News on Safety Properties for Timed Petri Nets

Patrick Totzke

Edinburgh

2018-09-24

Universal Safety for Timed Petri Nets is PSPACE-complete

Parosh Aziz Abdulla
Uppsala University, Sweden

Mohamed Faouzi Atig
Uppsala University, Sweden

Radu Ciobanu
University of Edinburgh, UK

Richard Mayr
University of Edinburgh, UK

Patrick Totzke
University of Edinburgh, UK
https://orcid.org/0000-0001-5274-8190

Abstract

A timed network consists of an arbitrary number of initially identical 1-clock timed automata, interacting via hand-shake communication. In this setting there is no unique central controller, since all automata are initially identical. We consider the universal safety problem for such controller-less timed networks, i.e., verifying that a bad event (enabling some given transition) is impossible regardless of the size of the network.

This universal safety problem is dual to the existential coverability problem for timed-arc Petri nets, i.e., does there exist a number m of tokens, such that starting with m tokens in a given place, and none in the other places, some given transition is eventually enabled.

We show that these problems are PSPACE-complete.

2012 ACM Subject Classification Theory of computation → Timed and hybrid models

Keywords and phrases timed networks, safety checking, Petri nets, coverability

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2018.6

Funding This work was supported by the EPSRC, grant EP/M027651/1.

1 Introduction

Networks of Timed Automata

- a (finite-state) control program ($\mathcal{S}$)
- K indistinguishable d-clock Timed Automata ($\mathcal{C}$)

Networks of Timed Automata

- a (finite-state) control program ($\mathcal{S}$)
- K indistinguishable d-clock Timed Automata ($\mathcal{C}$)
- global time ticks + handshake communication.

Networks of Timed Automata

- a (finite-state) control program ($\mathcal{S}$)
- K indistinguishable d-clock Timed Automata ($\mathcal{C}$)
- global time ticks + handshake communication.

Q: Suppose all K clients are identically initialized.
Can one of the clients reach a failure state?

Networks of Timed Automata

- a (finite-state) control program ($\mathcal{S}$)
- K indistinguishable d-clock Timed Automata ($\mathcal{C}$)
- global time ticks + handshake communication.

Q: Suppose all K clients are identically initialized.
Can one of the clients reach a failure state?
A: Simple! It's a big ($K * d$)-clock TA! [AD94]

Networks of Timed Automata

- a (finite-state) control program ($\mathcal{S}$)
- K indistinguishable d-clock Timed Automata ($\mathcal{C}$)
- global time ticks + handshake communication.

Q: Suppose all K clients are identically initialized.
Can one of the clients reach a failure state?
A: Simple! It's a big $(K * d)$-clock TA! [AD94]

Q: What if the number K of clients is a parameter?
Does there exist K such that $\mathcal{S} \times \mathcal{C}^K$ can fail?

Networks of Timed Automata

- a (finite-state) control program ($\mathcal{S}$)
- K indistinguishable d-clock Timed Automata ($\mathcal{C}$)
- global time ticks + handshake communication.

Q: Suppose all K clients are identically initialized.
Can one of the clients reach a failure state?
A: Simple! It's a big ($K * d$)-clock TA! [AD94]

Q: What if the number K of clients is a parameter?
Does there exist K such that $\mathcal{S} \times \mathcal{C}^K$ can fail?
A: Not so simple:

- decidable for $d = 1$ [AJ03];
- undecidable for $d \geq 2$ [ADM04].

Networks of Timed Automata

- a (finite-state) control program ($\mathcal{S}$)
- K indistinguishable d-clock Timed Automata ($\mathcal{C}$)
- global time ticks + handshake communication.

Q: Suppose all K clients are identically initialized.
Can one of the clients reach a failure state?
A: Simple! It's a big ($K * d$)-clock TA! [AD94]

Q: What if the number K of clients is a parameter?
Does there exist K such that $\mathcal{S} \times \mathcal{C}^K$ can fail?
A: Not so simple:
- decidable for $d = 1$ [AJ03];
- undecidable for $d \geq 2$ [ADM04].

Q: What if there is no controller? Exists K so that $\mathcal{C}^K$ can fail?

Networks of Timed Automata

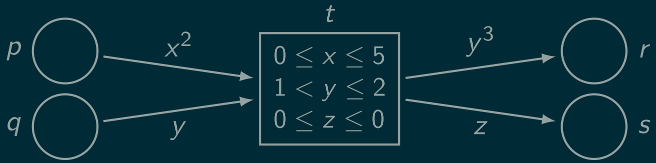
- a (finite-state) control program ($\mathcal{S}$)
- K indistinguishable d-clock Timed Automata ($\mathcal{C}$)
- global time ticks + handshake communication.

Q: Suppose all K clients are identically initialized.
Can one of the clients reach a failure state?
A: Simple! It's a big ($K * d$)-clock TA! [AD94]

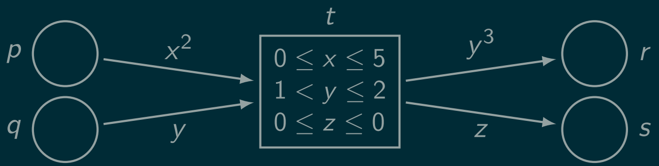
Q: What if the number K of clients is a parameter?
Does there exist K such that $\mathcal{S} \times \mathcal{C}^K$ can fail?
A: Not so simple:
- decidable for $d = 1$ [AJ03];
- undecidable for $d \geq 2$ [ADM04].

Q: What if there is no controller? Exists K so that $\mathcal{C}^K$ can fail?
A: PSPACE-complete [for $d = 1$, This paper]

Timed Petri Nets

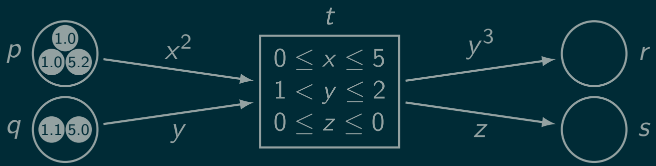


Timed Petri Nets



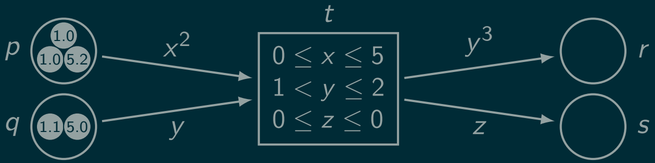
- Configurations are finite multisets over $P \times \mathbb{R}^d$

Timed Petri Nets



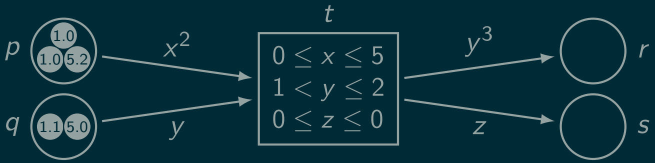
- Configurations are finite multisets over $P \times \mathbb{R}^d$

Timed Petri Nets



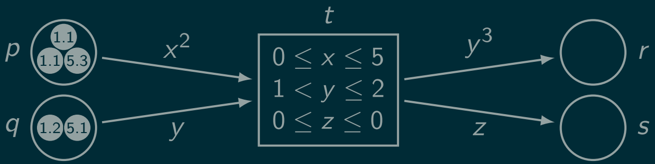
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks

