Reachability via saturation

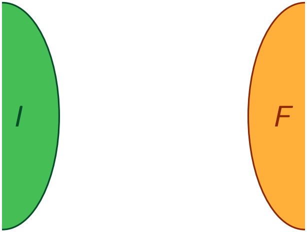
Gabriele Puppis

LaBRI / CNRS

Reachability is semi-decidable

A path connecting two sets, if exists, can be found in finitely many steps.

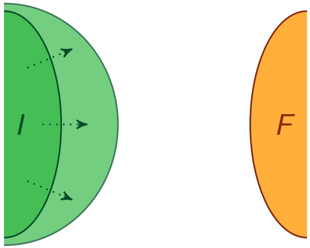
Forward analysis



Reachability is semi-decidable

A path connecting two sets, if exists, can be found in finitely many steps.

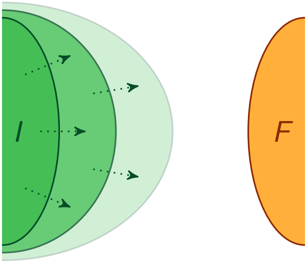
Forward analysis



Reachability is semi-decidable

A path connecting two sets, if exists, can be found in finitely many steps.

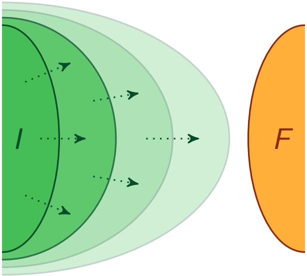
Forward analysis



Reachability is semi-decidable

A path connecting two sets, if exists, can be found in finitely many steps.

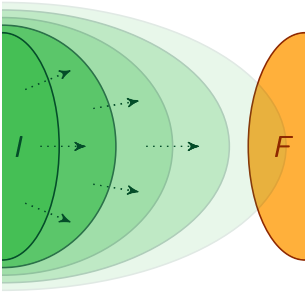
Forward analysis



Reachability is semi-decidable

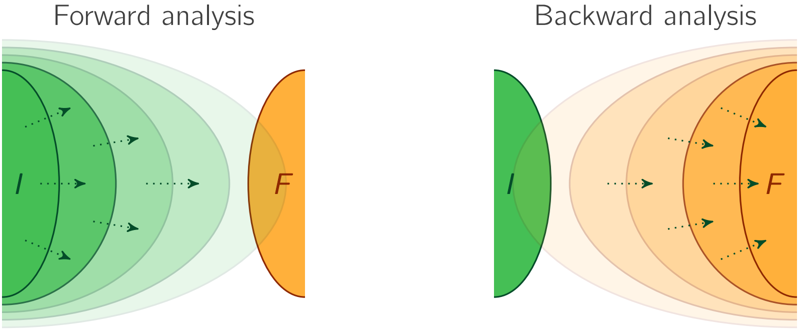
A path connecting two sets, if exists, can be found in finitely many steps.

Forward analysis



Reachability is semi-decidable

A path connecting two sets, if exists, can be found in finitely many steps.

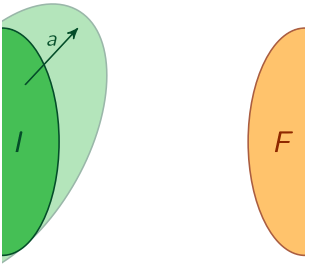


☞ The problem is of course termination, namely, to detect non-reachability...

Sometimes non-reachability can be checked effectively using "safe" **over-approximations** of reachable sets

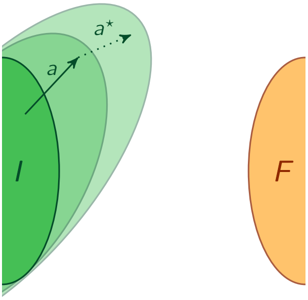
Sometimes non-reachability can be checked effectively using “safe” **over-approximations** of reachable sets

Acceleration / pumping



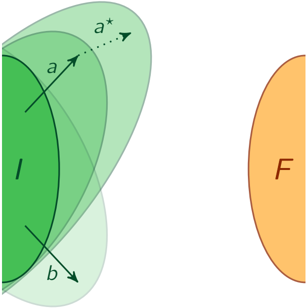
Sometimes non-reachability can be checked effectively using "safe" **over-approximations** of reachable sets

Acceleration / pumping



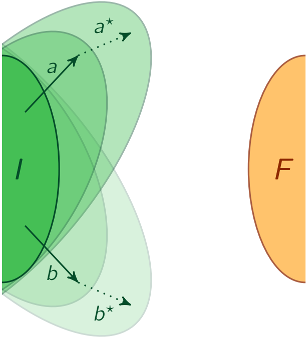
Sometimes non-reachability can be checked effectively using “safe” **over-approximations** of reachable sets

Acceleration / pumping



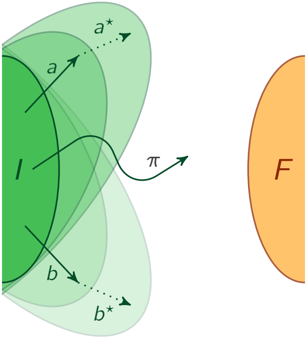
Sometimes non-reachability can be checked effectively using “safe” **over-approximations** of reachable sets

Acceleration / pumping



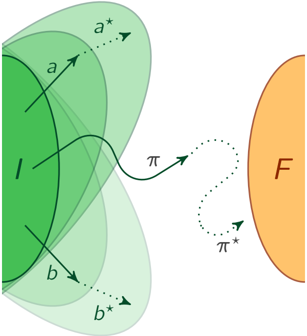
Sometimes non-reachability can be checked effectively using “safe” **over-approximations** of reachable sets

Acceleration / pumping



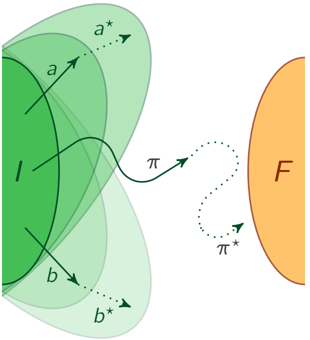
Sometimes non-reachability can be checked effectively using “safe” **over-approximations** of reachable sets

Acceleration / pumping

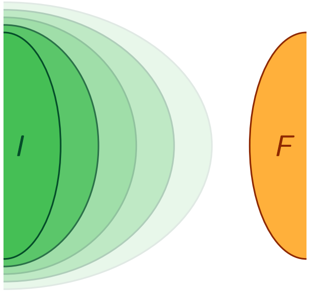


Sometimes non-reachability can be checked effectively using "safe" **over-approximations** of reachable sets

Acceleration / pumping

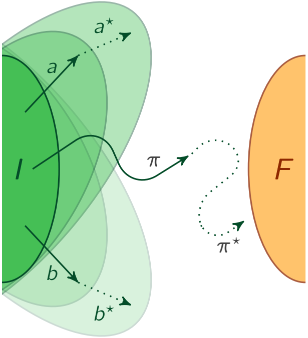


Invariant analysis

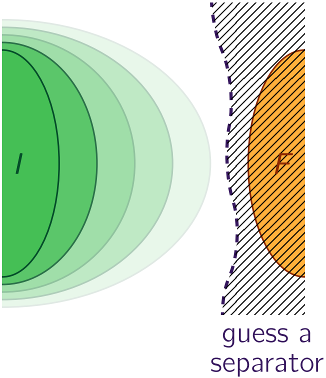


Sometimes non-reachability can be checked effectively using "safe" **over-approximations** of reachable sets

Acceleration / pumping

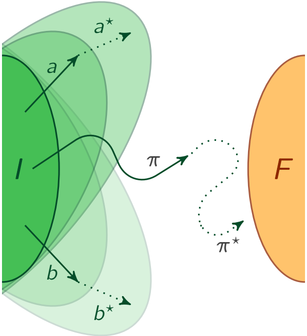


Invariant analysis

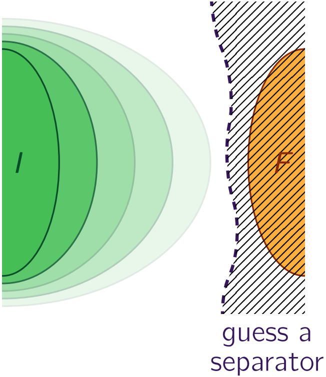


Sometimes non-reachability can be checked effectively
using "safe" **over-approximations** of reachable sets

Acceleration / pumping



Invariant analysis



☞ Both approaches require **symbolic representations** of infinite sets

Backward reachability for pushdown systems

Given a pushdown system $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta)$ and a set $B_0 \subseteq Q \cdot \Gamma^\star$ of target configurations, define:

$$B_{n+1} = B_n \cup \left\{ qz \;\middle|\; \exists q'z' \in B_n.\ \exists a \in \Sigma.\ qz \xrightarrow{a} q'z' \right\}$$

$$B_\omega = \bigcup_{n \in \mathbb{N}} B_n$$

Backward reachability for pushdown systems

Given a pushdown system $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta)$ and
a set $B_0 \subseteq Q \cdot \Gamma^\star$ of target configurations, define:

$$B_{n+1} \quad = \quad B_n \ \cup \ \left\{ qz \ \middle| \ \exists q'z' \in B_n.\ \exists a \in \Sigma.\ \boldsymbol{qz \xrightarrow{a} q'z'} \right\}$$

$$B_\omega \quad = \quad \bigcup\nolimits_{n \in \mathbb{N}} B_n$$

☞ B_ω contains the configurations from which one can reach B_0

Backward reachability for pushdown systems

Given a pushdown system $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta)$ and a set $B_0 \subseteq Q \cdot \Gamma^\star$ of target configurations, define:

$$B_{n+1} = B_n \cup \left\{ qz \;\middle|\; \exists q'z' \in B_n.\ \exists a \in \Sigma.\ \mathbf{qz} \xrightarrow{a} \mathbf{q'z'} \right\}$$

$$B_\omega = \bigcup_{n \in \mathbb{N}} B_n$$

B_ω contains the configurations from which one can reach B_0

B_ω is usually infinite, but is it perhaps regular?

