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

![Diagram showing acceleration and pumping](image)
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**

![Diagram showing acceleration and 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**

Both approaches require symbolic representations of infinite sets.
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 $B_0 \subseteq Q \cdot \Gamma^*$ of target configurations, define:

$$B_{n+1} = B_n \cup \{ qz \mid \exists q'z' \in B_n. \exists a \in \Sigma. qz \xrightarrow{a} q'z' \}$$

$$B_\omega = \bigcup_{n \in \mathbb{N}} B_n$$
Backward reachability for pushdown systems

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

\[
B_{n+1} = B_n \cup \{ qz \mid \exists q' z' \in B_n. \exists a \in \Sigma. qz \xrightarrow{a} q' z' \}
\]

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

\( B_\omega \) 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 $B_0 \subseteq Q \cdot \Gamma^*$ of target configurations, define:

$$B_{n+1} = B_n \cup \{ qz \mid \exists q'z' \in B_n. \exists a \in \Sigma. qz \xrightarrow{a} q'z' \}$$

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

$B_\omega$ contains the configurations from which one can reach $B_0$. $B_\omega$ is usually infinite, but is it perhaps regular?
Backward reachability for pushdown systems

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

$$B_n + 1 = B_n \cup \{ qz \mid \exists q' z' \in B_n. \forall a \in \Sigma. qz \xrightarrow{a} q' z' \}$$

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

$B_\omega$ contains the configurations from which one can reach $B_0$. $B_\omega$ 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 \( P = (Q, \Sigma, \Gamma, \Delta) \) and a set \( B_0 \subseteq Q \cdot \Gamma^* \) of target configurations, define:

\[
B_{n+1} = B_n \cup \{ qz | \exists q'z' \in B_n. \exists a \in \Sigma. qz \xrightarrow{a} q'z' \}
\]

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

\( B_\omega \) contains the configurations from which one can reach \( B_0 \). 

\( B_\omega \) is usually infinite, but is it perhaps regular?

Example

Consider the pushdown system

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

\( B_\omega = q\varepsilon^* \) 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:

(completeness) $\forall n \in \mathbb{N}. \quad B_n \subseteq C_n$

(soundness) $\forall n \in \mathbb{N}. \quad C_n \subseteq B_\omega$

(termination) $\exists n \in \mathbb{N}. \quad C_n = C_{n+1}$

$\bigcup_{n \in \mathbb{N}} C_n$ coincides with $B_\omega$
“Pump” the changes from $B_n$ to $B_{n+1}$ to obtain a new sequence $C_0, C_1, \ldots$ that converges more quickly:

(completeness) $\forall n \in \mathbb{N}. \quad B_n \subseteq C_n$

(soundness) $\forall n \in \mathbb{N}. \quad C_n \subseteq B_\omega$

(termination) $\exists n \in \mathbb{N}. \quad C_n = C_{n+1}$

The limit $\bigcup_{n \in \mathbb{N}} C_n$ coincides with $B_\omega$

The sets $C_0, C_1, \ldots$ will be defined by automata $A_0, A_1, \ldots$ 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 $A_0$ 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

Diagram:

- $A_0$
- $s_0$ (initial state)
- $s_1$
- $s_m$
- $\ldots$
Initial conditions

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

- The automaton $A_0$ 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 $A_0$ 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 $P$ has $m$ states $q_1, \ldots, q_m$
- The automaton $A_0$ 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 $A_0$ reaches the initial state $s_0$
- The unique $q_i$-labelled transition in $A_0$ is $(s_0, q_i, s_i)$
- The other transitions in $A_0$ are labelled by stack symbols
Saturation procedure

Construct $A_{n+1}$ from $A_n$ by adding transitions, as follows:

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

Construct $A_{n+1}$ from $A_n$ 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 $A_n$ reachable from $s_0$ via a $q_j z$-labelled path