Timed Petri Nets



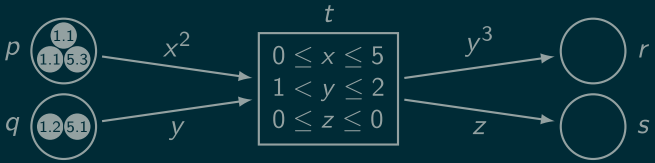
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$

Timed Petri Nets



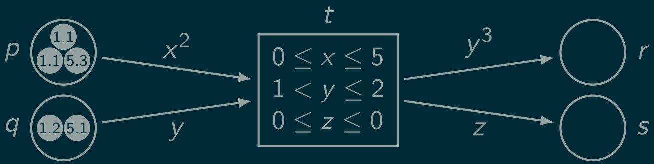
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$

Timed Petri Nets



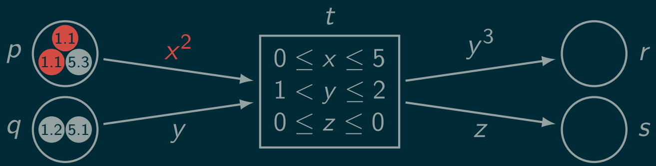
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$ and discrete steps

Timed Petri Nets



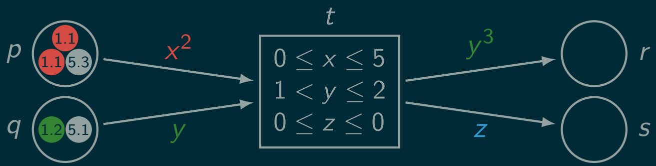
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$ and discrete steps $\xrightarrow{t}$

Timed Petri Nets



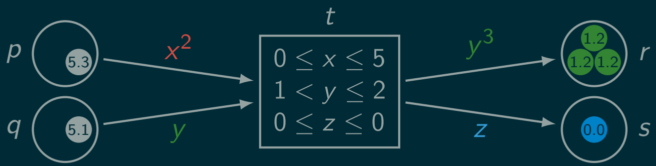
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$ and discrete steps $\xrightarrow{t}$

Timed Petri Nets



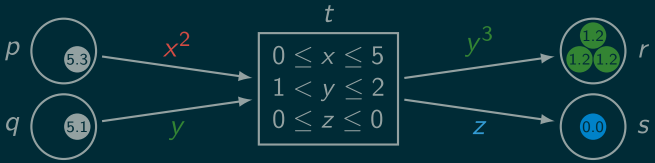
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$ and discrete steps $\xrightarrow{t}$

Timed Petri Nets



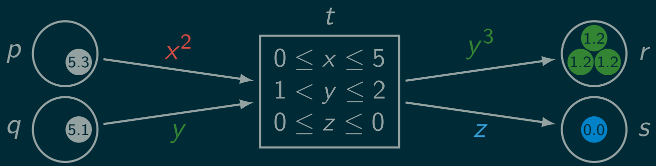
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$ and discrete steps $\xrightarrow{t}$

Timed Petri Nets



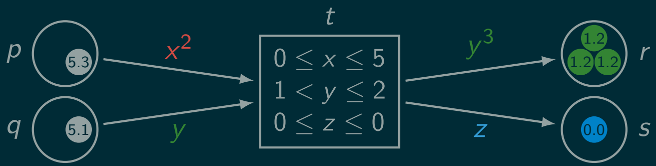
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$ and discrete steps $\xrightarrow{t}$
- Reachability, Coverability, Boundedness $(\ldots)$ undecidable

Timed Petri Nets



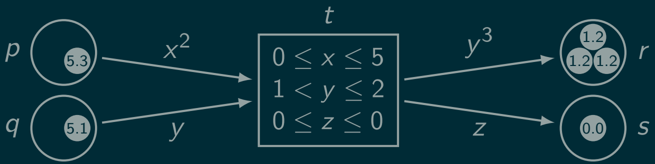
- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$ and discrete steps $\xrightarrow{t}$
- Reachability, Coverability, Boundedness $(\ldots)$ undecidable
- Coverability: $M \xrightarrow{*}\xrightarrow{t}$?
    - decidable via WSTS if $d = 1$
    - inter-reducible to Coverability for *ordered data nets*

Timed Petri Nets

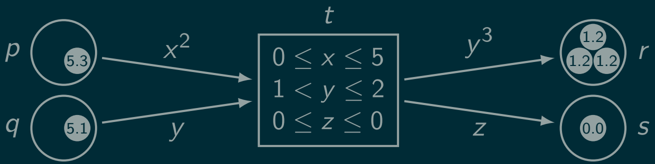


- Configurations are finite multisets over $P \times \mathbb{R}^d$
- global time ticks $\xrightarrow{0.1}$ and discrete steps $\xrightarrow{t}$
- Reachability, Coverability, Boundedness $(\dots)$ undecidable
- Coverability: $M \xrightarrow{*}\xrightarrow{t}$?
  - decidable via WSTS if $d = 1$
  - inter-reducible to Coverability for *ordered data nets*
- Existential Coverability: $\exists n \in \mathbb{N}$ with $M \cdot n \xrightarrow{*}\xrightarrow{t}$?

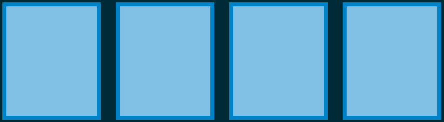
Coverability via Regions



Coverability via Regions

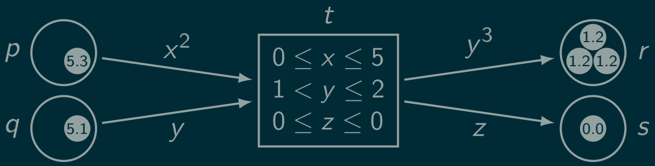


The *Region* of this configuration is a sequence

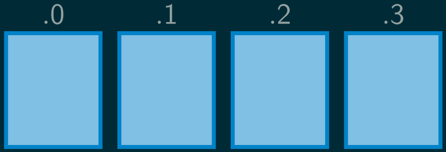


of multisets over $P \times \{0, \ldots, c_{\max}\}$, one for each fractional value*

Coverability via Regions

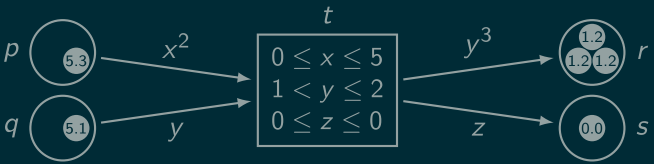


The *Region* of this configuration is a sequence

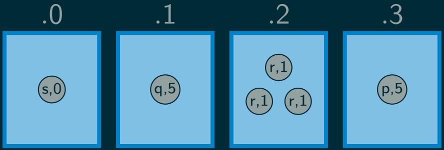


of multisets over $P \times \{0, \ldots, c_{\max}\}$, one for each fractional value*

Coverability via Regions

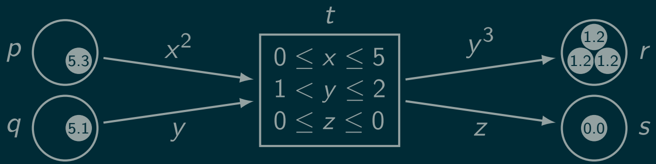


The *Region* of this configuration is a sequence

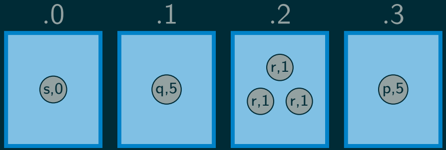


of multisets over $P \times \{0, \ldots, c_{\max}\}$, one for each fractional value*