Backward reachability for pushdown systems

Given a pushdown system $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta)$ and a set $B_0 \subseteq Q \cdot \Gamma^\star$ of target configurations, define:

$$B_{n+1} = B_n \cup \left\{ qz \,\middle|\, \exists q'z' \in B_n.\ \exists a \in \Sigma.\ qz \xrightarrow{a} q'z' \right\}$$

$$B_\omega = \bigcup_{n \in \mathbb{N}} B_n$$

B_ω contains the configurations from which one can reach B_0

B_ω is usually infinite, but is it perhaps regular?

Example

Consider the pushdown system

$B_0 = \{q\varepsilon\}$ $\quad B_1 = \{q\varepsilon, q\gamma\}$ $\quad B_2 = \{q\varepsilon, q\gamma, q\gamma\gamma\}$ $\quad \ldots$

Backward reachability for pushdown systems

Given a pushdown system $\mathcal{P} = (Q, \Sigma, \Gamma, \Delta)$ and a set $B_0 \subseteq Q \cdot \Gamma^\star$ of target configurations, define:

$$B_{n+1} = B_n \cup \left\{ qz \;\middle|\; \exists q'z' \in B_n.\ \exists a \in \Sigma.\ qz \xrightarrow{a} q'z' \right\}$$

$$B_\omega = \bigcup_{n \in \mathbb{N}} B_n$$

B_ω contains the configurations from which one can reach B_0

B_ω is usually infinite, but is it perhaps regular?

Example

Consider the pushdown system

$B_0 = \{q\varepsilon\}$ $\quad B_1 = \{q\varepsilon, q\gamma\}$ $\quad B_2 = \{q\varepsilon, q\gamma, q\gamma\gamma\}$ $\quad \ldots$

$B_\omega = q\varepsilon^\star$ is indeed regular, but how to efficiently compute it?

"Pump" the changes from B_n to B_{n+1} to obtain a new sequence $C_0, C_1, \ldots$ that converges more quickly:

$$
\begin{array}{lll}
\text{(completeness)} & \forall n \in \mathbb{N}. & \boldsymbol{B_n} \subseteq \boldsymbol{C_n} \\
\text{(soundness)} & \forall n \in \mathbb{N}. & \boldsymbol{C_n} \subseteq \boldsymbol{B_\omega} \\
\text{(termination)} & \exists n \in \mathbb{N}. & \boldsymbol{C_n} = \boldsymbol{C_{n+1}}
\end{array}
$$

the limit $\bigcup_{n \in \mathbb{N}} C_n$ coincides with B_ω

"Pump" the changes from B_n to B_{n+1} to obtain a new sequence C_0, C_1, ... that converges more quickly:

(completeness)	$\forall n \in \mathbb{N}. \quad \boldsymbol{B_n} \subseteq \boldsymbol{C_n}$
(soundness)	$\forall n \in \mathbb{N}. \quad \boldsymbol{C_n} \subseteq \boldsymbol{B_\omega}$
(termination)	$\exists n \in \mathbb{N}. \quad \boldsymbol{C_n} = \boldsymbol{C_{n+1}}$

the limit $\bigcup_{n \in \mathbb{N}} C_n$ coincides with B_ω

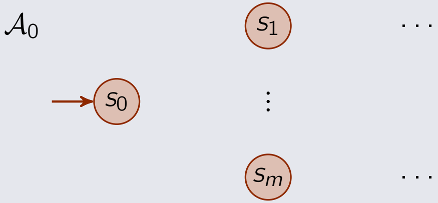
The sets C_0, C_1, ... will be defined by automata $\mathcal{A}_0, \mathcal{A}_1, \ldots$ sharing the **same state space**...

Initial conditions

- The pushdown system $\mathcal{P}$ has m states $q_1, \ldots, q_m$

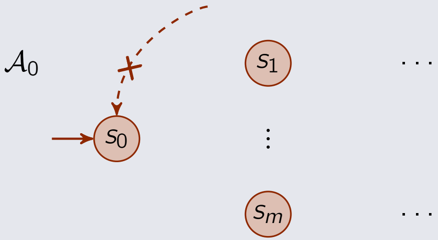
Initial conditions

- The pushdown system $\mathcal{P}$ has m states $q_1, \ldots, q_m$
- The automaton $\mathcal{A}_0$ recognizing $C_0 = B_0$ has a single initial non-final state s_0, m distinct states $s_1, \ldots, s_m$, and possibly other states



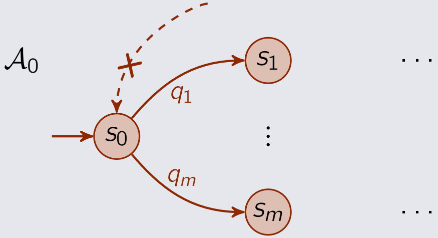
Initial conditions

- The pushdown system $\mathcal{P}$ has m states $q_1, \ldots, q_m$
- The automaton $\mathcal{A}_0$ recognizing $C_0 = B_0$ has a single initial non-final state s_0, m distinct states $s_1, \ldots, s_m$, and possibly other states
- No transition in $\mathcal{A}_0$ reaches the initial state s_0



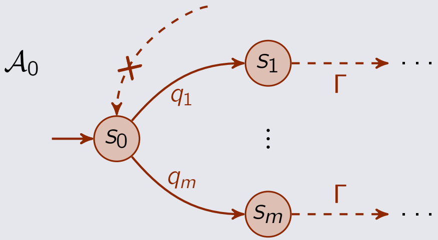
Initial conditions

- The pushdown system $\mathcal{P}$ has m states $q_1, \ldots, q_m$
- The automaton $\mathcal{A}_0$ recognizing $C_0 = B_0$ has a single initial non-final state s_0, m distinct states $s_1, \ldots, s_m$, and possibly other states
- No transition in $\mathcal{A}_0$ reaches the initial state s_0
- The unique q_i-labelled transition in $\mathcal{A}_0$ is (s_0, q_i, s_i)



Initial conditions

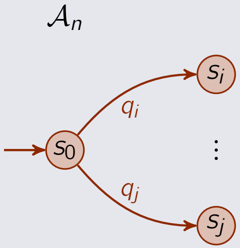
- The pushdown system $\mathcal{P}$ has m states $q_1, \ldots, q_m$
- The automaton $\mathcal{A}_0$ recognizing $C_0 = B_0$ has a single initial non-final state s_0, m distinct states $s_1, \ldots, s_m$, and possibly other states
- No transition in $\mathcal{A}_0$ reaches the initial state s_0
- The unique q_i-labelled transition in $\mathcal{A}_0$ is (s_0, q_i, s_i)
- The other transitions in $\mathcal{A}_0$ are labelled by stack symbols



Saturation procedure

Construct $\mathcal{A}_{n+1}$ from $\mathcal{A}_n$ by adding transitions, as follows:

1. select a transition rule $(q_i\gamma, a, q_j z)$ in the pushdown system $\mathcal{P}$



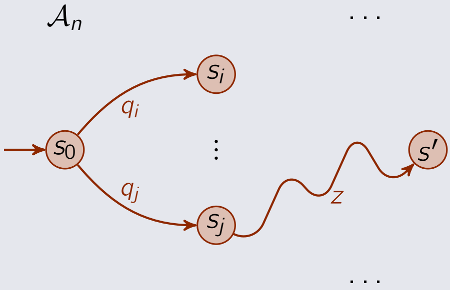
$\dots$

$\dots$

Saturation procedure

Construct $\mathcal{A}_{n+1}$ from $\mathcal{A}_n$ by adding transitions, as follows:

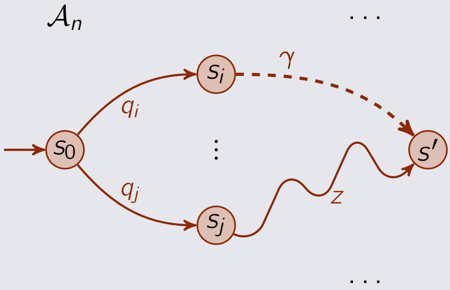
1. select a transition rule $(q_i\gamma, a, q_j z)$ in the pushdown system $\mathcal{P}$
2. select a state s' in $\mathcal{A}_n$ reachable from s_0 via a $q_j z$-labelled path



Saturation procedure

Construct $\mathcal{A}_{n+1}$ from $\mathcal{A}_n$ by adding transitions, as follows:

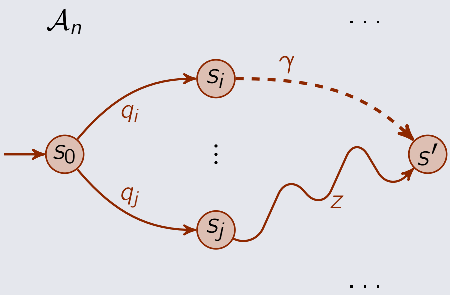
1. select a transition rule $(q_i\gamma, a, q_j z)$ in the pushdown system $\mathcal{P}$
2. select a state s' in $\mathcal{A}_n$ reachable from s_0 via a $q_j z$-labelled path
3. add transition (s_i, γ, s')



Saturation procedure

Construct $\mathcal{A}_{n+1}$ from $\mathcal{A}_n$ by adding transitions, as follows:

1. select a transition rule $(q_i\gamma, a, q_jz)$ in the pushdown system $\mathcal{P}$
2. select a state s' in $\mathcal{A}_n$ reachable from s_0 via a q_jz-labelled path
3. add transition (s_i, γ, s')



Termination: straightforward

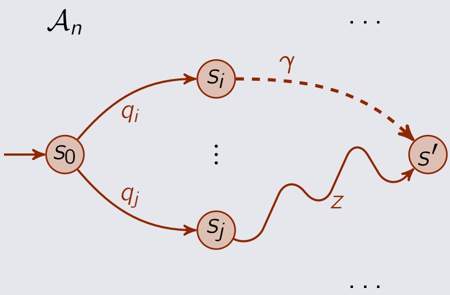
Only polynomially many transitions can be added

