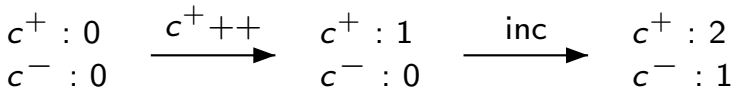




Encoding a counter on $\mathbb{Z}$

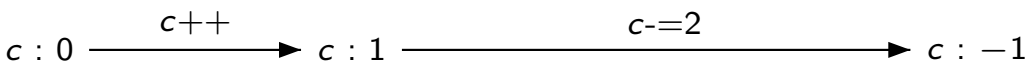
- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.

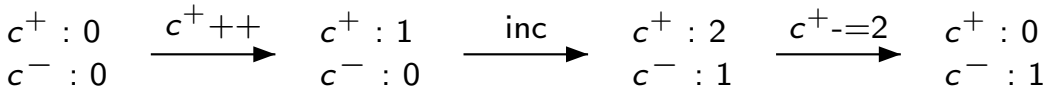




Encoding a counter on $\mathbb{Z}$

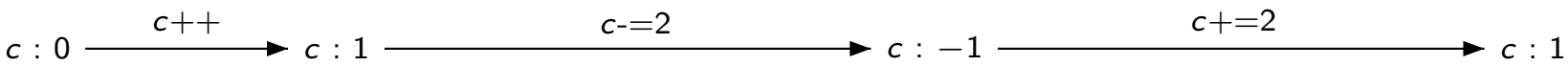
- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.

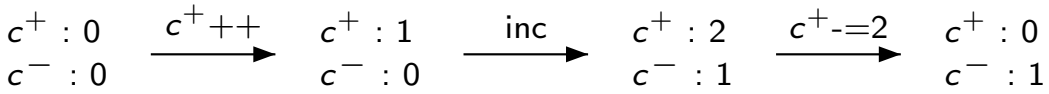




Encoding a counter on $\mathbb{Z}$

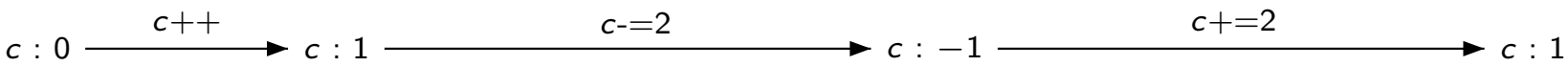
- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.

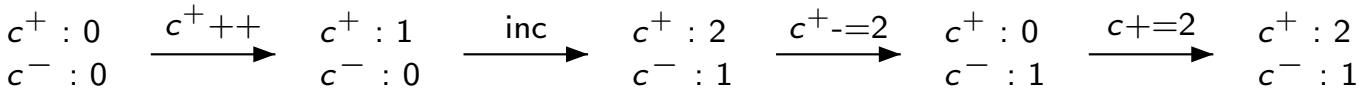




Encoding a counter on $\mathbb{Z}$

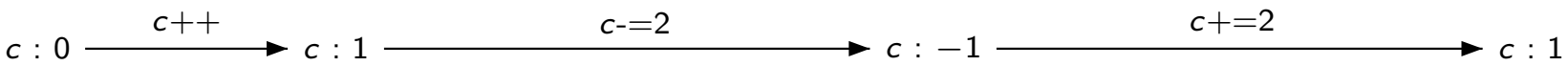
- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.

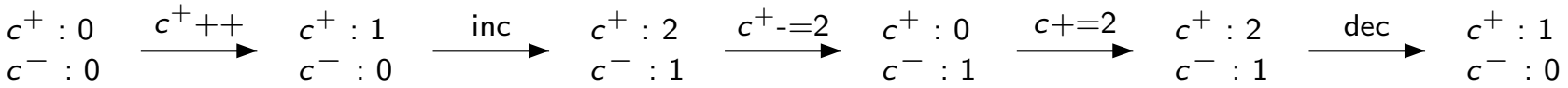




Encoding a counter on $\mathbb{Z}$

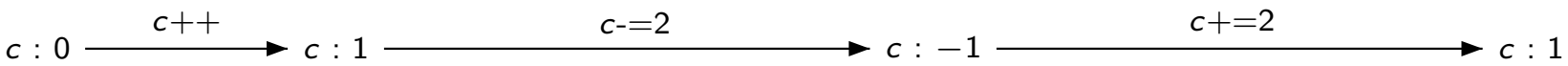
- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.

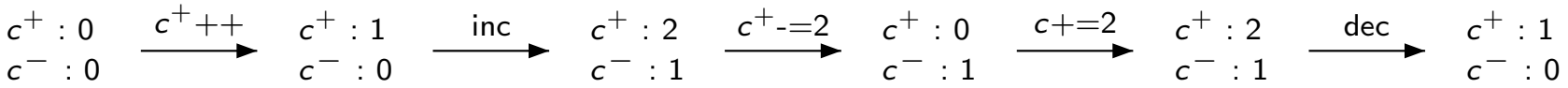




Encoding a counter on $\mathbb{Z}$

- A counter c on $\mathbb{Z}$ = two counters c^+, c^- on $\mathbb{N}$ that can at any time be simulataneously increased or decreased. Transitions apply to c^+.
- We have $c = c^+ - c^-$.





- Coverability for $\mathbb{Z}$-counters reduces to Reachability.