Coverability via Regions

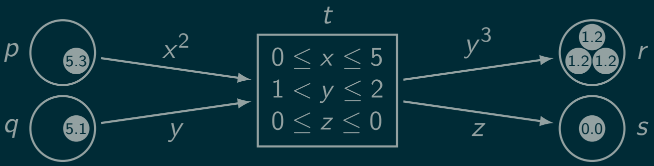


The *Region* of this configuration is a sequence

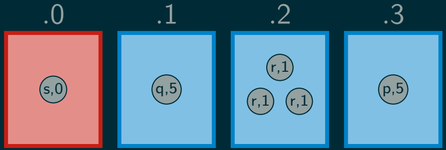


of multisets over $P \times \{0, \ldots, c_{\max}\}$, one for each fractional value*

Coverability via Regions

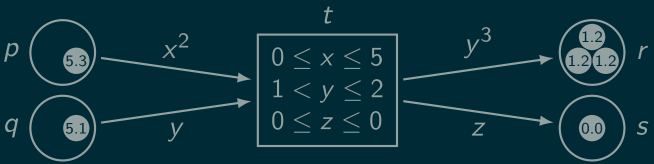


The *Region* of this configuration is a sequence

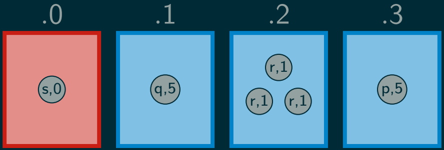


of multisets over $P \times \{0, \ldots, c_{\max}\}$, one for each fractional value*

Coverability via Regions



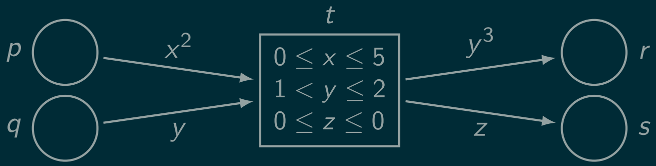
The *Region* of this configuration is a sequence



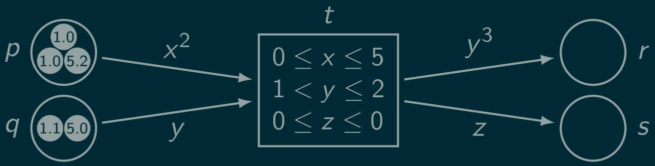
of multisets over $P \times \{0, \ldots, c_{\max}\}$, one for each fractional value*

Coverability via Regions

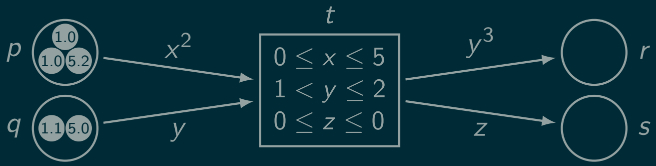
Coverability via Regions



Coverability via Regions

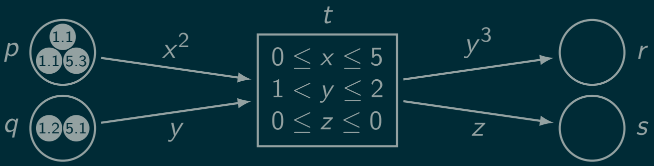


Coverability via Regions

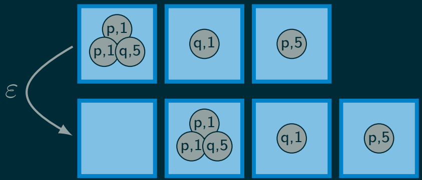


- short time ticks

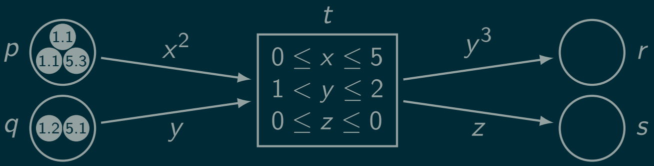
Coverability via Regions



- short time ticks

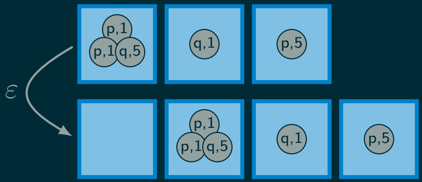


Coverability via Regions

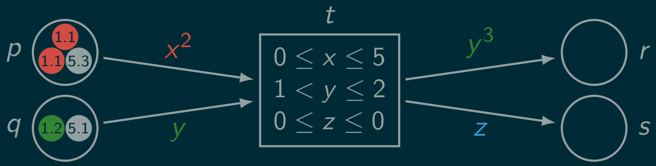


- short time ticks

- discrete transition firing

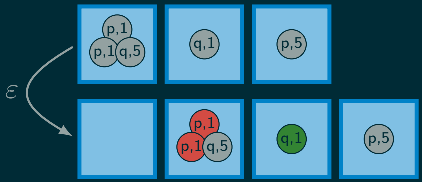


Coverability via Regions

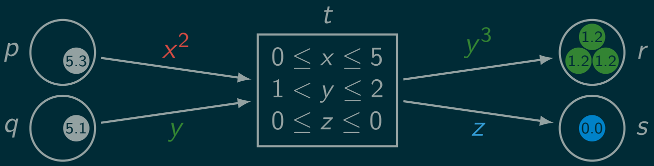


- short time ticks

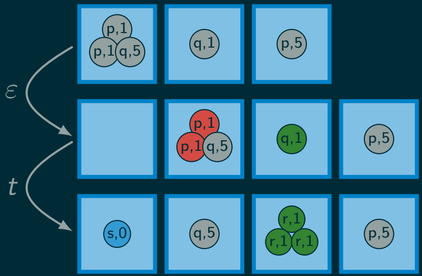
- discrete transition firing



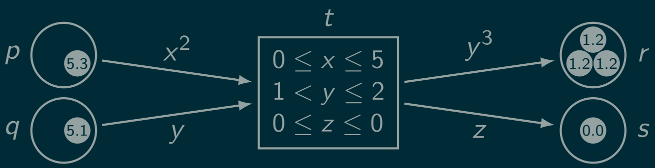
Coverability via Regions



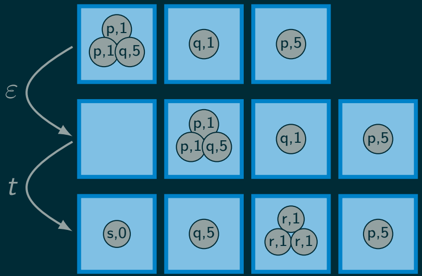
- short time ticks
- discrete transition firing