Termination: straightforward

Only polynomially many transitions can be added ($\Rightarrow$ reachability in PTIME)

Soundness: by induction on $n$

Completeness: $\forall$ config. $q_i \gamma w \in B_{n+1}$ $\exists$ trans. $q_i \gamma w \rightarrow Pqjzw$ with $q_j zw \in B_n$
Saturation procedure

Construct $A_{n+1}$ from $A_n$ by adding transitions, as follows:

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

### Diagram

```
\begin{array}{cccccccc}
A_n & \rightarrow & \cdots & \rightarrow & A_{n+1} \\
\vdots & \rightarrow & \gamma & \rightarrow & \vdots \\
\end{array}
```

```
\begin{array}{cccccccc}
\text{\textit{s}_0} & \rightarrow & \text{\textit{s}_i} & \rightarrow & \text{\textit{s}_j} & \rightarrow & \text{\textit{s}_i} & \rightarrow & \text{\textit{s}'_j} & \rightarrow & \text{\textit{s}'_i} \\
\text{\textit{q}_i} & \rightarrow & \gamma & \rightarrow & \text{\textit{q}_j} & \rightarrow & \\
\end{array}
```
Saturation procedure

Construct $A_{n+1}$ from $A_n$ by adding transitions, as follows:

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

**Termination**: straightforward

Only polynomially many transitions can be added

$(\Rightarrow$ reachability in PTIME)$
Saturation procedure

Construct $A_{n+1}$ from $A_n$ by adding transitions, as follows:

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

**Termination:** straightforward

**Soundness:** by induction on $n$

Select rule $(q_i; \gamma, a, q_j z)$ in $P$ and path $s_0 s_0 s_0 q_j z q_j z q_j z$ to prove that $q_i \gamma w q_i \gamma w q_i \gamma w \in L(A_{n+1})$
Saturation procedure

Construct $A_{n+1}$ from $A_n$ by adding transitions, as follows:

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

**Termination**: straightforward

**Soundness**: by induction on $n$

**Completeness**: 

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

$\exists$ trans. $q_i; \gamma; w \xrightarrow{a}_P q_j; z; w$ 
with $q_j; z; w \in B_n$

Select rule $(q_i; \gamma, a, q_j; z)$ in $P$ 
and path $s_0 \xrightarrow{q_j; z}_A s'$ in $A_n$

to prove that $q_i; \gamma; w \in L(A_{n+1})$
Example

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

\[
\begin{align*}
B_0 = \{q_2\gamma_1\gamma_2\gamma_3\} \quad &\quad C_0 = \{q_2\gamma_1\gamma_2\gamma_3\} \\
\end{align*}
\]
Example

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

\[
\begin{align*}
\gamma_6/\varepsilon & \quad \gamma_5/\gamma_4 \gamma_3 & \quad \gamma_4/\gamma_1 \gamma_2 \\
q_1 & \quad q_2 & \quad C_0 = \{ q_2 \gamma_1 \gamma_2 \gamma_3 \}
\end{align*}
\]
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

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

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

\[
\begin{align*}
\gamma_6/\varepsilon & \quad \gamma_5/\gamma_4\gamma_3 & \quad \gamma_4/\gamma_1\gamma_2 \\
q_1 & \quad q_2 \\
C_0 &= \{ q_2\gamma_1\gamma_2\gamma_3 \} \\
C_1 &= \{ q_2\gamma_1\gamma_2\gamma_3, q_2\gamma_4\gamma_3 \} \\
C_2 &= \{ q_2\gamma_1\gamma_2\gamma_3, q_2\gamma_4\gamma_3, q_1\gamma_5 \}
\end{align*}
\]
Example