Detecting an increasing loop

Proposition

Let $I \in \mathbb{N}_\omega^{d-1}$ and q control state. Deciding whether there exists $x \leq I$ and $y \geq x$ such that $(q,(0,x)) \xrightarrow{*} (q,(0,y))$ reduces to reachability in VAS_0.

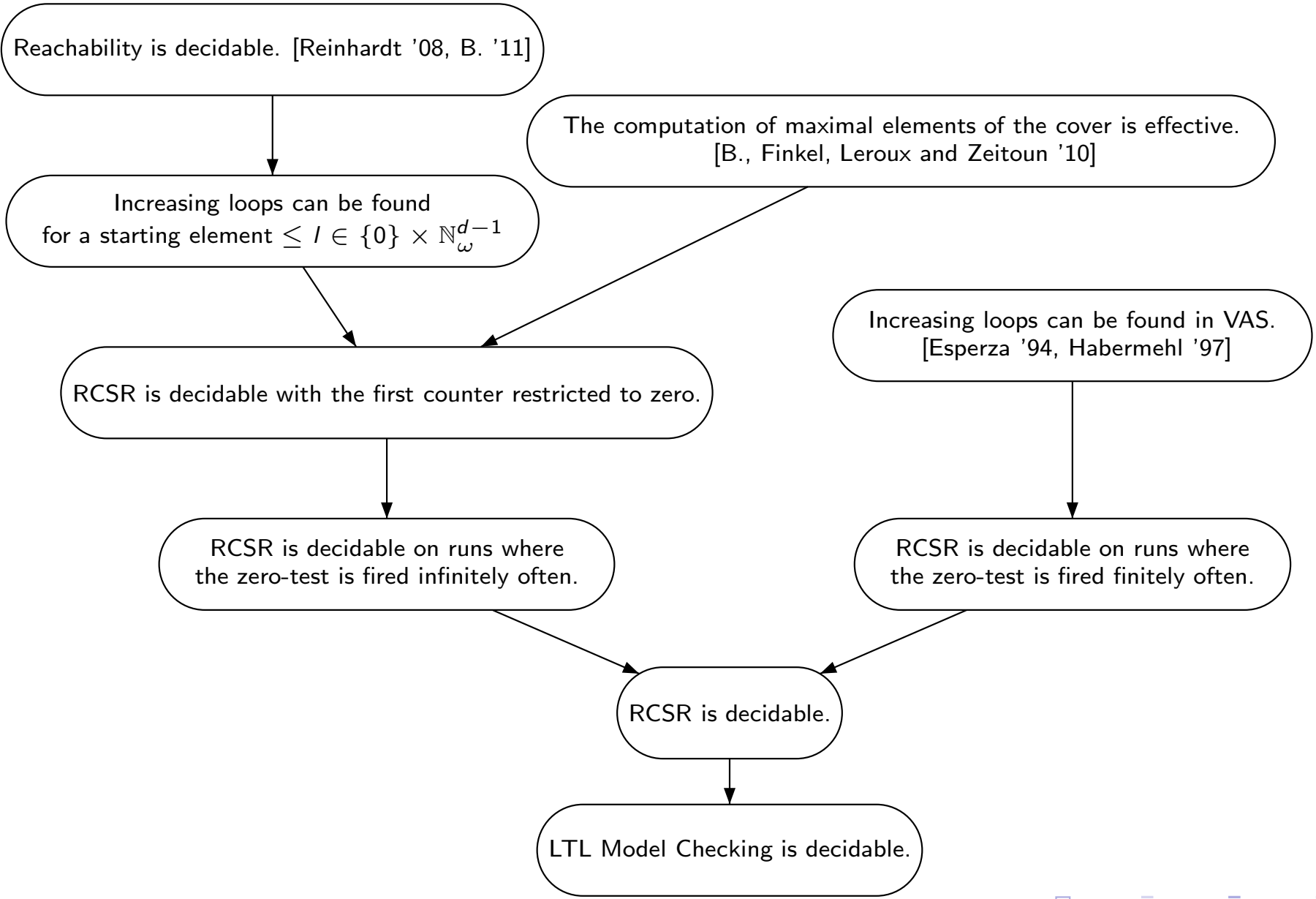
Thanks to the effective computation of the maximal elements of the cover and the decidability of the reachability:

Theorem

Let q control state and $s_{in} \in Q \times \mathbb{N}^d$. It is decidable whether there exists $x, y \in \mathbb{N}^{d-1}$ such that:

- $(q,(0,x)) \in \mathit{Cover}(s_{in})$
- $x \leq y$
- $(q,(0,x)) \xrightarrow{*} (q,(0,y))$

Summing up


Reachability is decidable. [Reinhardt '08, B. '11]
The computation of maximal elements of the cover is effective. [B., Finkel, Leroux and Zeitoun '10]
Increasing loops can be found for a starting element $\leq I \in \{0\} \times \mathbb{N}_\omega^{d-1}$
Increasing loops can be found in VAS. [Esperza '94, Habermehl '97]
RCSR is decidable with the first counter restricted to zero.
RCSR is decidable on runs where the zero-test is fired infinitely often.
RCSR is decidable on runs where the zero-test is fired finitely often.
RCSR is decidable.
LTL Model Checking is decidable.

Conclusion

- We have shown the decidability of LTL Model Checking for Vector Addition Systems with one zero-test.
- This was (to my knowledge) the only canonic problem that was known decidable for VAS and not for VAS_0.
- Strangely enough, if all decidability results match, complexity for VAS_0 is unknown.
- Little results regarding properties of the set of traces are known.