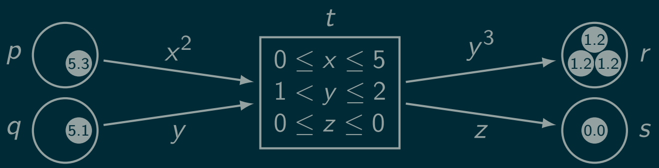
Coverability via Regions



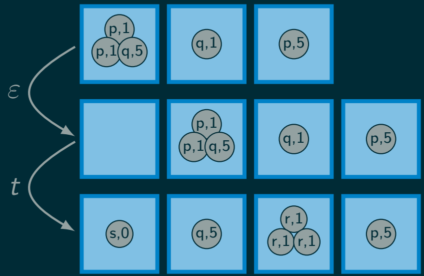
- short time ticks
- discrete transition firing



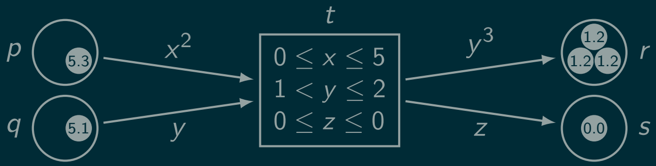
Coverability via Regions



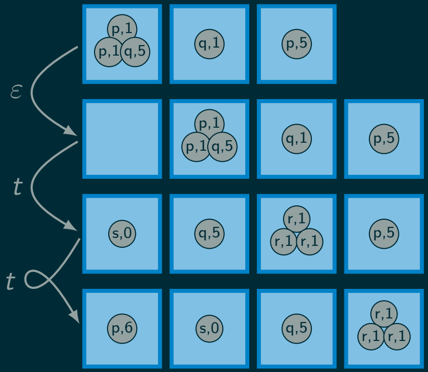
- short time ticks
- discrete transition firing
- long time ticks



Coverability via Regions



- short time ticks
- discrete transition firing
- long time ticks



Coverability via Regions

Obs. 1

- region equality is a time-abstract bisimulation
- unlike for TA, it has infinite index

Coverability via Regions

Obs. 1

- region equality is a time-abstract bisimulation
- unlike for TA, it has infinite index

Obs. 2

- steps between regions are monotone wrt. region embedding
- embedding is a well-quasi-order

Coverability via Regions

Obs. 1

- region equality is a time-abstract bisimulation
- unlike for TA, it has infinite index

Obs. 2

- steps between regions are monotone wrt. region embedding
- embedding is a well-quasi-order

Together, this yields decidability via the WSTS approach
(and completeness for $\mathcal{F}_{\omega^\omega}$).

Coverability via Regions

Obs. 1

- region equality is a time-abstract bisimulation
- unlike for TA, it has infinite index

Obs. 2

- steps between regions are monotone wrt. region embedding
- embedding is a well-quasi-order

Together, this yields decidability via the WSTS approach
(and completeness for $\mathcal{F}_{\omega^\omega}$).

NB: this fails for $d \geq 2$, for several reasons... Indeed we have undecidability in general.

Existential Coverability

In: A TPN, a marking M, a transition t

Question: Does there exist $\exists n \in \mathbb{N}$ with $M \cdot n \xrightarrow{*} \xrightarrow{t}$?

Existential Coverability

In: A TPN, a marking M, a transition t

Question: Does there exist $\exists n \in \mathbb{N}$ with $M \cdot n \xrightarrow{*}\xrightarrow{t}$?

$\approx$ parametrized safety checking Networks of Timed Automata

Existential Coverability

In: A TPN, a marking M, a transition t

Question: Does there exist $\exists n \in \mathbb{N}$ with $M \cdot n \xrightarrow{*} \xrightarrow{t}$?

- $\approx$ parametrized safety checking Networks of Timed Automata
- $\approx$ Coverability for TPN with continuous firing semantics á la Haddad et al.'17.

Existential Coverability

In: A TPN, a marking M, a transition t

Question: Does there exist $\exists n \in \mathbb{N}$ with $M \cdot n \xrightarrow{*} \xrightarrow{t}$?

- ≈ parametrized safety checking Networks of Timed Automata
- ≈ Coverability for TPN with continuous firing semantics á la Haddad et al.'17.
- – (logspace) reduces to Coverability

Existential Coverability

In: A TPN, a marking M, a transition t

Question: Does there exist $\exists n \in \mathbb{N}$ with $M \cdot n \xrightarrow{*}\xrightarrow{t}$?

- ≈ parametrized safety checking Networks of Timed Automata
- ≈ Coverability for TPN with continuous firing semantics á la Haddad et al.'17.
- – (logspace) reduces to Coverability
- – We show PSPACE-completeness

Existential Coverability

In: A TPN, a marking M, a transition t

Question: Does there exist $\exists n \in \mathbb{N}$ with $M \cdot n \xrightarrow{*}\xrightarrow{t}$?

- ≈ parametrized safety checking Networks of Timed Automata
- ≈ Coverability for TPN with continuous firing semantics á la Haddad et al.'17.
- – (logspace) reduces to Coverability
- – We show PSPACE-completeness

Existential Coverability

In: A TPN, a marking M, a transition t

Question: Does there exist $\exists n \in \mathbb{N}$ with $M \cdot n \xrightarrow{*} \xrightarrow{t}$?

≈ parametrized safety checking Networks of Timed Automata

≈ Coverability for TPN with continuous firing semantics á la Haddad et al.'17.

– (logspace) reduces to Coverability

– We show PSPACE-completeness

LB: iterated monotone circuits

Existential Coverability

In: A TPN, a marking M, a transition t

Question: Does there exist $\exists n \in \mathbb{N}$ with $M \cdot n \xrightarrow{*}\xrightarrow{t}$?

- $\approx$ parametrized safety checking Networks of Timed Automata
- $\approx$ Coverability for TPN with continuous firing semantics á la Haddad et al.'17.
- – (logspace) reduces to Coverability
- – We show PSPACE-completeness

  LB: iterated monotone circuits

  UB: Regions + forward acceleration

Existential Coverability

Q: What's different compared to Coverability?

Existential Coverability

Q: What's different compared to Coverability?
A: Token multiplicities do not matter. So,

Existential Coverability

Q: What's different compared to Coverability?
A: Token multiplicities do not matter. So,

1. A *Region* a sequence of multisets over $\Sigma \stackrel{\text{def}}{=} P \times \{0, \ldots, c_{\text{max}}\}$

Existential Coverability

Q: What's different compared to Coverability?
A: Token multiplicities do not matter. So,

1. A *Region* a sequence of ~~multisets over~~ sets $S \subseteq \Sigma$ $\Sigma \stackrel{\text{def}}{=} P \times \{0, \ldots, c_{\text{max}}\}$

Existential Coverability

Q: What's different compared to Coverability?
A: Token multiplicities do not matter. So,

1. A *Region* a sequence of ~~multisets over~~ $\Sigma \stackrel{\text{def}}{=} P \times \{0, \ldots, c_{\text{max}}\}$
   sets $S \subseteq \Sigma$

   This already improves the upper bound to $\mathcal{F}_\omega$

Existential Coverability

Q: What's different compared to Coverability?
A: Token multiplicities do not matter. So,

1. A *Region* a sequence of ~~multisets over~~ $\Sigma \stackrel{\text{def}}{=} P \times \{0, \ldots, c_{\text{max}}\}$
   sets $S \subseteq \Sigma$
   This already improves the upper bound to $\mathcal{F}_\omega$
2. Wlog., the net is *non-consuming*: ${}^\bullet t \subseteq t^\bullet$ for all transitions t.

