Reachability in Vector Addition Systems is Primitive-Recursive in Fixed Dimension

Jérôme Leroux & Sylvain Schmitz

LICS 2019
vector addition systems (VAS)
  ▶ central model of computation

reachability problem
  ▶ hard conceptually and computationally
  ▶ decision via decomposition algorithm

this talk
  ▶ new complexity upper bounds
**Outline**

- vector addition systems (VAS)
  - central model of computation
- reachability problem
  - hard conceptually and computationally
  - decision via decomposition algorithm
- this talk
  - new complexity upper bounds
Outline

vector addition systems (VAS)
  ▶ central model of computation

reachability problem
  ▶ hard conceptually and computationally
  ▶ decision via decomposition algorithm

this talk
  ▶ new complexity upper bounds
Vector Addition Systems
VECTOR ADDITION SYSTEMS
Vector Addition Systems

Springfield Power Plant

(1,1) produce electricity

(0,1) uranium waste

(-1,-2) recycle uranium

electricity
Can we produce unbounded electricity with no left-over uranium waste?
Can we produce unbounded electricity with no left-over uranium waste? Yes, \((\infty, 0)\) is reachable.
IMPORTANCE OF THE PROBLEM

REACHABILITY PROBLEM

- input: *a vector addition system and two configurations source and target*
- question: *source →* target?

- modelling of discrete resources (items, money, molecules, active threads, active data domain, …)
- many decision problems interreducible with reachability
IMPORTANCE OF THE PROBLEM

REACHABILITY PROBLEM

input: a vector addition system and two configurations source and target

question: source $\rightarrow^*$ target?

- modelling of discrete resources (items, money, molecules, active threads, active data domain, ...)
- many decision problems interreducible with reachability
**Importance of the Problem**

- **1962**
  - R. J. Lipton: EXPSPACE lower bound

- **1976**
  - E. W. Mayr: decidability by decomposition
  - S. R. Kosaraju: decidability by decomposition

- **1981**
  - J.-L. Lambert: decidability by decomposition

- **1982**
  - J. Leroux: decidability by Presburger inductive invariants

- **1992**
  - J. Leroux & S.: cubic Ackermann upper bound ($\mathcal{F}_{\omega^3}$)

- **2011**
  - S.: quadratic Ackermann upper bound ($\mathcal{F}_{\omega^2}$)

- **2015, 2017, 2019**
  - W. Czerwinski, S. Lasota, R. Lazić, J. Leroux, F. Mazowiecki: Tower lower bound ($\mathcal{F}_3$)
NEW UPPER BOUNDS

\[ F_0(x) = x + 1 \]
\[ F_1(x) = F_0 \circ \cdots \circ F_0(x) = 2x + 1 \]
\[ F_2(x) = F_1 \circ \cdots \circ F_1(x) \approx 2^x \]
\[ F_3(x) = F_2 \circ \cdots \circ F_2(x) \approx \text{tower}(x) \]
\[ \vdots \]
\[ F_\omega(x) = F_{x+1}(x) \approx \text{ackermann}(x) \]

UPPER BOUND THEOREM
VAS Reachability is in \( F_\omega \), and in \( F_{d+4} \) in fixed dimension \( d \)
**New Upper Bounds**

\[
\begin{align*}
F_0(x) &= x + 1 \\
F_1(x) &= F_0 \circ \cdots \circ F_0(x) = 2x + 1 \\
F_2(x) &= F_1 \circ \cdots \circ F_1(x) \approx 2^x \\
F_3(x) &= F_2 \circ \cdots \circ F_2(x) \approx \text{tower}(x) \\
&\vdots \\
F_\omega(x) &= F_{x+1}(x) \approx \text{ackermann}(x)
\end{align*}
\]