(⇒ reachability in PTIME)

Saturation procedure

Construct $\mathcal{A}_{n+1}$ from $\mathcal{A}_n$ by adding transitions, as follows:

1. select a transition rule $(q_i\gamma, a, q_j z)$ in the pushdown system $\mathcal{P}$
2. select a state s' in $\mathcal{A}_n$ reachable from s_0 via a $q_j z$-labelled path
3. add transition (s_i, γ, s')



Termination: straightforward

Soundness: by induction on n

$$s_0 \xrightarrow[\mathcal{A}_{n+1}]{q'z'} s'$$

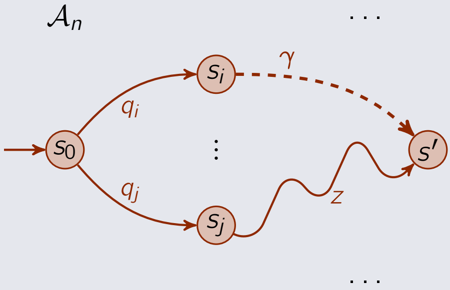
$$\Downarrow$$

$$s_0 \xrightarrow[\mathcal{A}_0]{qz} s' \quad \wedge \quad q'z' \xrightarrow[\mathcal{P}]{\star} qz$$

Saturation procedure

Construct $\mathcal{A}_{n+1}$ from $\mathcal{A}_n$ by adding transitions, as follows:

1. select a transition rule $(q_i\gamma, a, q_j z)$ in the pushdown system $\mathcal{P}$
2. select a state s' in $\mathcal{A}_n$ reachable from s_0 via a $q_j z$-labelled path
3. add transition (s_i, γ, s')



Termination: straightforward

Soundness: by induction on n

Completeness:

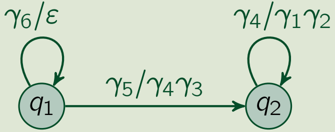
$\forall$ config. $q_i\gamma w \in B_{n+1} \setminus B_n$

$\exists$ trans. $q_i\gamma w \xrightarrow[\mathcal{P}]{a} q_j z w$ with $q_j z w \in B_n$

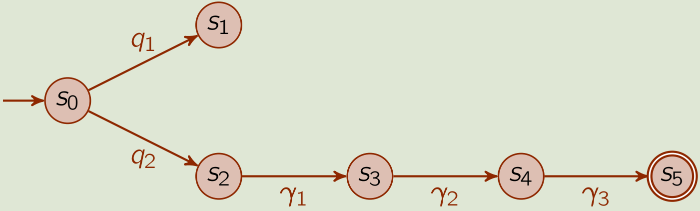
☞ Select rule $(q_i\gamma, a, q_j z)$ in $\mathcal{P}$ and path $s_0 \xrightarrow[\mathcal{A}_n]{q_j z} s'$ in $\mathcal{A}_n$ to prove that $q_i\gamma w \in \mathscr{L}(\mathcal{A}_{n+1})$

Example

Consider the target set $B_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$ over the pushdown system

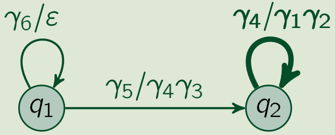


$C_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$

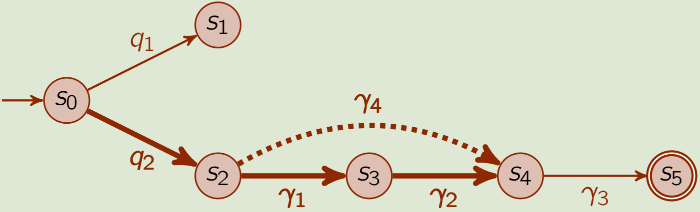


Example

Consider the target set $B_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$ over the pushdown system

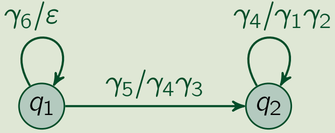


$C_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$



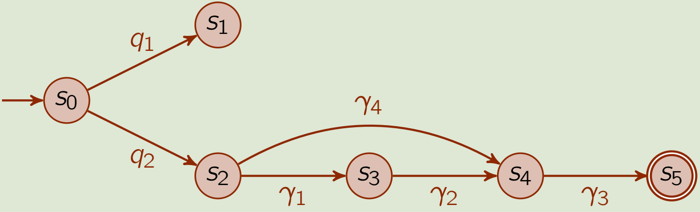
Example

Consider the target set $B_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$ over the pushdown system



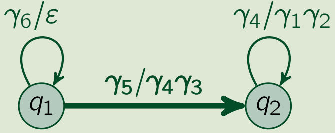
$C_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$

$C_1 = \{q_2\gamma_1\gamma_2\gamma_3,\ q_2\gamma_4\gamma_3\}$



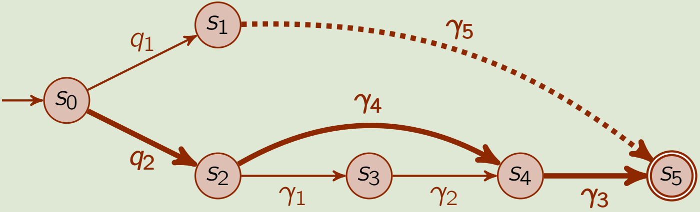
Example

Consider the target set $B_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$ over the pushdown system



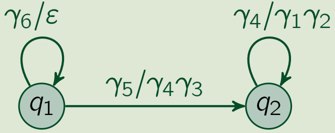
$C_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$

$C_1 = \{q_2\gamma_1\gamma_2\gamma_3,\ q_2\gamma_4\gamma_3\}$



Example

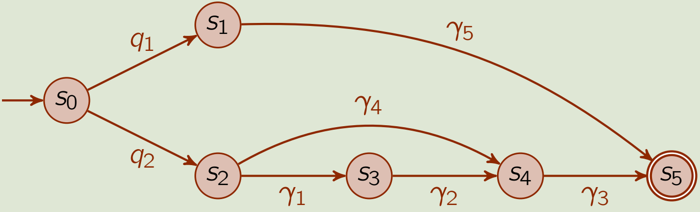
Consider the target set $B_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$ over the pushdown system



$C_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$

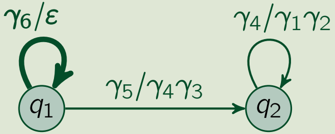
$C_1 = \{q_2\gamma_1\gamma_2\gamma_3,\ q_2\gamma_4\gamma_3\}$

$C_2 = \{q_2\gamma_1\gamma_2\gamma_3,\ q_2\gamma_4\gamma_3,\ q_1\gamma_5\}$



Example

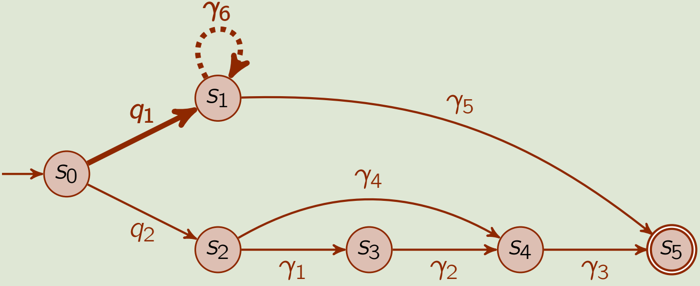
Consider the target set $B_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$ over the pushdown system



$$C_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$$

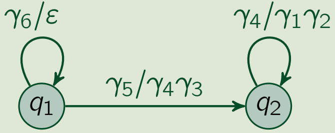
$$C_1 = \{q_2\gamma_1\gamma_2\gamma_3,\ q_2\gamma_4\gamma_3\}$$

$$C_2 = \{q_2\gamma_1\gamma_2\gamma_3,\ q_2\gamma_4\gamma_3,\ q_1\gamma_5\}$$



Example

Consider the target set $B_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$ over the pushdown system

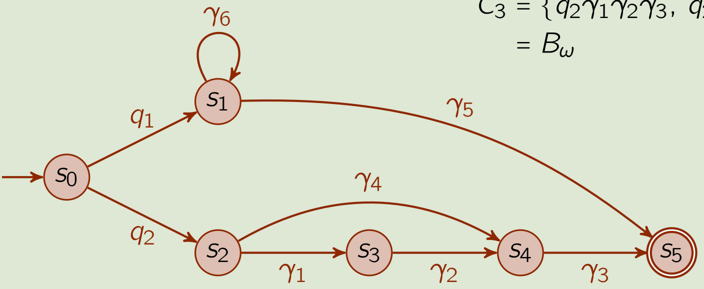


$$C_0 = \{q_2\gamma_1\gamma_2\gamma_3\}$$

$$C_1 = \{q_2\gamma_1\gamma_2\gamma_3,\ q_2\gamma_4\gamma_3\}$$

$$C_2 = \{q_2\gamma_1\gamma_2\gamma_3,\ q_2\gamma_4\gamma_3,\ q_1\gamma_5\}$$

$$C_3 = \{q_2\gamma_1\gamma_2\gamma_3,\ q_2\gamma_4\gamma_3,\ q_1\gamma_6^\star\gamma_5\} = B_\omega$$



Theorem (Bouajjani, Esparza & Maler '97)

Given a pushdown system $\mathcal{P}$ and a regular set B of configurations,
the set of configurations that can reach B
is regular and can be computed in polynomial time.

Theorem (Bouajjani, Esparza & Maler '97)

Given an **alternating** pushdown system $\mathcal{P}$ and a regular set B of conf., the **winning region** for the **B-reachability game** is regular and can be computed in polynomial time.

Theorem (Bouajjani, Esparza & Maler '97)

Given an **alternating** pushdown system $\mathcal{P}$ and a regular set B of conf., the **winning region** for the **B-reachability game** is regular and can be computed in polynomial time.