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

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

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

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

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

$= B_\omega$
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 \( P \) 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 $P$ 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

\[\text{if [input is north west] then}\]
\[\quad (x, y) := (x, y) + (1, 3)\]

\[\text{else if [input is north east] then}\]
\[\quad (x, y) := (x, y) + (-1, 1)\]

\[\text{else if [input is south] then}\]
\[\quad (x, y) := (x, y) + (0, -2)\]
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 \(\Delta\) is a finite subset of \(\mathbb{Z}^k\) and

\[
\bar{x} \rightarrow \bar{y} \quad \text{iff} \quad \begin{cases} 
\bar{x}, \bar{y} \geq 0 \\
\bar{y} - \bar{x} \in \Delta
\end{cases}
\]
A **lossy VAS** is a transition system \((\mathbb{N}^k, \Delta)\), where \(\Delta\) is a finite subset of \(Q \times \mathbb{Z}^k \times Q\) and

\[
\bar{x} \rightarrow \bar{y} \quad \text{iff} \quad \begin{cases} \bar{x}, \bar{y} \geq 0 \\ \bar{y}' - \bar{x} \in \Delta \end{cases} \quad \text{for some } \bar{y}' \geq \bar{y}
\]
A **lossy VAS** is a transition system \((\mathbb{N}^k, \Delta)\), where \(\Delta\) is a finite subset of \(Q \times \mathbb{Z}^k \times Q\) and

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

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

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

States do not add power, as they can be implemented by counters e.g. 2 states = 2 additional counters that sum up to 1.
A **lossy VAS** is a transition system \((\mathbb{N}^k, \Delta)\), where \(\Delta\) is a finite subset of \(Q \times \mathbb{Z}^k \times Q\) and

\[
\bar{x} \rightarrow \bar{y} \quad \text{iff} \quad \begin{cases}
\bar{x}, \bar{y} \geq 0 \\
\bar{y}' - \bar{x} \in \Delta \quad \text{for some } \bar{y}' \geq \bar{y}
\end{cases}
\]

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

\[
(p, \bar{x}) \rightarrow (q, \bar{y}) \quad \text{iff} \quad \begin{cases}
\bar{x}, \bar{y} \geq 0 \\
(p, \bar{y} - \bar{x}, q) \in \Delta
\end{cases}
\]

States do not add power, as they can be implemented by counters e.g. 2 states = 2 additional counters that sum up to 1

\[(p, \bar{x}) \rightarrow (q, \bar{y}) \quad \text{becomes} \quad (0, 1, \bar{x}) \rightarrow (1, 0, \bar{y})\]
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))
VAS are the same as **Petri nets**:

- **Configurations**: tokens per location (e.g. \( (2, 1, 3, 0, 0) \))
- **Transitions**: transfers of tokens (e.g. \( (0, -1, -1, 0, 1) \))
VAS are the same as Petri nets:

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

transitions = transfers of tokens (e.g. \((0, -1, -1, 0, 1)\))
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))
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))
VAS are the same as **Petri nets**:

- **Configurations** = tokens per location (e.g. $\langle 2, 1, 3, 0, 0 \rangle$)
- **Transitions** = transfers of tokens (e.g. $\langle 0, -1, -1, 0, 1 \rangle$)
VAS are the same as Petri nets:

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

transitions = transfers of tokens (e.g. \((0, -1, -1, 0, 1)\))
VAS are the same as Petri nets:

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

transitions = transfers of tokens (e.g. $(0,-1,-1,0,1)$)
VAS are the same as Petri nets:

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

transitions = transfers of tokens (e.g. \((0,-1,-1,0,1)\))
VAS are the same as **Petri nets**:

- **configurations** = tokens per location  
  \( (2, 1, 3, 0, 0) \)

- **transitions** = transfers of tokens  
  \( (0, -1, -1, 0, 1) \)
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 \textbf{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 $\text{FO}[\mathbb{N},+]$

e.g. $\varphi(x, y) = \exists z. x + y = z + z$
We may expect that reachable sets are **linear**... but they are not!