Existential Coverability

Q: What's different compared to Coverability?
A: Token multiplicities do not matter. So,

1. A *Region* a sequence of ~~multisets over~~ $\Sigma \stackrel{\text{def}}{=} P \times \{0, \ldots, c_{\max}\}$
   sets $S \subseteq \Sigma$

   This already improves the upper bound to $\mathcal{F}_\omega$

2. Wlog., the net is *non-consuming*: ${}^\bullet t \subseteq t^\bullet$ for all transitions t. This means that discrete transition firing is non-decreasing and for every region R

   - there is a unique maximal region R' with $R \xrightarrow{disc}{}^* R'$
   - R' is (Ptime) computable

Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”

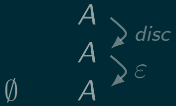
A

Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"

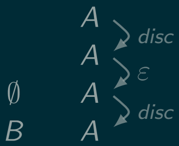
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



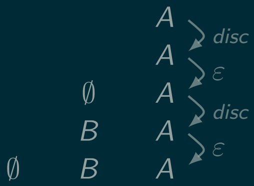
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



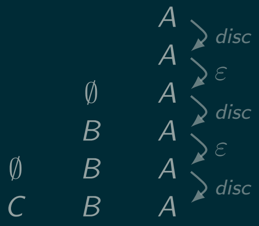
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



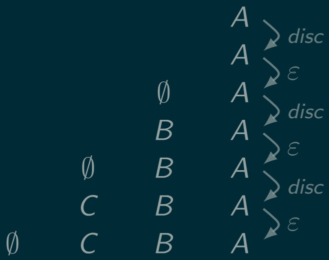
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"



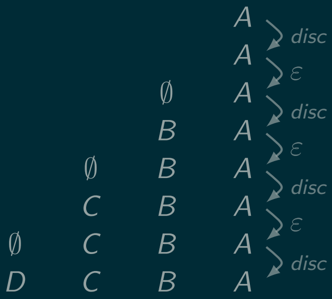
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"



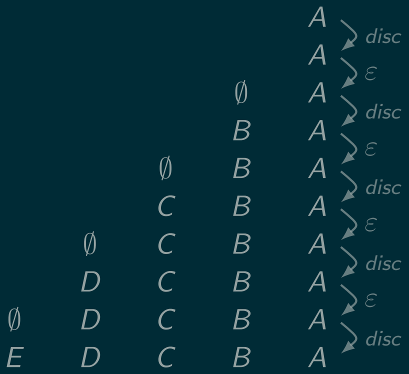
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"

				A	$\downarrow$ *disc*
				A	$\downarrow$ ε
			$\emptyset$	A	$\downarrow$ *disc*
			B	A	$\downarrow$ ε
		$\emptyset$	B	A	$\downarrow$ *disc*
		C	B	A	$\downarrow$ ε
	$\emptyset$	C	B	A	$\downarrow$ *disc*
	D	C	B	A	$\downarrow$ ε
$\emptyset$	D	C	B	A	

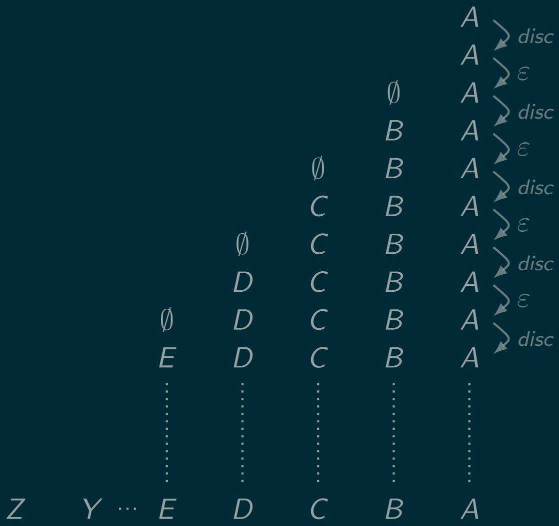
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"



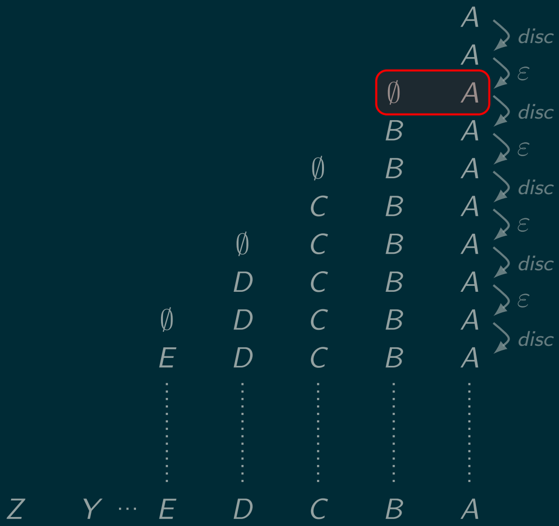
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"



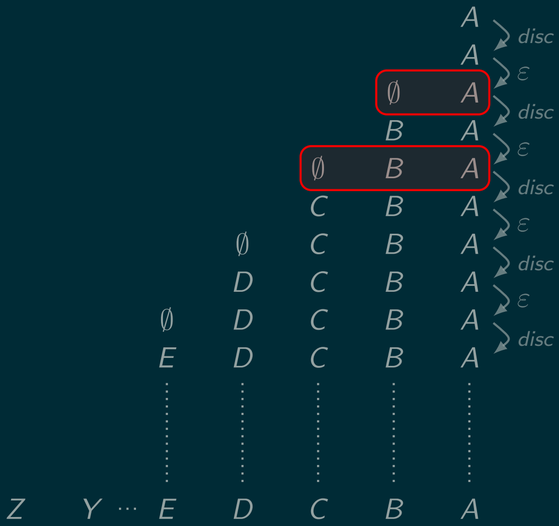
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



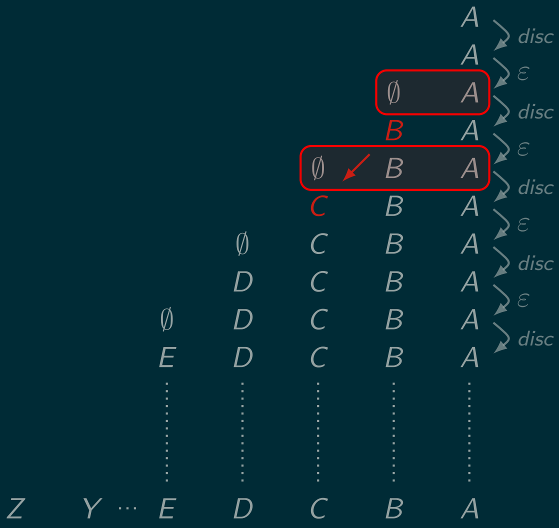
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"



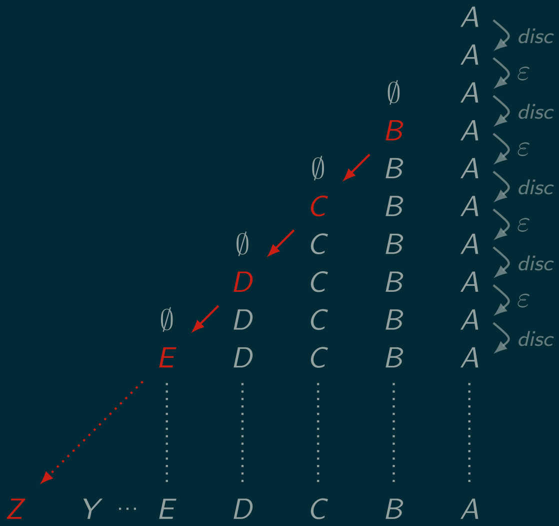
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"