Similar generalizations can be proved for:

- **tree rewriting systems**
  (Löding '06, ...)
- reachability games on **higher-order pushdown systems**
  (Bouajjani & Meyer '04, Hague & Ong '07, ...)
- ...

Next we will focus on reachability for systems that use **variables over natural numbers** instead of a stack...

$(x, y) := (0, 0)$

while $(x, y) \neq (0, 1)$ do

if [*input is north west*] then

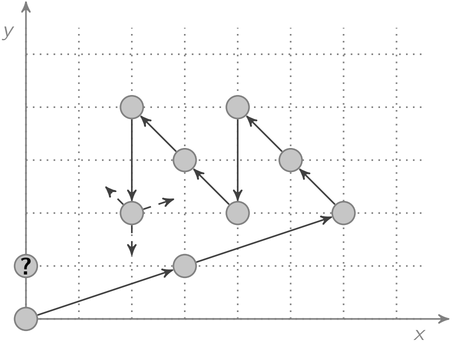
$(x, y) := (x, y) + (1, 3)$

else if [*input is north east*] then

$(x, y) := (x, y) + (-1, 1)$

else if [*input is south*] then

$(x, y) := (x, y) + (0, -2)$



Next we will focus on reachability for systems that use **variables over natural numbers** instead of a stack...

$(x, y) := (0, 0)$

while $(x, y) \neq (0, 1)$ do

if [*input is north west*] then

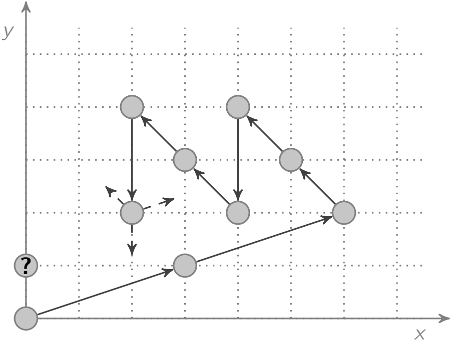
$(x, y) := (x, y) + (1, 3)$

else if [*input is north east*] then

$(x, y) := (x, y) + (-1, 1)$

else if [*input is south*] then

$(x, y) := (x, y) + (0, -2)$



Definition

A **vector addition system** (**VAS**) is a transition system $(\mathbb{N}^k, \Delta)$, where Δ is a finite subset of $\mathbb{Z}^k$ and

$$\bar{x} \longrightarrow \bar{y} \qquad \text{iff} \qquad \begin{cases} \bar{x}, \bar{y} \geq 0 \\ \bar{y} - \bar{x} \in \Delta \end{cases}$$

Definition

A **lossy VAS** is a transition system $(\mathbb{N}^k, \Delta)$,
where Δ is a finite subset of $Q \times \mathbb{Z}^k \times Q$ and

$$\bar{x} \longrightarrow \bar{y} \qquad \text{iff} \qquad \begin{cases} \bar{x}, \bar{y} \geq 0 \\ \bar{y}' - \bar{x} \in \Delta \quad \text{for some } \bar{y}' \geq \bar{y} \end{cases}$$

Definition

A **lossy VAS** is a transition system $(\mathbb{N}^k, \Delta)$,
where Δ is a finite subset of $Q \times \mathbb{Z}^k \times Q$ and

$$\bar{x} \longrightarrow \bar{y} \qquad \text{iff} \qquad \begin{cases} \bar{x}, \bar{y} \geq 0 \\ \bar{y}' - \bar{x} \in \Delta \quad \text{for some } \bar{y}' \geq \bar{y} \end{cases}$$

A **VAS with states** is a transition system $(Q \times \mathbb{N}^k, \Delta)$,
where Δ is a finite subset of $Q \times \mathbb{Z}^k \times Q$ and

$$(p, \bar{x}) \longrightarrow (q, \bar{y}) \qquad \text{iff} \qquad \begin{cases} \bar{x}, \bar{y} \geq 0 \\ (p,\ \bar{y} - \bar{x},\ q) \in \Delta \end{cases}$$

Definition

A **lossy VAS** is a transition system $(\mathbb{N}^k, \Delta)$,
where Δ is a finite subset of $Q \times \mathbb{Z}^k \times Q$ and

$$\bar{x} \longrightarrow \bar{y} \qquad \text{iff} \qquad \begin{cases} \bar{x}, \bar{y} \geq 0 \\ \bar{y}' - \bar{x} \in \Delta \quad \text{for some } \bar{y}' \geq \bar{y} \end{cases}$$

A **VAS with states** is a transition system $(Q \times \mathbb{N}^k, \Delta)$,
where Δ is a finite subset of $Q \times \mathbb{Z}^k \times Q$ and

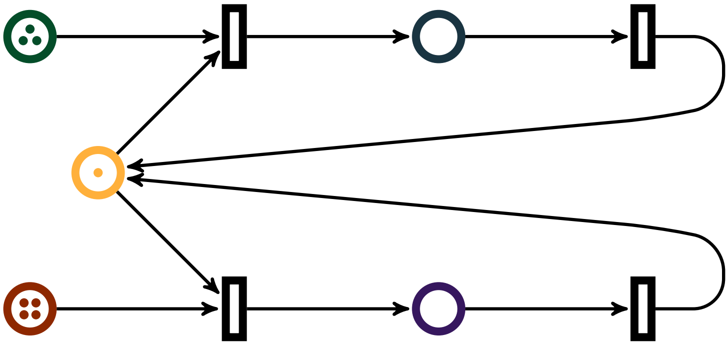
$$(p, \bar{x}) \longrightarrow (q, \bar{y}) \qquad \text{iff} \qquad \begin{cases} \bar{x}, \bar{y} \geq 0 \\ (p,\ \bar{y} - \bar{x},\ q) \in \Delta \end{cases}$$

☞ States do not add power, as they can be implemented by counters

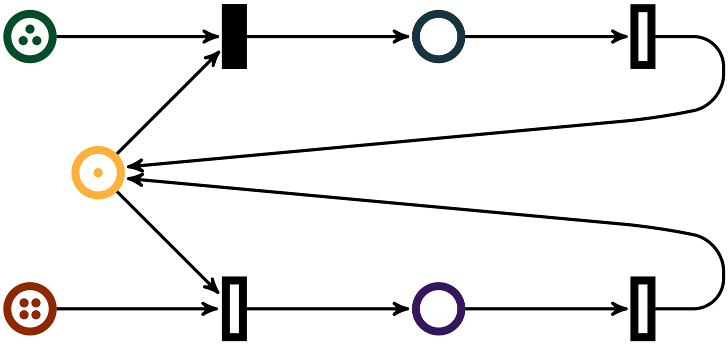
e.g. 2 states $=$ 2 additional counters that sum up to 1

$(p, \bar{x}) \longrightarrow (q, \bar{y})$ becomes $(0, 1, \bar{x}) \longrightarrow (1, 0, \bar{y})$

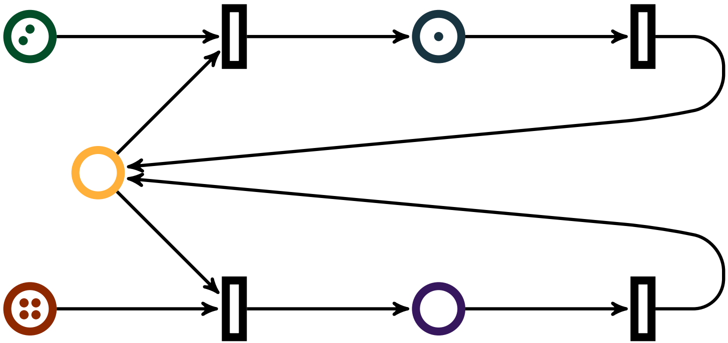
VAS are the same as **Petri nets**:



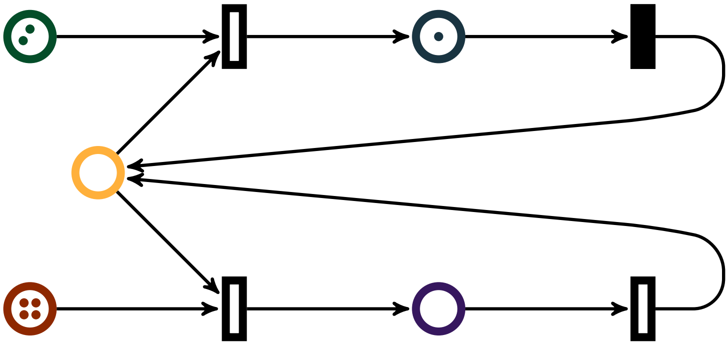
VAS are the same as **Petri nets**:



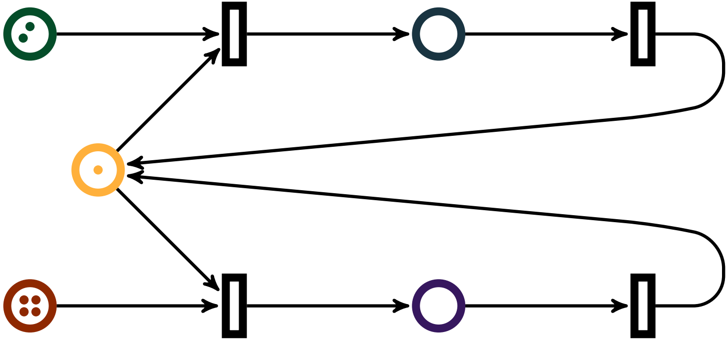
VAS are the same as **Petri nets**:



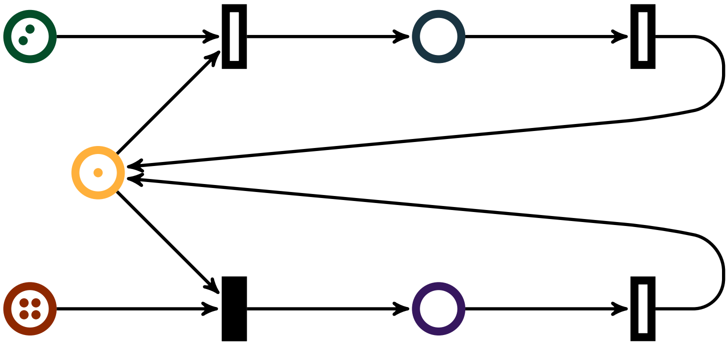
VAS are the same as **Petri nets**:



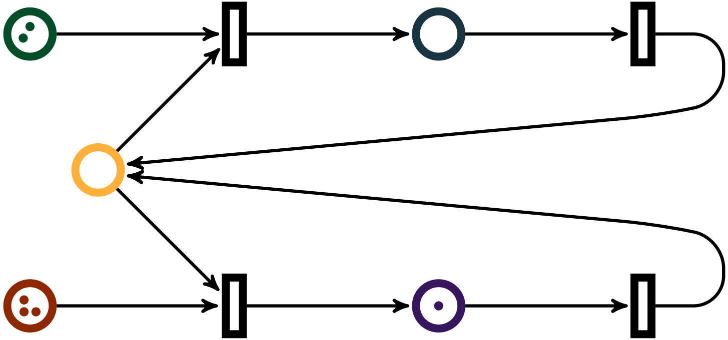
VAS are the same as **Petri nets**:



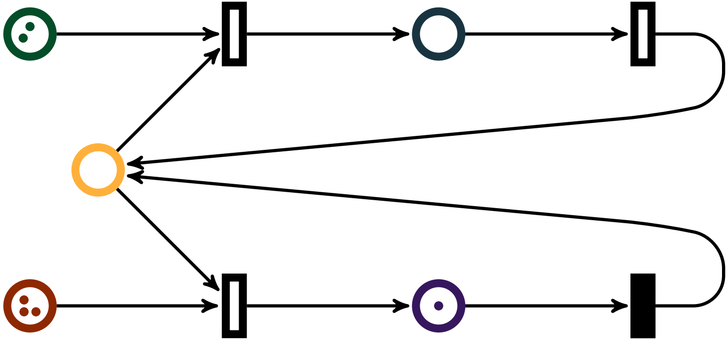
VAS are the same as **Petri nets**:



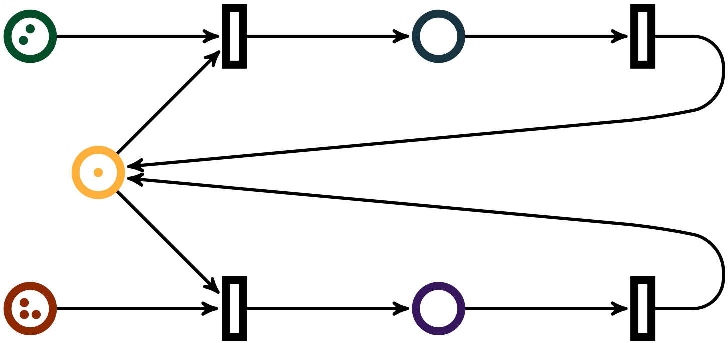
VAS are the same as **Petri nets**:



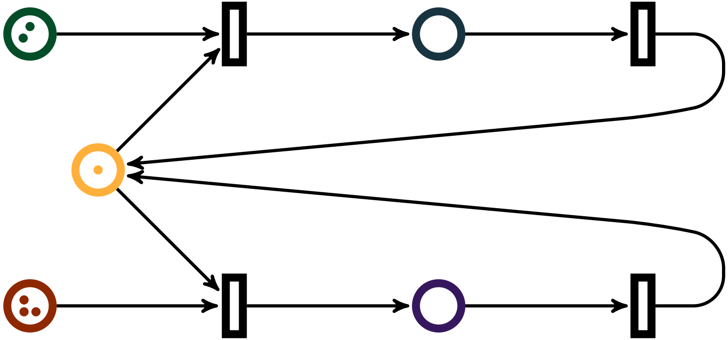
VAS are the same as **Petri nets**:



VAS are the same as **Petri nets**:



VAS are the same as **Petri nets**:

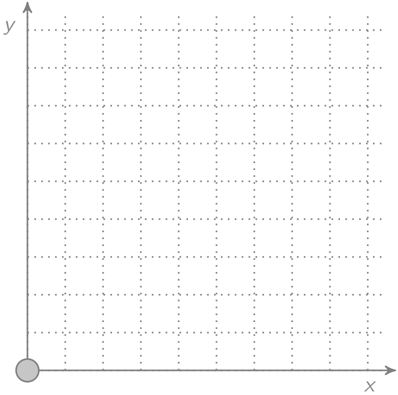


☞ configurations = tokens per location (e.g. $(2,1,3,0,0)$)

transitions = transfers of tokens (e.g. $(0,-1,-1,0,1)$)

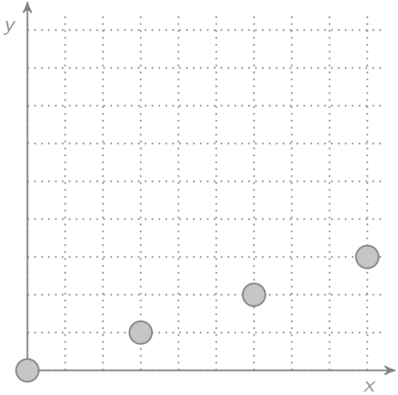
We may expect that reachable sets are **linear**...

$(0, 0)$



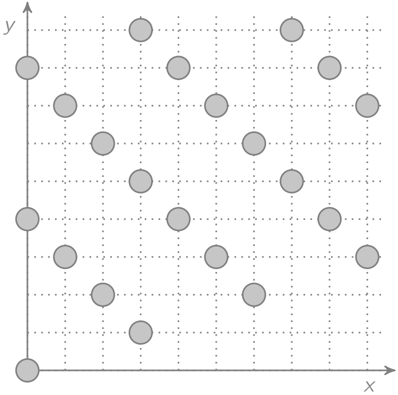
We may expect that reachable sets are **linear**...

$(0, 0) + (3, 1)\mathbb{N}$



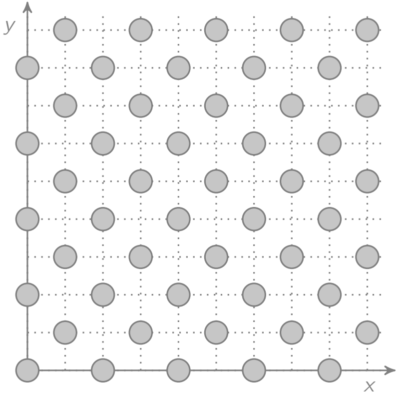
We may expect that reachable sets are **linear**...

$(0,0) + (3,1)\mathbb{N} + (-1,1)\mathbb{N}$



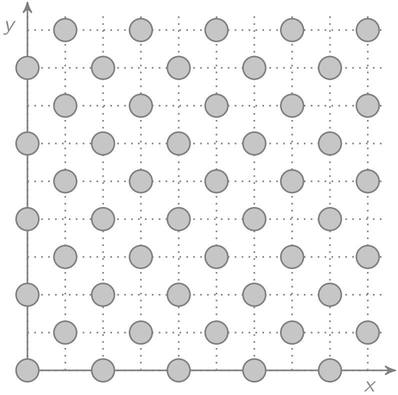
We may expect that reachable sets are **linear**...

$$(0,0) + (3,1)\mathbb{N} + (-1,1)\mathbb{N} + (0,-2)\mathbb{N}$$



We may expect that reachable sets are **linear**...

$$(0,0) + (3,1)\mathbb{N} + (-1,1)\mathbb{N} + (0,-2)\mathbb{N}$$

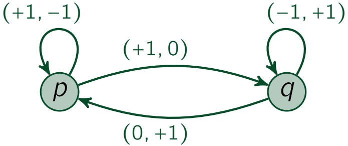


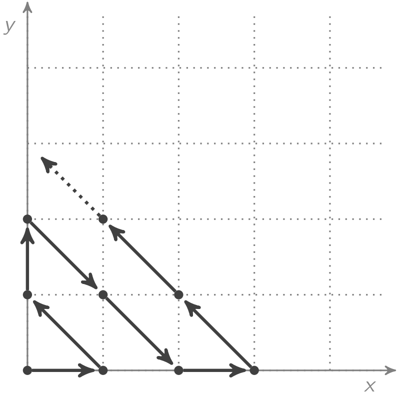
Theorem (Ginsburg '66)

Finite unions of linear sets are precisely the **Presburger sets** i.e. sets definable in $\mathsf{FO}[\mathbb{N},+]$

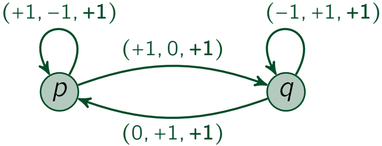
e.g. $\varphi(x,y) \;=\; \exists z.\ x + y = z + z$

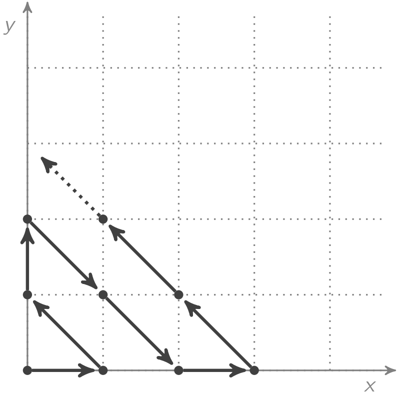
We may expect that reachable sets are **linear**... but they are not!



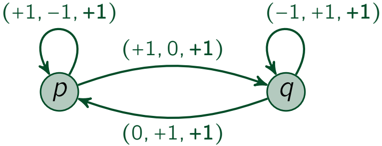


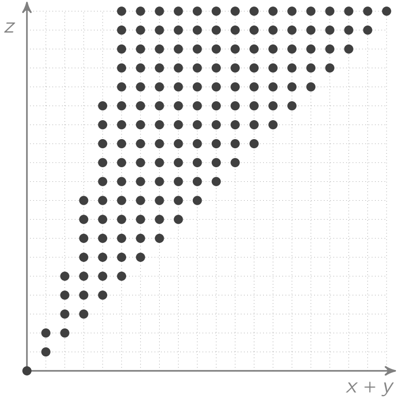
We may expect that reachable sets are **linear**... but they are not!