**Upper Bound Theorem**

VAS Reachability is in \( F_\omega \), and in \( F_{d+4} \) in fixed dimension \( d \).
**New Upper Bounds**

\[ F_0(x) = x + 1 \]
\[ F_1(x) = F_0 \circ \cdots \circ F_0(x) = 2x + 1 \]
\[ x+1 \text{ times} \]
\[ F_2(x) = F_1 \circ \cdots \circ F_1(x) \approx 2^x \]
\[ x+1 \text{ times} \]
\[ F_3(x) = F_2 \circ \cdots \circ F_2(x) \approx \text{tower}(x) \]
\[ \vdots \]
\[ F_\omega(x) = F_{x+1}(x) \approx \text{ackermann}(x) \]

**Upper Bound Theorem**

*VAS Reachability is in \( F_\omega \), and in \( F_{d+4} \) in fixed dimension \( d \)*
“Simple Runs” (Θ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
“Simple Runs” (Θ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
“Simple Runs” (Θ Condition)

[Mayr‘81, Kosaraju‘82, Lambert‘92]

Characteristic System

\[ 0 + 1 \cdot a - 1 \cdot b = c \]
\[ 1 + 1 \cdot a - 2 \cdot b = 0 \]

Solution Path

\[ (0, -1) \]
“Simple Runs” (Θ Condition)

[Mayr‘81, Kosaraju‘82, Lambert‘92]
"Simple Runs" (Θ Condition)

[Mayr'81, Kosaraju'82, Lambert'92]
**“Simple Runs” (Θ Condition)**

[Mayr’81, Kosaraju’82, Lambert’92]

---

**Homogeneous System**

\[ 1 \cdot a - 1 \cdot b = c \]
\[ 1 \cdot a - 2 \cdot b = 0 \]
\[ a, b, c > 0 \]

**Unbounded Path**

---
“Simple Runs” (Ω Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
“Simple Runs” (Θ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
"Simple Runs" (Θ Condition)

[Mayr'81, Kosaraju'82, Lambert'92]
“Simple Runs” ($\Theta$ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]

Pumpable Paths

- **pump up**
  - $(0,1)$
  - $(\infty, \infty)$

- **pump down**
  - $(\infty, 0)$
  - $(\infty, \infty)$
“Simple Runs” (Θ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
"Simple Runs" (Θ Condition)

[Mayr'81, Kosaraju'82, Lambert'92]
“Simple Runs” (Θ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
“Simple Runs” ($\Theta$ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
“Simple Runs” (Θ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
"Simple Runs" ($\Theta$ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
"Simple Runs" (Θ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
"Simple Runs" (Θ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]
"Simple Runs" (Θ Condition)

[Mayr’81, Kosaraju’82, Lambert’92]

- Pump up: $\times 3$
- Solution path: $\times 1$
- Remainder: $\times 3$
- Pump down: $\times 3$
**DECOMPOSITION ALGORITHM**

[Mayr‘81, Kosaraju‘82, Lambert‘92]

can we build a “simple run”? 

\[
\begin{array}{c}
\rightarrow \\
\rightarrow \\
\end{array}
\]
**Decomposition Algorithm**

[Mayr’81, Kosaraju’82, Lambert’92]

**can we build a “simple run”?**  yes

\[ \{ \} \]

- ▶: no execution  
  ⇝: empty decomposition

- ▶: no or no  
  ▶: unfold and track bounded counter value

- ▶: bounded  
  ▶: saturate with bounded value
  ▶: bounded transition use: unfold and track bounded transition count
**DECOMPOSITION ALGORITHM**

[Mayr‘81, Kosaraju‘82, Lambert‘92]

can we build a “simple run”?  \( \text{no} \)

\[
\begin{align*}
\text{\rightarrow} & \quad \text{no} \\
\text{\rightarrow} & \quad \text{no}
\end{align*}
\]

\( \Rightarrow \) empty decomposition

\[
\begin{align*}
\text{\rightarrow} & \quad \text{bounded} \\
\infty & \quad \text{saturate with bounded value}
\end{align*}
\]

\[
\begin{align*}
\text{\rightarrow} & \quad \text{bounded transition use: unfold and track bounded transition count} \\
\text{\rightarrow} & \quad \text{no or no}
\end{align*}
\]