\[
(x + y) \leq z \leq O / (x + y)^2
\]
We may expect that reachable sets are **linear**... but they are not! 😞
We may expect that reachable sets are linear... but they are not!

\[
\begin{align*}
(x + y) &\leq z \leq O((x + y)^2) \\
\end{align*}
\]
To overcome the problem of representing reachable sets, we try to **over-approximate by downward closures**:

\[ V^\downarrow = \{ \bar{z} \mid \exists \bar{y} \in V. \bar{z} \leq \bar{y} \} \]
To overcome the problem of representing reachable sets, we try to **over-approximate by downward closures**:

\[ V^\downarrow = \{ \vec{z} \mid \exists \vec{y} \in V. \, \vec{z} \leq \vec{y} \} \]

⚠️ This is not an approximation for **lossy VAS**!
To overcome the problem of representing reachable sets, we try to over-approximate by downward closures:

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

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**:

\[ \mathcal{V}^\downarrow = \{ \bar{z} \mid \exists \bar{y} \in \mathcal{V}. \bar{z} \leq \bar{y} \} \]

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**:

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

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)

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
Saturation of downward-closed sets via transition function $\Delta$

acceleration on emerging dominating sets...
Karp & Miller Algorithm ’69
Saturation of downward-closed sets via transition function $\Delta$

acceleration on emerging dominating sets...

Example

Correctness of acceleration

$\bar{x}$ via $\bar{\delta}$ for some $\bar{\delta} \in \mathbb{N}$

$\bar{x}$ via $\bar{n} \cdot \bar{\delta} \leq \bar{x} + \lim_{n \to \infty} (n \cdot \bar{\delta})$
Karp & Miller Algorithm ’69
Saturation of downward-closed sets via transition function $\Delta$

acceleration on emerging dominating sets...

Example

\[
\begin{align*}
(-1, +1) & \quad (+1, -1) \\
(+2, 0) & \\
(0, +2) & \\
\end{align*}
\]
Karp & Miller Algorithm ’69
Saturation of downward-closed sets via transition function $\Delta$

acceleration on emerging dominating sets...

Example

$(-1, +1)$

$(+1, -1)$

$(+2, 0)$

$(0, +2)$
Karp & Miller Algorithm ’69

Saturation of downward-closed sets via transition function $\Delta$

$\Delta$ acceleration on emerging dominating sets...

Example

Correctness of acceleration $\bar{x}$/\$\rightarrow \bar{x} + \bar{\delta}$ for some $\bar{\delta} \in \mathbb{N}$

$\Rightarrow (\text{by linearity})$

$\bar{x}$/\$\rightarrow \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 $\Delta$

acceleration on emerging dominating sets...

Example

Correctness of acceleration

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

$\Downarrow$ (by linearity)

$\bar{x} \xrightarrow{*} \bar{x} + n \cdot \bar{\delta} \leq \bar{x} + \lim_{n \to \infty} (n \cdot \bar{\delta})$
Karp & Miller Algorithm ’69

Saturation of downward-closed sets via transition function $\Delta$

acceleration on emerging dominating sets...

Example

Correctness of acceleration

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

$\Downarrow$ (by linearity)

$\bar{x} \xrightarrow{\ast} \bar{x} + n \cdot \bar{\delta} \leq \bar{x} + \lim_{n \to \infty} (n \cdot \bar{\delta})$
Karp & Miller Algorithm ’69
Saturation of downward-closed sets via transition function $\Delta$
+ acceleration on emerging dominating sets...

Example

Correctness of acceleration $\overline{x}/\text{unital}/\rightarrow \overline{x} + \overline{\delta}$ for some $\overline{\delta} \in \mathbb{N}$

$\overline{x}/\text{unital}/\rightarrow \overline{x} + n \cdot \overline{\delta} \leq \overline{x} + \lim_{n \to \infty} (n \cdot \overline{\delta})$
Theorem (Rackoff ’78)

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

Corollary 1
Reachability on lossy VAS is EXPSPACE-complete.