We may expect that reachable sets are **linear**... but they are not!





$$(x + y) \ \leq \ z \ \leq \ \mathcal{O}\big((x + y)^2\big)$$

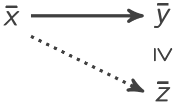
To overcome the problem of representing reachable sets,
we try to **over-approximate by downward closures**:

$$V^{\downarrow} \;=\; \left\{ \bar{z} \;\middle|\; \exists\, \bar{y} \in V.\ \bar{z} \leq \bar{y} \right\}$$

To overcome the problem of representing reachable sets,
we try to **over-approximate by downward closures**:

$$V^{\downarrow} = \left\{ \bar{z} \;\middle|\; \exists\, \bar{y} \in V.\ \bar{z} \leq \bar{y} \right\}$$

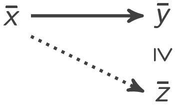
This is not an approximation for **lossy** VAS!



To overcome the problem of representing reachable sets,
we try to **over-approximate by downward closures**:

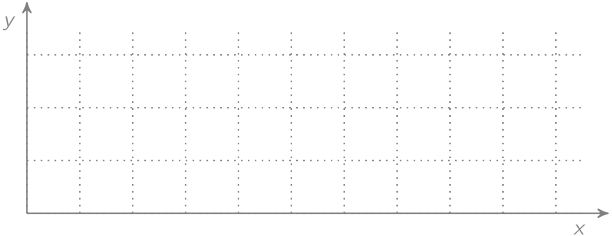
$$V^{\downarrow} = \left\{ \bar{z} \;\middle|\; \exists\, \bar{y} \in V.\ \bar{z} \leq \bar{y} \right\}$$

This is not an approximation for **lossy** VAS!



Dickson's Lemma 1913

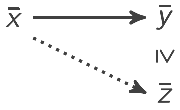
The pointwise order $\leq$ on $\mathbb{N}^k$ is a **well partial order**
(i.e. all **decreasing chains** and all **antichains** are finite)



To overcome the problem of representing reachable sets, we try to **over-approximate by downward closures**:

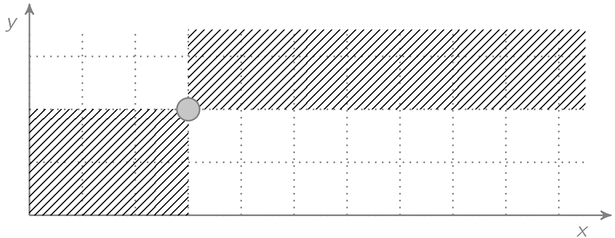
$$V^{\downarrow} = \left\{ \bar{z} \;\middle|\; \exists\, \bar{y} \in V.\ \bar{z} \le \bar{y} \right\}$$

☞ This is not an approximation for **lossy** VAS!



Dickson's Lemma 1913

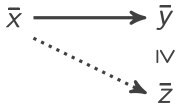
The pointwise order $\le$ on $\mathbb{N}^k$ is a **well partial order** (i.e. all **decreasing chains** and all **antichains** are finite)



To overcome the problem of representing reachable sets,
we try to **over-approximate by downward closures**:

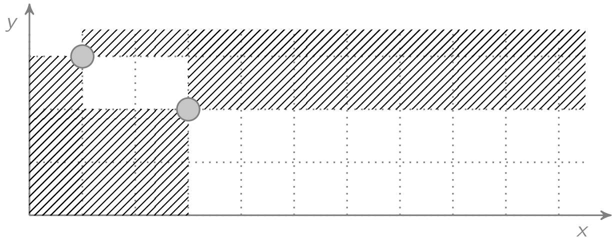
$$V^{\downarrow} = \{ \bar{z} \mid \exists \bar{y} \in V.\ \bar{z} \leq \bar{y} \}$$

This is not an approximation for **lossy** VAS!



Dickson's Lemma 1913

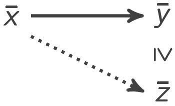
The pointwise order $\leq$ on $\mathbb{N}^k$ is a **well partial order**
(i.e. all **decreasing chains** and all **antichains** are finite)



To overcome the problem of representing reachable sets,
we try to **over-approximate by downward closures**:

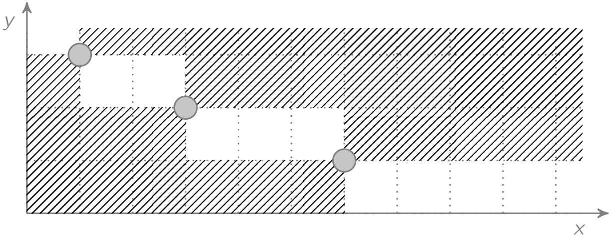
$$V^{\downarrow} = \{\bar{z} \mid \exists \bar{y} \in V.\ \bar{z} \leq \bar{y}\}$$

This is not an approximation for **lossy** VAS!



Dickson's Lemma 1913

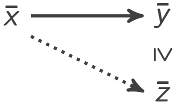
The pointwise order $\leq$ on $\mathbb{N}^k$ is a **well partial order**
(i.e. all **decreasing chains** and all **antichains** are finite)



To overcome the problem of representing reachable sets,
we try to **over-approximate by downward closures**:

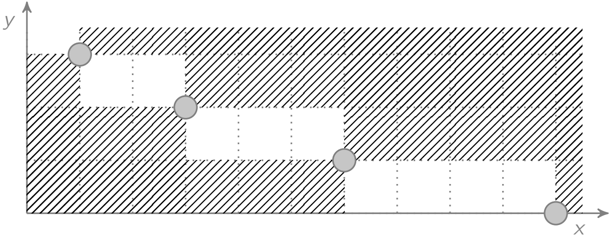
$$V^{\downarrow} = \left\{ \bar{z} \;\middle|\; \exists \bar{y} \in V.\ \bar{z} \leq \bar{y} \right\}$$

This is not an approximation for **lossy** VAS!



Dickson's Lemma 1913

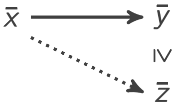
The pointwise order $\leq$ on $\mathbb{N}^k$ is a **well partial order**
(i.e. all **decreasing chains** and all **antichains** are finite)



To overcome the problem of representing reachable sets,
we try to **over-approximate by downward closures**:

$$V^{\downarrow} = \left\{ \bar{z} \;\middle|\; \exists\, \bar{y} \in V.\ \bar{z} \le \bar{y} \right\}$$

This is not an approximation for **lossy** VAS!



Dickson's Lemma 1913

The pointwise order $\le$ on $\mathbb{N}^k$ is a **well partial order**
(i.e. all **decreasing chains** and all **antichains** are finite)

Lemma

For all subsets V of $(\mathbb{N} \cup \{\infty\})^k$, there is an **antichain** W such that

$$V^{\downarrow} = W^{\downarrow}$$

$\Rightarrow$ we can finitely represent downward-closed sets by antichains

Karp & Miller Algorithm '69

Saturation of downward-closed sets via transition function Δ

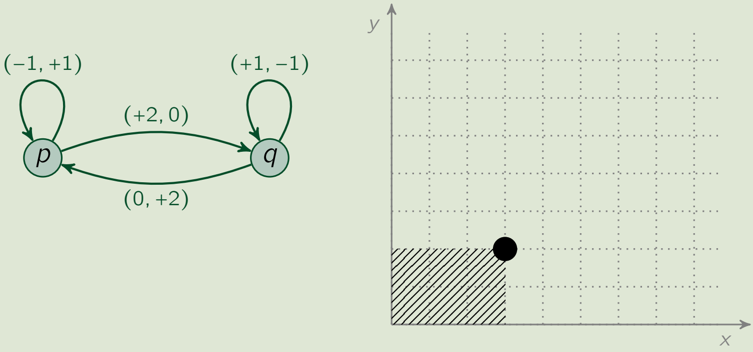
- acceleration on emerging dominating sets...

Karp & Miller Algorithm '69

Saturation of downward-closed sets via transition function Δ

- acceleration on emerging dominating sets...

Example

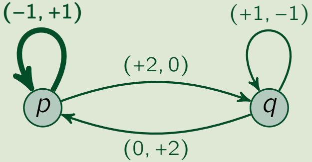


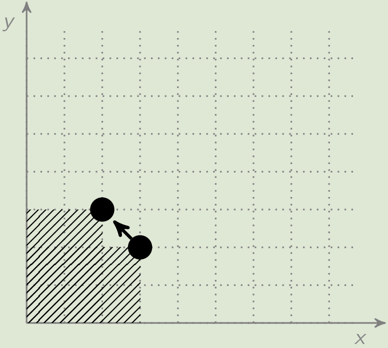
Karp & Miller Algorithm '69

Saturation of downward-closed sets via transition function Δ

- acceleration on emerging dominating sets...

Example



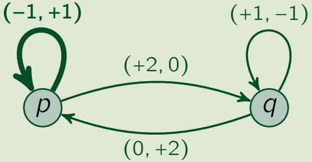


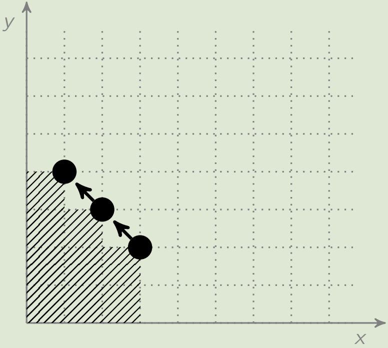
Karp & Miller Algorithm '69

Saturation of downward-closed sets via transition function Δ

- acceleration on emerging dominating sets...