\( \Rightarrow \) unfold and track bounded counter value

\[
\begin{align*}
\{ & \quad \text{no}
\end{align*}
\]

decompose
**DECOMPOSITION ALGORITHM**

[Mayr’81, Kosaraju’82, Lambert’92]

Can we build a “simple run”? **No**

de decompose

{, , , }
**DECOMPOSITION ALGORITHM**

[Mayr’81, Kosaraju’82, Lambert’92]

Can we build a “simple run”? **no**

- **no**: no execution \(\mapsto\) empty decomposition
- **no**:
  - bounded \(\infty\): saturate with bounded value
  - bounded transition use: unfold and track bounded transition count
  - no \(\uparrow\) or no \(\downarrow\): unfold and track bounded counter value
**Decomposition Algorithm**

[Mayr’81, Kosaraju’82, Lambert’92]
**DECOMPOSITION ALGORITHM**

[Mayr’81, Kosaraju’82, Lambert’92]
**DECOMPOSITION ALGORITHM**

[Mayr’81, Kosaraju’82, Lambert’92]
**Termination**

**Ranking Function**

\[ \alpha_0 \]
Termination

Ranking Function

\[ \alpha_0 \lor \alpha_1 \]
**TERMINATION**

**RANKING FUNCTION**

\[ \alpha_0 \lor \alpha_1 \lor \alpha_2 \]
**Termination**

**Ranking Function**

\[ \alpha_0 \lor \alpha_1 \lor \alpha_2 \lor \ldots \]
TECHNICAL CONTRIBUTIONS

1. new ranking function:

order type $\omega^{d+1}$

$\omega^3$ in [Leroux & S. ’15]

$\omega^\omega \cdot (d + 1)$ in [S. ’17]

2. refined analysis of pumpable paths:

Rackoff-style analysis

improves complexity from $F_{2d+2}$ to $F_{d+4}$
**Technical Contributions**

1. **new ranking function:**
   
   order type $\omega^{d+1}$
   
   $\omega^{\omega^3}$ in [Leroux & S. ’15]
   
   $\omega^\omega \cdot (d + 1)$ in [S. ’17]

2. **refined analysis of pumpable paths:**

   Rackoff-style analysis improves complexity from $F_{2d+2}$ to $F_{d+4}$
**Technical Contributions**

1. **new ranking function:**
   
   order type $\omega^{d+1}$
   
   $\omega^3$ in [Leroux & S. ’15]
   $\omega \cdot (d + 1)$ in [S. ’17]

2. **refined analysis of pumpable paths:**
   
   Rackoff-style analysis
   improves complexity from $F_{2d+2}$ to $F_{d+4}$
TECHNICAL CONTRIBUTIONS

1. **new ranking function:**
   
   order type $\omega^{d+1}$
   
   $\omega^3$ in [Leroux & S. ’15]
   $\omega^\omega \cdot (d + 1)$ in [S. ’17]

2. **refined analysis of pumpable paths:**
   
   Rackoff-style analysis
   improves complexity from $F_{2d+2}$ to $F_{d+4}$
**Rank of a Transition**

For a transition $t$ in $(0, 1) \rightarrow (\infty, \infty) \rightarrow (\infty, 0)$

\[\{\text{effects of cycles } C \mid t \in C\}\]
**Rank of a Transition**

For a transition $t$ in $(0,1) \rightarrow (\infty,0)$, the rank is given by:

$$\{ m \cdot \vec{\alpha} + n \cdot \sqrt{\Box} \mid m \geq 0, n > 0 \}$$
**Rank of a Transition**

For a transition \( t \) in \((0,1)\) to \((\infty,0)\), we have:

\[
\text{span}_Q \left( \{ m \cdot \uparrow + n \cdot \sqrt{} \mid m \geq 0, n > 0 \} \right) = Q^2
\]
**Rank of a Transition**

For a transition \( t \) in \((0,1) \rightarrow (\infty,0)\)

\[
\dim\left( \text{span}_Q\left( \{ m \cdot \begin{array}{c} 1 \\ 0 \end{array} + n \cdot \begin{array}{c} 1 \\ 1 \end{array} \mid m \geq 0, n > 0 \} \right) \right) = \mathbb{Q}^2 = 2
\]
**Rank of a Transition**