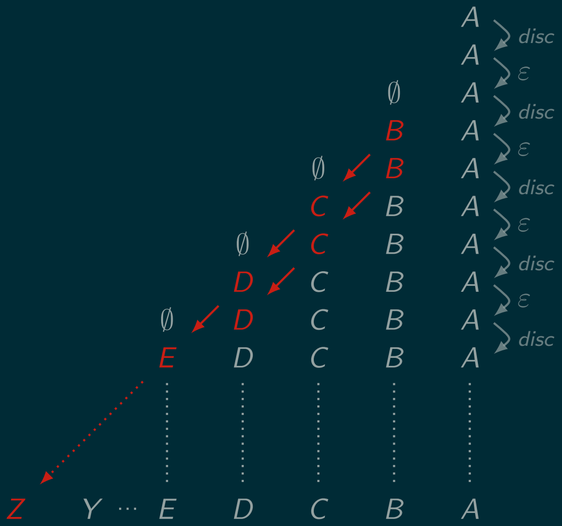
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



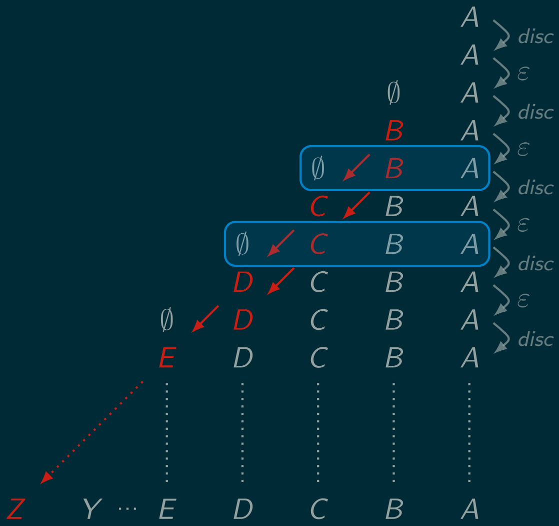
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



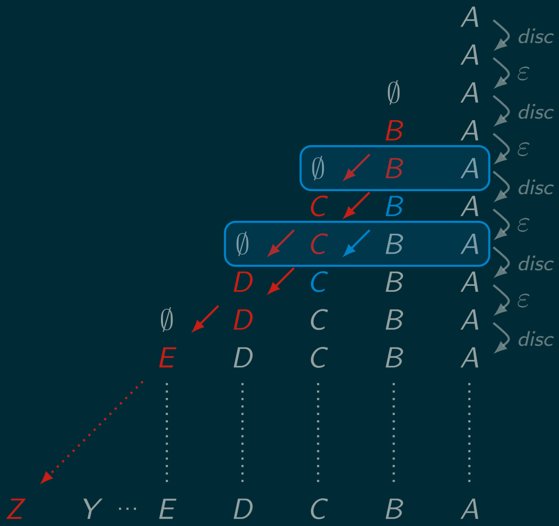
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



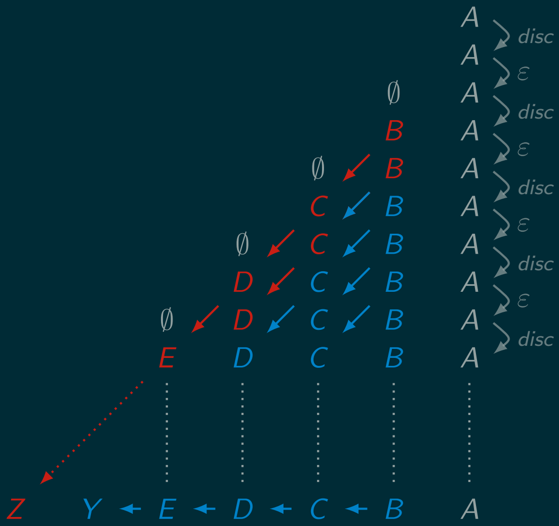
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



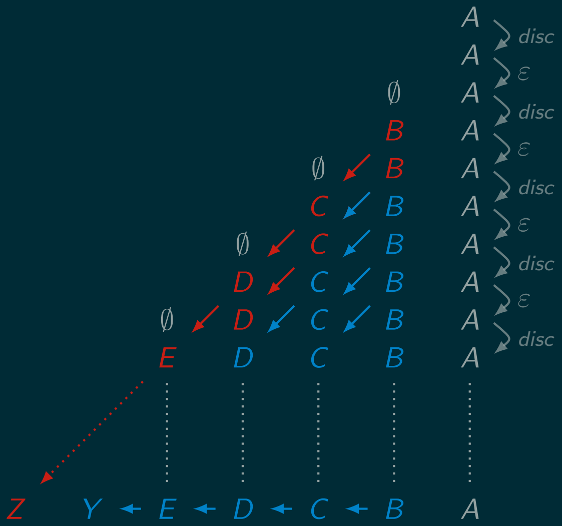
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"



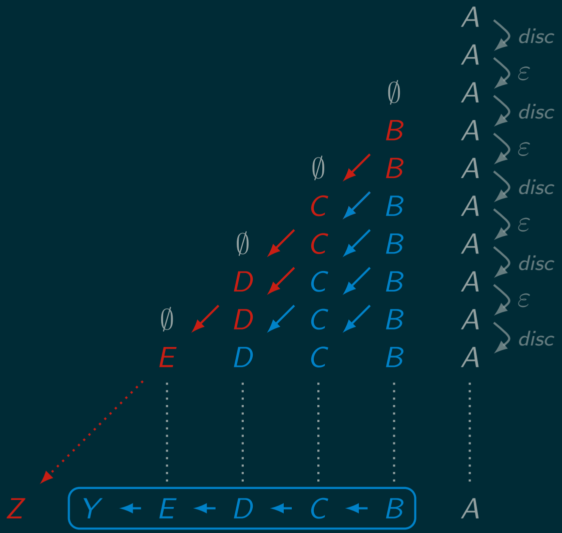
Existential Coverability: Key Observation

When forward exploring zeno behaviour regions “stabilize”



Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"

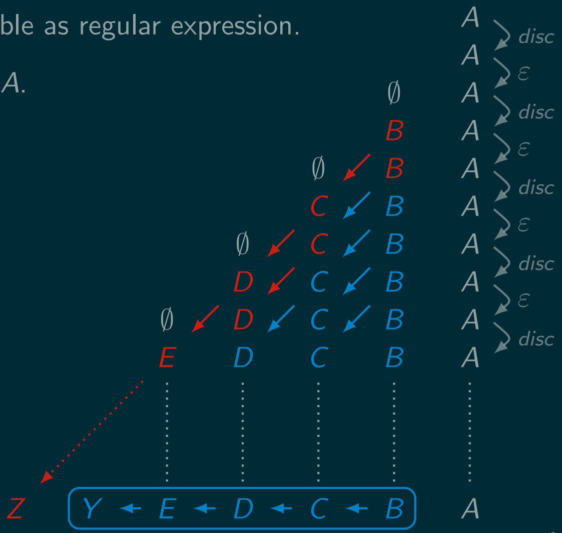


Existential Coverability: Key Observation

When forward exploring zeno behaviour regions "stabilize"

and the limit is expressible as regular expression.