Example



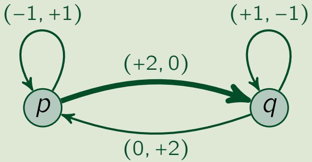


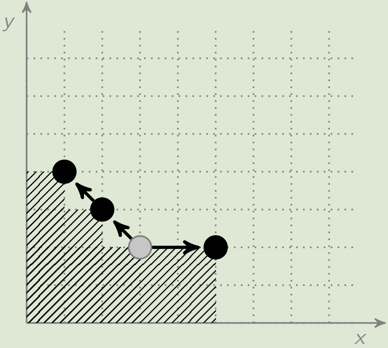
Karp & Miller Algorithm '69

Saturation of downward-closed sets via transition function Δ

- acceleration on emerging dominating sets...

Example



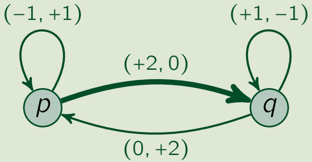


Karp & Miller Algorithm '69

Saturation of downward-closed sets via transition function Δ

- acceleration on emerging dominating sets...

Example

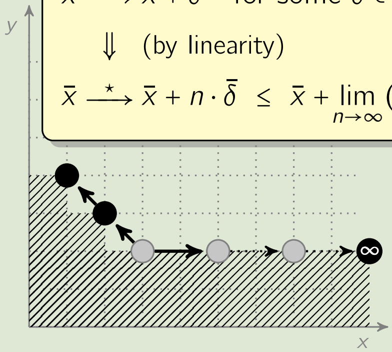


Correctness of acceleration

$$\bar{x} \xrightarrow{\star} \bar{x} + \bar{\delta} \quad \text{for some } \bar{\delta} \in \mathbb{N}^k$$

$\Downarrow$ (by linearity)

$$\bar{x} \xrightarrow{\star} \bar{x} + n \cdot \bar{\delta} \;\leq\; \bar{x} + \lim_{n\to\infty} (n \cdot \bar{\delta})$$

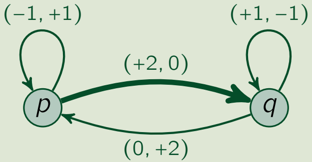


Karp & Miller Algorithm '69

Saturation of downward-closed sets via transition function Δ

- acceleration on emerging dominating sets...

Example

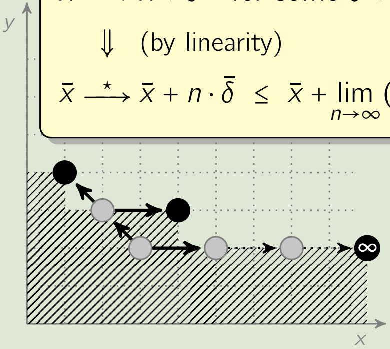


Correctness of acceleration

$\bar{x} \xrightarrow{\star} \bar{x} + \bar{\delta}$ for some $\bar{\delta} \in \mathbb{N}^k$

$\Downarrow$ (by linearity)

$\bar{x} \xrightarrow{\star} \bar{x} + n \cdot \bar{\delta} \leq \bar{x} + \lim_{n \to \infty} (n \cdot \bar{\delta})$

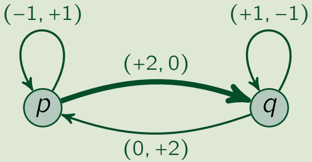


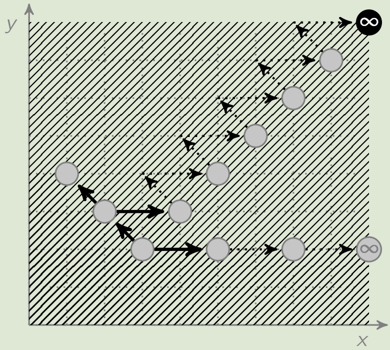
Karp & Miller Algorithm ’69

Saturation of downward-closed sets via transition function Δ

- acceleration on emerging dominating sets…

Example





Theorem (Rackoff '78)

Coverability on VAS (i.e. given $\bar{x}, \bar{y}$, tell if $\exists\, \bar{z} \geq \bar{y}.\ \bar{x} \xrightarrow{\star} \bar{z}$) is EXPSPACE-complete.

Theorem (Rackoff '78)

Coverability on VAS (i.e. given $\bar{x}, \bar{y}$, tell if $\exists\, \bar{z} \geq \bar{y}.\ \bar{x} \xrightarrow{\star} \bar{z}$) is EXPSPACE-complete.

Corollary 1

Reachability on lossy VAS is EXPSPACE-complete.

Corollary 2

Control-state reachability on VAS is EXPSPACE-complete.

There are other results similar in spirit...

Theorem (Adbulla, Cerans & Jonsson '96, . . .)

Coverability is decidable (**non-primitive recursive**) on VAS with

- **resets** (e.g. $x := 0$)
- **transfers** (e.g. $x := y + z$)
- **positive guards** (e.g. *if* $[x > 0]$ *then* . . .)

Reachability is decidable on analogous extensions of **lossy VAS**.

There are other results similar in spirit...

Theorem (Adbulla, Cerans & Jonsson '96, ...)

Coverability is decidable (**non-primitive recursive**) on VAS with

- **resets** (e.g. $x := 0$)
- **transfers** (e.g. $x := y + z$)
- **positive guards** (e.g. *if* $[x > 0]$ *then* ...)

Reachability is decidable on analogous extensions of **lossy VAS**.

Unfortunately, acceleration for the above systems does not work,

e.g. $(1, 0) \xrightarrow[y := 1]{\text{reset } x} (0, 1) \xrightarrow[y := y-1]{x := x+2} (2, 0)$, but $(1, 0) \not\xrightarrow{\star} (3, 0)$

There are other results similar in spirit...

Theorem (Adbulla, Cerans & Jonsson '96, ...)

Coverability is decidable (**non-primitive recursive**) on VAS with

- **resets** (e.g. $x := 0$)
- **transfers** (e.g. $x := y + z$)
- **positive guards** (e.g. *if* $[x > 0]$ *then* ...)

Reachability is decidable on analogous extensions of **lossy VAS**.

Unfortunately, acceleration for the above systems does not work,

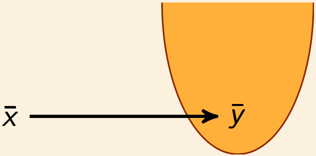
e.g. $(1,0) \xrightarrow[y:=1]{reset\ x} (0,1) \xrightarrow[y:=y-1]{x:=x+2} (2,0)$, but $(1,0) \not\xrightarrow{\star} (3,0)$

However, we can still exploit Dickson's Lemma with

1. **upward-closed sets**
   they cover more vectors than downward-closed sets!
2. **backward reachability**
   i.e. compute $B_{n+1} = \{ \bar{x} \mid \exists\, \bar{y} \in B_n.\ \bar{x} \longrightarrow \bar{y} \}$

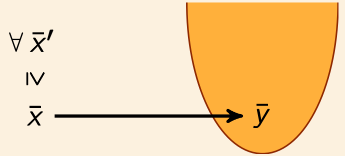
Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.



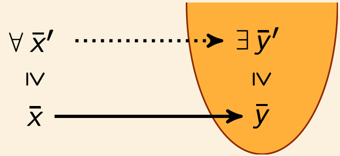
Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.



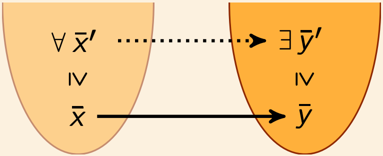
Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.



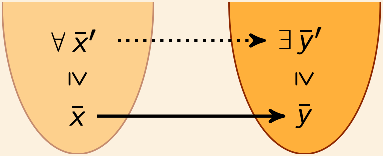
Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.

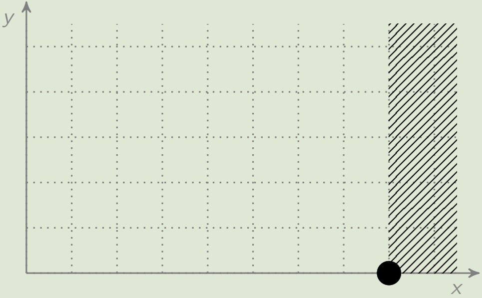


Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.

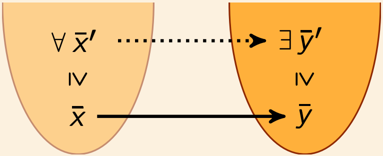


Example of backward coverability analysis

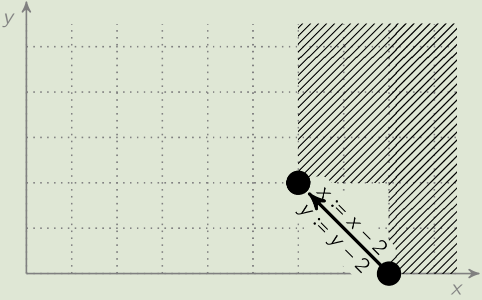


Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.

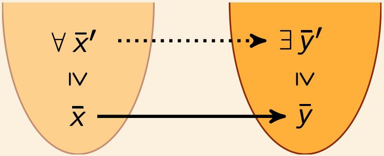


Example of backward coverability analysis

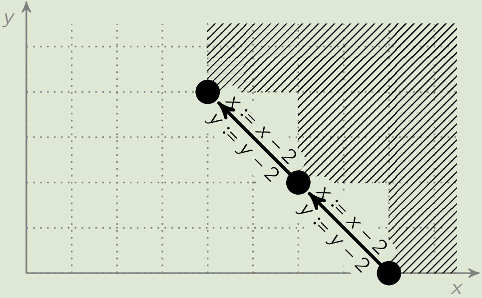


Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.