For a transition $t$ in $(0,1)$ to $(\infty,0)$

$$\dim \left( \text{span}_Q \left( \{ m \cdot (1,1) + n \cdot (0,1) \mid m \geq 0, n > 0 \} \right) = Q^2 \right) = 2$$

Here, $\text{rank}(t) = (1,0,0) \in \mathbb{N}^{d+1}$

**Definition**

$$\text{rank}(G) \overset{\text{def}}{=} \sum_{t \in G} \text{rank}(t) \in \mathbb{N}^{d+1}$$

ordered lexicographically
**Rank of a VAS**

For a transition $t$ in $\langle 0,1 \rangle$ to $\langle \infty,0 \rangle$,

$$\dim \left( \text{span}_Q \left( \{ m \cdot \mathbf{1} + n \cdot \mathbf{2} \mid m \geq 0, n > 0 \} \right) \right) = Q^2 = 2$$

here, $\text{rank}(t) = (1,0,0) \in \mathbb{N}^{d+1}$

**Definition**

$$\text{rank}(G) \overset{\text{def}}{=} \sum_{t \in G} \text{rank}(t) \in \mathbb{N}^{d+1}$$ ordered lexicographically
**Decreasing Ranks**

**Recall:**

- no $\implies$ no execution $\leadsto$ empty decomposition

- bounded $\infty$: saturate with bounded value
- bounded transition use: unfold and track bounded transition count

- no $\rightarrow$ or no $\downarrow$: unfold and track bounded counter value
Decreasing Ranks

Recall:

- no $\bigtriangleup$: no execution $\rightarrow$ empty decomposition

- no $\blacklozenge$: 
  - bounded $\infty$: saturate with bounded value
  - bounded transition use: unfold and track bounded transition count

- no $\triangledown$ or no $\bigtriangledown$: unfold and track bounded counter value
**DECREASING RANKS**

**Proof Idea**

Consider a strongly connected VAS $G$:

Claim: if $T' \subset T$, then $\text{rank}(G') < \text{rank}(G)$

- let $V$, resp. $V'$ be the vector space associated to cycles of $T$, resp. $T'$
- we want to show $\text{dim}(V') < \text{dim}(V)$
- as $V' \subset V$, it suffices to show that $V' = V$ implies $T' = T$

  - pick cycle using every transition: effect $x + z + u + v \in V$
  - $V = V'$ thus $\exists \lambda \in \mathbb{Q}$ s.t. $x + z + u + v = \lambda (x + u + v)$
  - pick $p \in \mathbb{N}_{>0}$ s.t. $p \lambda \in \mathbb{Z}$
  - $\exists q \in \mathbb{N}$ s.t. $q a, q b, q c \geq p \lambda$
  - $[(p + qa - p \lambda)x, pz, (p + qb - p \lambda)u, (p + qc - p \lambda)v]$ also hom. sol
**Decreasing Ranks**

**Proof Idea**

Consider a strongly connected VAS $G$:

- $T \setminus T'$: not in any hom. sol.
- $T'$: in an homogeneous solution $[ax, bu, cv]$

Claim: if $T' \subsetneq T$, then $\text{rank}(G') < \text{rank}(G)$

- let $V$, resp. $V'$ be the vector space associated to cycles of $T$, resp. $T'$
- we want to show $\dim(V') < \dim(V)$
- as $V' \subseteq V$, it suffices to show that $V' = V$ implies $T' = T$
  - pick cycle using every transition: effect $x + z + u + v \in V$
  - $V = V'$ thus $\exists \lambda \in \mathbb{Q}$ s.t. $x + z + u + v = \lambda(x + u + v)$
  - pick $p \in \mathbb{N}_{>0}$ s.t. $p\lambda \in \mathbb{Z}$
  - $\exists q \in \mathbb{N}$ s.t. $qa, qb, qc \geq p\lambda$
  - $[(p + qa - p\lambda)x, pz, (p + qb - p\lambda)u, (p + qc - p\lambda)v]$ also hom. sol.
**Proof Idea**