In this example as ZY^*A.



Existential Coverability: Construction

- use regular expressions over 2^{Σ} to represent (limit) regions
- careful forward exploration, using intermediate compression steps that add Kleene *s

Forward Exploration

Forward Exploration

x_1 x_0^* start

Forward Exploration

x_1	x_0^*	start
x_1^1	$(x_0^1)^*$	saturate

Forward Exploration

	x_1	x_0^*	start
	x_1^1	$(x_0^1)^*$	saturate
$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate

Forward Exploration

	x_1	x_0^*	start
	x_1^1	$(x_0^1)^*$	saturate
$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
x_2^2	x_1^2	$(x_0^2)^*$	saturate

Forward Exploration

		x_1	x_0^*	start
		x_1^1	$(x_0^1)^*$	saturate
	$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
	x_2^2	x_1^2	$(x_0^2)^*$	saturate
$(x_0^2 + 1)$	x_2^2	x_1^2	$(x_0^2)^*$	rotate

Forward Exploration

		x_1	x_0^*	start
		x_1^1	$(x_0^1)^*$	saturate
	$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
	x_2^2	x_1^2	$(x_0^2)^*$	saturate
$(x_0^2 + 1)$	x_2^2	x_1^2	$(x_0^2)^*$	rotate
x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	saturate

Forward Exploration

			x_1	x_0^*	start
			x_1^1	$(x_0^1)^*$	saturate
		(x_0^1+1)	x_1^1	$(x_0^1)^*$	rotate
		x_2^2	x_1^2	$(x_0^2)^*$	saturate
	(x_0^2+1)	x_2^2	x_1^2	$(x_0^2)^*$	rotate
	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	saturate
(x_0^3+1)	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	rotate

Forward Exploration

			x_1	x_0^*	start
			x_1^1	$(x_0^1)^*$	saturate
		$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
		x_2^2	x_1^2	$(x_0^2)^*$	saturate
	$(x_0^2 + 1)$	x_2^2	x_1^2	$(x_0^2)^*$	rotate
	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	saturate
$(x_0^3 + 1)$	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	rotate
x_4^4	x_3^4	x_2^4	x_1^4	$(x_0^4)^*$	saturate

Forward Exploration

				x_1	x_0^*	start
				x_1^1	$(x_0^1)^*$	saturate
			$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
			x_2^2	x_1^2	$(x_0^2)^*$	saturate
		$(x_0^2 + 1)$	x_2^2	x_1^2	$(x_0^2)^*$	rotate
		x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	saturate
	$(x_0^3 + 1)$	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	rotate
	x_4^4	x_3^4	x_2^4	x_1^4	$(x_0^4)^*$	saturate

Forward Exploration

			x_1	x_0^*	start
			x_1^1	$(x_0^1)^*$	saturate
		$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
		x_2^2	x_1^2	$(x_0^2)^*$	saturate
	$(x_0^2 + 1)$	x_2^2	x_1^2	$(x_0^2)^*$	rotate
	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	saturate
$(x_0^3 + 1)$	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	rotate
x_4^4	x_3^4	x_2^4	x_1^4	$(x_0^4)^*$	saturate

Forward Exploration

			x_1	x_0^*	start
			x_1^1	$(x_0^1)^*$	saturate
		$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
		x_2^2	x_1^2	$(x_0^2)^*$	saturate
	$(x_0^2 + 1)$	x_2^2	x_1^2	$(x_0^2)^*$	rotate
	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	saturate
$(x_0^3 + 1)$	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	rotate
x_4^4	x_3^4	x_2^4	x_1^4	$(x_0^4)^*$	saturate

Forward Exploration

			x_1	x_0^*	start
			x_1^1	$(x_0^1)^*$	saturate
		$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
		x_2^2	x_1^2	$(x_0^2)^*$	saturate
	$(x_0^2 + 1)$	x_2^2	x_1^2	$(x_0^2)^*$	rotate
	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	saturate
$(x_0^3 + 1)$	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	rotate
x_4^4	x_3^4	x_2^4	x_1^4	$(x_0^4)^*$	saturate

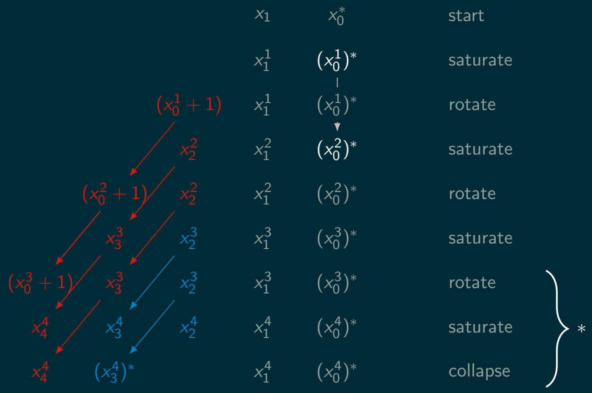
Forward Exploration

			x_1	x_0^*	start
			x_1^1	$(x_0^1)^*$	saturate
		$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
		x_2^2	x_1^2	$(x_0^2)^*$	saturate
	$(x_0^2 + 1)$	x_2^2	x_1^2	$(x_0^2)^*$	rotate
	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	saturate
$(x_0^3 + 1)$	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	rotate
x_4^4	x_3^4	x_2^4	x_1^4	$(x_0^4)^*$	saturate

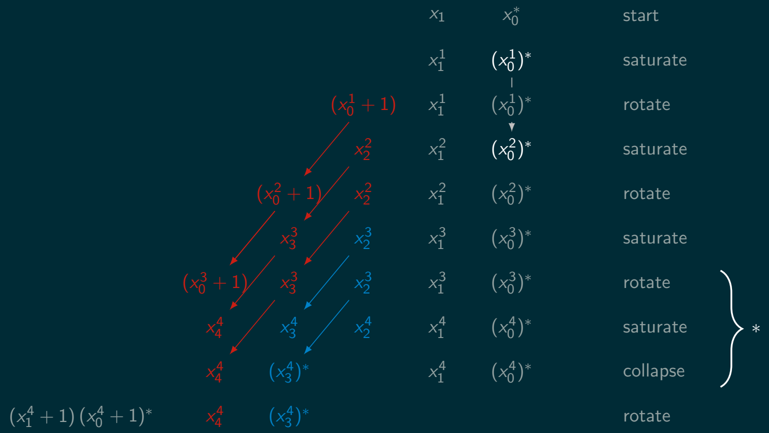
Forward Exploration

			x_1	x_0^*	start
			x_1^1	$(x_0^1)^*$	saturate
		$(x_0^1 + 1)$	x_1^1	$(x_0^1)^*$	rotate
		x_2^2	x_1^2	$(x_0^2)^*$	saturate
	$(x_0^2 + 1)$	x_2^2	x_1^2	$(x_0^2)^*$	rotate
	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	saturate
$(x_0^3 + 1)$	x_3^3	x_2^3	x_1^3	$(x_0^3)^*$	rotate
x_4^4	x_3^4	x_2^4	x_1^4	$(x_0^4)^*$	saturate
x_4^4	$(x_3^4)^*$		x_1^4	$(x_0^4)^*$	collapse