Corollary 2
Control-state reachability on VAS is EXPSPACE-complete.
Theorem (Rackoff ’78)

**Coverability on VAS** (i.e. given $\bar{x}, \bar{y}$, tell if $\exists \bar{z} \geq \bar{y}. \bar{x}^* \rightarrow \bar{z}$) 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. \( \text{if } [x > 0] \text{ then } \ldots \))

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. $\text{if } [x > 0] \text{ then } \ldots$)

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

Unfortunately, acceleration for the above systems does not work, e.g. $(1, 0) \xrightarrow{\text{reset } x \ y:=1} (0, 1) \xrightarrow{x:=x+2 \ y:=y-1} (2, 0)$, but $(1, 0) \not\xrightarrow{\ast} (3, 0)$
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. $(1, 0) \xrightarrow{\text{reset } x \ y := 1} (0, 1) \xrightarrow{x := x+2} (2, 0)$, but $(1, 0) \not\xrightarrow{*} (3, 0)$

💡 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 $B_{n+1} = \{ \bar{x} \mid \exists \bar{y} \in B_n. \bar{x} \rightarrow \bar{y} \}$
Lemma

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

\[
\begin{array}{c}
\bar{x} \\
\downarrow \\
\bar{y}
\end{array}
\]
Lemma

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

\[ \forall \bar{x}' \rightarrow \bar{x} \rightarrow \bar{y} \]

Example of backward coverability analysis
Lemma

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

\[
\forall \overline{x}' \quad \exists \overline{y}'
\]

\[
\overline{x} \quad \overline{y}
\]
Lemma

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

\[ \forall \bar{x}' \rightarrow \exists \bar{y}' \]

\[ \bar{x} \rightarrow \bar{y} \]
Lemma

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

\[ \forall \bar{x}' \quad \exists \bar{y}' \]

Example of backward coverability analysis
Lemma

VAS transitions with resets, transfers, and positive guards are \textbf{backward-compatible with upward-closures}, i.e.

\begin{align*}
\forall \bar{x}' & \Rightarrow \exists \bar{y}' \\
\bar{x} & \Rightarrow \bar{y}
\end{align*}

\[ \forall x' \Rightarrow \exists y' \]

\[ \bar{x} \Rightarrow \bar{y} \]

Example of backward coverability analysis

\[ x = x - 2 \]

\[ y = y - 2 \]
Lemma

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

$$\forall \bar{x}' \quad \exists \bar{y}'$$

Example of backward coverability analysis

Termination by Dickson's Lemma:

- Infinitely many emerging points
- Infinite decreasing chain or antichain
Lemma

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

\[
\forall \bar{x}' \quad \exists \bar{y}'
\]

Example of backward coverability analysis

Termination by Dickson's Lemma: infinitely many emerging points \( \Rightarrow \) infinite decreasing chain or antichain
Lemma

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

\[ \forall \bar{x}' \quad \exists \bar{y}' \]

Example of backward coverability analysis
Lemma

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

$$\forall \bar{x}' \quad \exists \bar{y}'$$

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 $\bar{x} \xrightarrow*{\Delta} \bar{y}$, 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 Presburger logic $\text{FO}[\mathbb{N}, +]$)

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 \( \bar{x} \xrightarrow{\Delta} \bar{y} \), 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 Presburger 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 \( \bar{x} \xrightarrow{\cdot} \Delta \bar{y} \), 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 Presburger logic \(\text{FO}[\mathbb{N}, +]\))

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 \( \bar{x} \xrightarrow{\Delta} \bar{y} \), 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 Presburger logic \( \text{FO}[\mathbb{N},+] \))

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 \( \bar{x} \xrightarrow{\Delta} \bar{y} \), 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 Presburger logic \(\text{FO}[\mathbb{N}, +]\))
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 $\pi$ 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!”