Consider a strongly connected VAS $G$:

- $T \setminus T'$: not in any hom. sol.
- $T'$: in an homogeneous solution $[ax, bu, cv]$

Claim: if $T' \subsetneq T$, then $\text{rank}(G') < \text{rank}(G)$

- let $V$, resp. $V'$ be the vector space associated to cycles of $T$, resp. $T'$
- we want to show $\dim(V') < \dim(V)$
- as $V' \subseteq V$, it suffices to show that $V' = V$ implies $T' = T$
  - pick cycle using every transition: effect $x + z + u + v \in V$
  - $V = V'$ thus $\exists \lambda \in \mathbb{Q}$ s.t. $x + z + u + v = \lambda(x + u + v)$
  - pick $p \in \mathbb{N}_{>0}$ s.t. $p\lambda \in \mathbb{Z}$
  - $\exists q \in \mathbb{N}$ s.t. $qa, qb, qc \geq p\lambda$
  - $[(p + qa - p\lambda)x, pz, (p + qb - p\lambda)u, (p + qc - p\lambda)v]$ also hom. sol
**Decreasing Ranks**

**Proof Idea**
Consider a strongly connected VAS $G$:

- $T \setminus T'$: not in any hom. sol.
- $T'$: in an homogeneous solution $[ax, bu, cv]$

Claim: if $T' \subset T$, then $\text{rank}(G') < \text{rank}(G)$

- let $V$, resp. $V'$ be the vector space associated to cycles of $T$, resp. $T'$
- we want to show $\text{dim}(V') < \text{dim}(V)$
- as $V' \subseteq V$, it suffices to show that $V' = V$ implies $T' = T$

- pick cycle using every transition: effect $x + z + u + v \in V$
- $V = V'$ thus $\exists \lambda \in \mathbb{Q}$ s.t. $x + z + u + v = \lambda(x + u + v)$
- pick $p \in \mathbb{N}_{>0}$ s.t. $p\lambda \in \mathbb{Z}$
- $\exists q \in \mathbb{N}$ s.t. $qa, qb, qc \geq p\lambda$
- $[(p + qa - p\lambda)x, pz, (p + qb - p\lambda)u, (p + qc - p\lambda)v]$ also hom. sol
**Decreasing Ranks**

**Proof Idea**
Consider a strongly connected VAS $G$:

- $T \setminus T'$: not in any hom. sol.
- $T'$: in an homogeneous solution $[ax, bu, cv]$

Claim: if $T' \subsetneq T$, then $\text{rank}(G') < \text{rank}(G)$

- let $V$, resp. $V'$ be the vector space associated to cycles of $T$, resp. $T'$
- we want to show $\text{dim}(V') < \text{dim}(V)$
- as $V' \subseteq V$, it suffices to show that $V' = V$ implies $T' = T$

- pick cycle using every transition: effect $x + z + u + v \in V$
- $V = V'$ thus $\exists \lambda \in \mathbb{Q}$ s.t. $x + z + u + v = \lambda(x + u + v)$
- pick $p \in \mathbb{N}_{>0}$ s.t. $p\lambda \in \mathbb{Z}$
- $\exists q \in \mathbb{N}$ s.t. $qa, qb, qc \geq p\lambda$
- $[(p + qa - p\lambda)x, pz, (p + qb - p\lambda)u, (p + qc - p\lambda)v]$ also hom. sol
**DECREASING RANKS**

**Proof Idea**
Consider a strongly connected VAS \( G \):

- \( T \setminus T' \): not in any hom. sol.
- \( T' \): in an homogeneous solution \([ax, bu, cv]\)

Claim: if \( T' \subseteq T \), then \( \text{rank}(G') < \text{rank}(G) \)

- let \( V \), resp. \( V' \) be the vector space associated to cycles of \( T \), resp. \( T' \)
- we want to show \( \text{dim}(V') < \text{dim}(V) \)
- as \( V' \subseteq V \), it suffices to show that \( V' = V \) implies \( T' = T \)