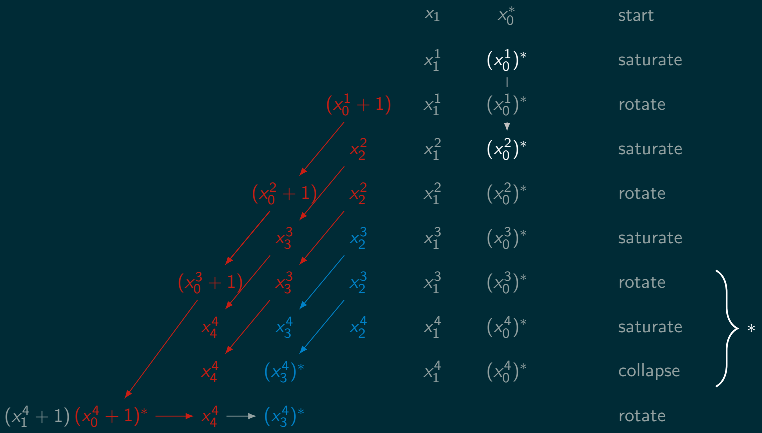
Forward Exploration



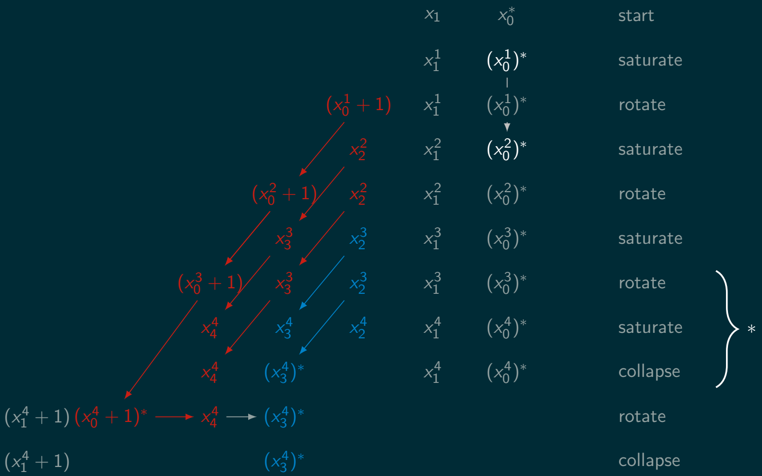
Forward Exploration



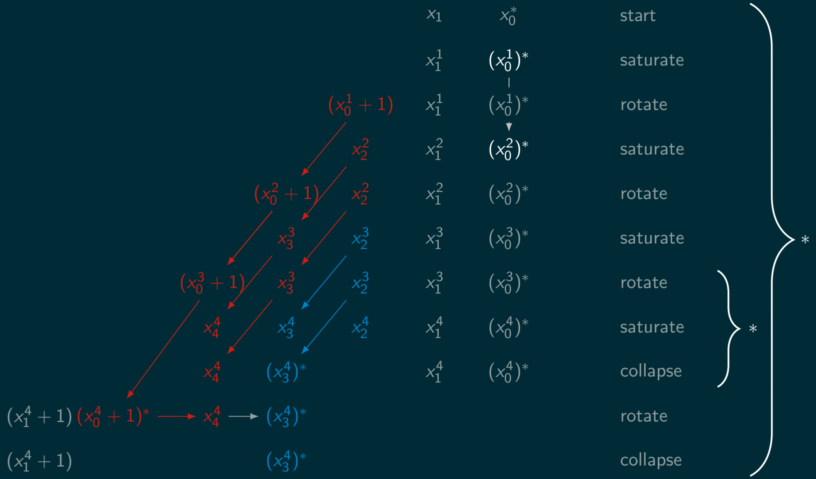
Forward Exploration



Forward Exploration



Forward Exploration



Existential Coverability: Construction

- use regular expressions over 2^{Σ} to represent (limit) regions
- careful forward exploration, using intermediate compression steps that add Kleene *s

Existential Coverability: Construction

- use regular expressions over 2^{Σ} to represent (limit) regions
- careful forward exploration, using intermediate compression steps that add Kleene *s

Properties

- Computes the set of coverable regions

Existential Coverability: Construction

- use regular expressions over 2^{Σ} to represent (limit) regions
- careful forward exploration, using intermediate compression steps that add Kleene *s

Properties

- Computes the set of coverable regions
- does not need nondeterministic branching

Existential Coverability: Construction

- use regular expressions over 2^{Σ} to represent (limit) regions
- careful forward exploration, using intermediate compression steps that add Kleene *s

Properties

- Computes the set of coverable regions
- does not need nondeterministic branching
- every explored RE has length ≤ 5 .

Existential Coverability: Construction

- use regular expressions over 2^{Σ} to represent (limit) regions
- careful forward exploration, using intermediate compression steps that add Kleene *s

Properties

- Computes the set of coverable regions
- does not need nondeterministic branching
- every explored RE has length ≤ 5 .

Existential Coverability: Construction

- use regular expressions over 2^{Σ} to represent (limit) regions
- careful forward exploration, using intermediate compression steps that add Kleene *s

Properties

- Computes the set of coverable regions
- does not need nondeterministic branching
- every explored RE has length ≤ 5 .

Corollary

- the sequence is singly exponential
- checking Existential Coverability is in PSPACE.

WIP 1: multi-dimensional ECOVER

WIP 1: multi-dimensional ECOVER

Conjecture

ECOVER is PSPACE-completeness for any fixed dimension d.

WIP 1: multi-dimensional ECOVER

Conjecture

ECOVER is PSPACE-completeness for any fixed dimension d.

- Regions become directed (hyper) graphs with edges labelled by subsets of $P \times [0 \ldots c_{\mathrm{max}} + 1]^d$

WIP 1: multi-dimensional ECOVER

Conjecture

ECOVER is PSPACE-completeness for any fixed dimension d.

- Regions become directed (hyper) graphs with edges labelled by subsets of $P \times [0 \ldots c_{\max} + 1]^d$
- Semantics of Kleene stars?

WIP 2: Coverability for TBPP

BPP nets: every transition consumes at most one token

WIP 2: Coverability for TBPP

BPP nets: every transition consumes at most one token

- $\sim$ context-free controlled TPN

WIP 2: Coverability for TBPP

BPP nets: every transition consumes at most one token

- $\sim$ context-free controlled TPN
- Coverability
  - in PSPACE for all d

WIP 2: Coverability for TBPP

BPP nets: every transition consumes at most one token

- $\sim$ context-free controlled TPN
- Coverability
  - in PSPACE for all d
    (lossy semantics $\rightsquigarrow$ witnesses visit only small regions)

WIP 2: Coverability for TBPP

BPP nets: every transition consumes at most one token

- $\sim$ context-free controlled TPN
- Coverability
    - in PSPACE for all d
      (lossy semantics $\rightsquigarrow$ witnesses visit only small regions)
    - conjecture: NP-complete for $d = 1$

WIP 2: Coverability for TBPP

BPP nets: every transition consumes at most one token

- $\sim$ context-free controlled TPN
- Coverability
  - in PSPACE for all d
    (lossy semantics $\rightsquigarrow$ witnesses visit only small regions)
  - conjecture: NP-complete for $d = 1$
- Reachability:
  - 1st step: bound ”time to kill” a region?

thank you.