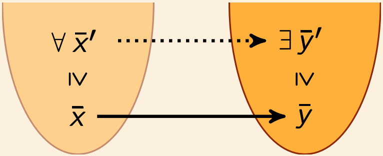


Example of backward coverability analysis

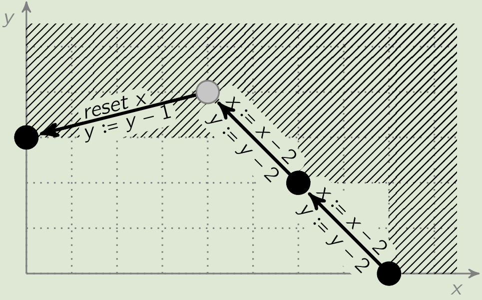


Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.

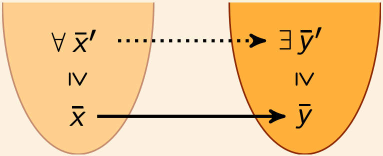


Example of backward coverability analysis

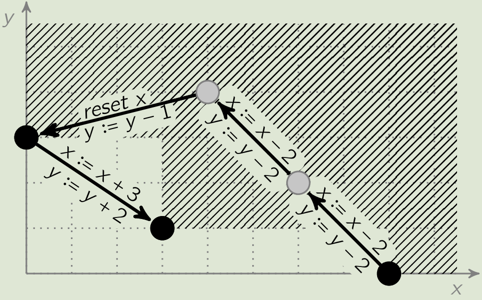


Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.

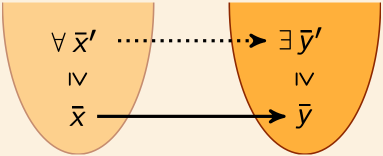


Example of backward coverability analysis

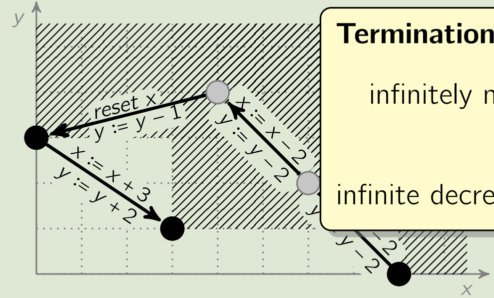


Lemma

VAS transitions with resets, transfers, and positive guards are **backward-compatible with upward-closures**, i.e.



Example of backward coverability analysis



Termination by Dickson's Lemma:

infinitely many emerging points

$\Downarrow$

infinite decreasing chain or antichain

These ideas for coverability analysis can be extended to:

- **Lossy Channel Systems**
  (instead of Dickson's Lemma,
  use Higman's Lemma for the sub-sequence partial order)
- **Timed Petri nets**
  (token have time-stamps, transitions have time constraints)
- **Alternating Finite Memory Automata**
  (finite control states + one register to store
  and compare symbols from an infinite alphabet)
- ...

Now, back to the original reachability problem on VAS...

Separation Theorem (Leroux '92, '09, . . . , '12)

If $\bar{x} \not\xrightarrow{*}_{\Delta} \bar{y}$, then there is a partition (X, Y) of $\mathbb{N}^k$ such that

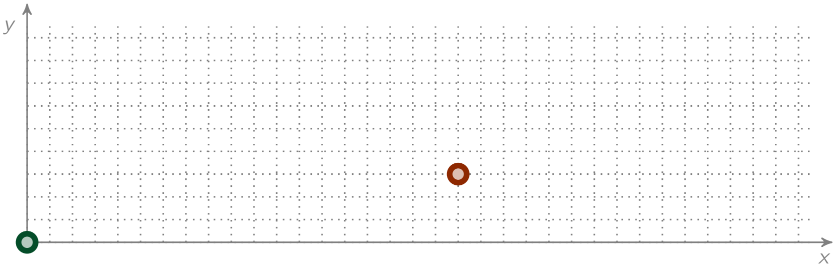
1. X and Y are **finite unions of linear sets** (or, equally, sets definable in **Presburder logic FO[$\mathbb{N}, +$]**)
2. $\bar{x} \in X$ and $\bar{y} \in Y$
3. X is a **forward invariant**, i.e. $(X + \Delta) \cap \mathbb{N}^k \subseteq X$
4. Y is a **backward invariant**, i.e. $(Y - \Delta) \cap \mathbb{N}^k \subseteq Y$

Now, back to the original reachability problem on VAS...

Separation Theorem (Leroux '92, '09, . . . , '12)

If $\bar{x} \not\xrightarrow{*}_{\Delta} \bar{y}$, then there is a partition (X, Y) of $\mathbb{N}^k$ such that

1. X and Y are **finite unions of linear sets** (or, equally, sets definable in **Presburder logic FO[$\mathbb{N}$, +]**)
2. $\bar{x} \in X$ and $\bar{y} \in Y$
3. X is a **forward invariant**, i.e. $(X + \Delta) \cap \mathbb{N}^k \subseteq X$
4. Y is a **backward invariant**, i.e. $(Y - \Delta) \cap \mathbb{N}^k \subseteq Y$

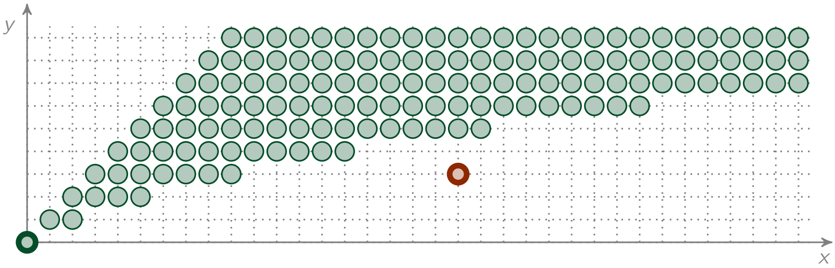


Now, back to the original reachability problem on VAS...

Separation Theorem (Leroux '92, '09, . . . , '12)

If $\bar{x} \overset{*}{\not\to}_{\Delta} \bar{y}$, then there is a partition (X, Y) of $\mathbb{N}^k$ such that

1. X and Y are **finite unions of linear sets** (or, equally, sets definable in **Presburder logic FO[$\mathbb{N}, +$]**)
2. $\bar{x} \in X$ and $\bar{y} \in Y$
3. X is a **forward invariant**, i.e. $(X + \Delta) \cap \mathbb{N}^k \subseteq X$
4. Y is a **backward invariant**, i.e. $(Y - \Delta) \cap \mathbb{N}^k \subseteq Y$

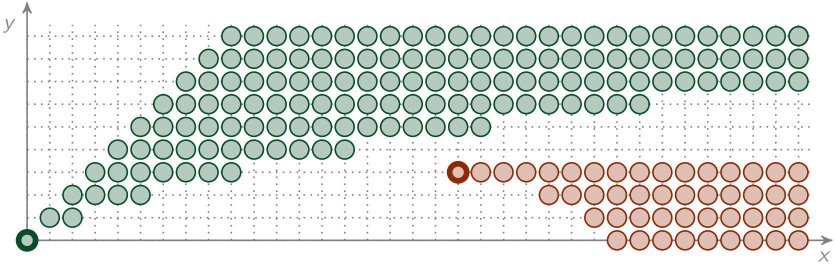


Now, back to the original reachability problem on VAS...

Separation Theorem (Leroux '92, '09, . . . , '12)

If $\bar{x} \not\xrightarrow{*}_{\Delta} \bar{y}$, then there is a partition (X, Y) of $\mathbb{N}^k$ such that

1. X and Y are **finite unions of linear sets** (or, equally, sets definable in **Presburder logic FO[$\mathbb{N}, +$]**)
2. $\bar{x} \in X$ and $\bar{y} \in Y$
3. X is a **forward invariant**, i.e. $(X + \Delta) \cap \mathbb{N}^k \subseteq X$
4. Y is a **backward invariant**, i.e. $(Y - \Delta) \cap \mathbb{N}^k \subseteq Y$

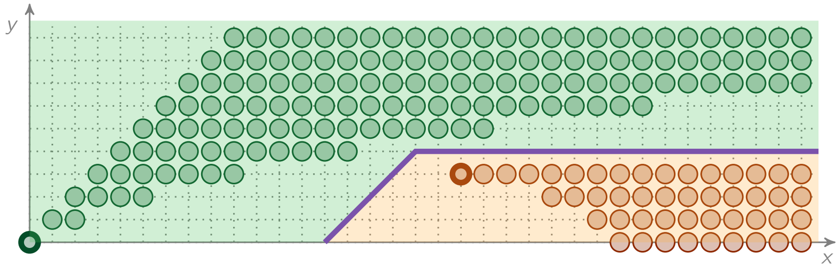


Now, back to the original reachability problem on VAS...

Separation Theorem (Leroux '92, '09, . . . , '12)

If $\bar{x} \not\xrightarrow{*}_{\Delta} \bar{y}$, then there is a partition (X, Y) of $\mathbb{N}^k$ such that

1. X and Y are **finite unions of linear sets** (or, equally, sets definable in **Presburder logic FO[$\mathbb{N}$, +]**)
2. $\bar{x} \in X$ and $\bar{y} \in Y$
3. X is a **forward invariant**, i.e. $(X + \Delta) \cap \mathbb{N}^k \subseteq X$
4. Y is a **backward invariant**, i.e. $(Y - \Delta) \cap \mathbb{N}^k \subseteq Y$



Corollary (Lipton '76, Mayr '81, Kosaraju '82, Reutenauer '90, ...)

The reachability problem for VAS is decidable
with complexity between EXPSPACE and non-primitive recursive.

Corollary (Lipton '76, Mayr '81, Kosaraju '82, Reutenauer '90, ...)

The reachability problem for VAS is decidable
with complexity between EXPSPACE and non-primitive recursive.

Enumerate in parallel:

1. the possible finite sequences π of transitions
(answer positively if $\bar{x} \xrightarrow{\pi} \bar{y}$)

2. the possible Presburger formulas defining partitions (X, Y) of $\mathbb{N}^k$
(answer negatively if (X, Y) is an invariant separating $\bar{x}$ and $\bar{y}$)

"That's all Folks!"