- pick cycle using every transition: effect \( x + z + u + v \in V \)
- \( V = V' \) thus \( \exists \lambda \in \mathbb{Q} \) s.t. \( x + z + u + v = \lambda(x + u + v) \)
- \( \text{pick } p \in \mathbb{N}_{>0} \) s.t. \( p\lambda \in \mathbb{Z} \)
- \( \exists q \in \mathbb{N} \) s.t. \( qa, qb, qc \geq p\lambda \)
- \( [(p + qa - p\lambda)x, pz, (p + qb - p\lambda)u, (p + qc - p\lambda)v] \) also hom. sol
**Decreasing Ranks**

**Proof Idea**

Consider a strongly connected VAS $G$:

- $T \setminus T'$: not in any hom. sol.
- $T'$: in an homogeneous solution $[ax, bu, cv]$

Claim: if $T' \subseteq T$, then $\text{rank}(G') < \text{rank}(G)$

- let $V$, resp. $V'$ be the vector space associated to cycles of $T$, resp. $T'$
- we want to show $\text{dim}(V') < \text{dim}(V)$
- as $V' \subseteq V$, it suffices to show that $V' = V$ implies $T' = T$

- pick cycle using every transition: effect $x + z + u + v \in V$
- $V = V'$ thus $\exists \lambda \in \mathbb{Q}$ s.t. $x + z + u + v = \lambda(x + u + v)$
- pick $p \in \mathbb{N}_{>0}$ s.t. $p\lambda \in \mathbb{Z}$
- $\exists q \in \mathbb{N}$ s.t. $qa, qb, qc \geq p\lambda$
- $[(p + qa - p\lambda)x, pz, (p + qb - p\lambda)u, (p + qc - p\lambda)v]$ also hom. sol.
**Proof Idea**

Consider a strongly connected VAS $G$:

- $T \setminus T'$: not in any hom. sol.
- $T'$: in an homogeneous solution $[ax, bu, cv]$

Claim: if $T' \subsetneq T$, then $\text{rank}(G') < \text{rank}(G)$

- let $V$, resp. $V'$ be the vector space associated to cycles of $T$, resp. $T'$
- we want to show $\dim(V') < \dim(V)$
- as $V' \subseteq V$, it suffices to show that $V' = V$ implies $T' = T$
  - pick cycle using every transition: effect $x + z + u + v \in V$
  - $V = V'$ thus $\exists \lambda \in \mathbb{Q}$ s.t. $x + z + u + v = \lambda(x + u + v)$
  - pick $p \in \mathbb{N}_{>0}$ s.t. $p\lambda \in \mathbb{Z}$
  - $\exists q \in \mathbb{N}$ s.t. $qa, qb, qc \geq p\lambda$
  - $[(p + qa - p\lambda)x, pz, (p + qb - p\lambda)u, (p + qc - p\lambda)v]$ also hom. sol.
DECREASING RANKS

PROOF IDEA

Consider a strongly connected VAS $G$:

- $T \setminus T'$: not in any hom. sol.
- $T'$: in an homogeneous solution $[ax, bu, cv]$

Claim: if $T' \subsetneq T$, then $\text{rank}(G') < \text{rank}(G)$

- let $V$, resp. $V'$ be the vector space associated to cycles of $T$, resp. $T'$
- we want to show $\dim(V') < \dim(V)$
- as $V' \subseteq V$, it suffices to show that $V' = V$ implies $T' = T$

- pick cycle using every transition: effect $x + z + u + v \in V$
- $V = V'$ thus $\exists \lambda \in \mathbb{Q}$ s.t. $x + z + u + v = \lambda(x + u + v)$
- pick $p \in \mathbb{N}_{>0}$ s.t. $p\lambda \in \mathbb{Z}$
- $\exists q \in \mathbb{N}$ s.t. $qa, qb, qc \geq p\lambda$
- $[(p + qa - p\lambda)x, pz, (p + qb - p\lambda)u, (p + qc - p\lambda)v]$ also hom. sol
The Length of Decomposition Branches

Consequence of (S. ‘14)

The decomposition tree is of size at most $F_{d+4}(e(n))$ for some elementary function $e$. 
The length of decomposition branches

\[ \omega^{d+1} \]
\[ \lor \]
\[ \alpha_0 \]
\[ \lor \]
\[ \alpha_1 \]
\[ \lor \]
\[ \alpha_2 \]
\[ \lor \]
\[ \vdots \]

Consequence of (S. ’14)

The decomposition tree is of size at most \( F_{d+4}(e(n)) \) for some elementary function \( e \).
**The Length of Decomposition Branches**

The decomposition tree is of size at most $F_{d+4}(e(n))$ for some elementary function $e$. 

Consequence of (S. ’14)

*The decomposition tree is of size at most $F_{d+4}(e(n))$ for some elementary function $e.*
New Upper Bounds

\[ F_0(x) = x + 1 \]
\[ F_1(x) = F_0 \circ \cdots \circ F_0(x) = (x + 1)^{x+1 \text{ times}} \]
\[ F_2(x) = F_1 \circ \cdots \circ F_1(x) \approx 2^x \]
\[ F_3(x) = F_2 \circ \cdots \circ F_2(x) \approx \text{tower}(x) \]
\[ \vdots \]
\[ F_\omega(x) = F_{x+1}(x) \approx \text{ackermann}(x) \]

Upper Bound Theorem

VAS Reachability is in \( F_\omega \), and in \( F_{d+4} \) in fixed dimension \( d \)
New Upper Bounds

\[ F_0(x) = x + 1 \]
\[ F_1(x) = F_0 \circ \cdots \circ F_0(x) = 2x + 1 \]
\[ F_2(x) = F_1 \circ \cdots \circ F_1(x) \approx 2^x \]
\[ F_3(x) = F_2 \circ \cdots \circ F_2(x) \approx \text{tower}(x) \]
\[ \vdots \]
\[ F_\omega(x) = F_{x+1}(x) \approx \text{ackermann}(x) \]

Upper Bound Theorem

VAS Reachability is in \( F_\omega \), and in \( F_{d+4} \) in fixed dimension \( d \)

Theorem

VAS Reachability reduces to bounded VAS Reachability
A Related Problem

labelled VAS transitions carry labels from some alphabet $L(V, source, target)$ the language of labels in runs from source to target

$\downarrow L$ the set of scattered subwords of the words in the language $L$

Example

aba $\leq^* baaacabbbab$
A Related Problem

labelled VAS transitions carry labels from some alphabet $L(\mathcal{V}, \text{source}, \text{target})$ the language of labels in runs from source to target

$\downarrow L$ the set of scattered subwords of the words in the language $L$

Downwards Language Inclusion Problem

input: two labelled VAS $\mathcal{V}$ and $\mathcal{V}'$ and configurations source, target, source', target'

question: $\downarrow L(\mathcal{V}, \text{source}, \text{target}) \subseteq \downarrow L(\mathcal{V}', \text{source}', \text{target}')$?
A RELATED PROBLEM

**Downwards Language Inclusion Problem**

input: two labelled VAS \( V \) and \( V' \) and configurations

source, target, source', target'

question: \( \downarrow L(V, \text{source, target}) \subseteq \downarrow L(V', \text{source'}, \text{target'}) \) ?

**Theorem (Habermehl, Meyer & Wimmel’10)**

*Given a labelled VAS \( V \) and configurations source and target and its decomposition, one can construct a finite automaton for \( \downarrow L(V, \text{source, target}) \) in polynomial time.*

**Corollary**

The Downwards Language Inclusion is in \textbf{Ackermann}. 
A Related Problem

Downwards Language Inclusion Problem

input: two labelled VAS $\mathcal{V}$ and $\mathcal{V}'$ and configurations $\text{source}$, $\text{target}$, $\text{source}'$, $\text{target}'$

question: $\downarrow L(\mathcal{V}, \text{source}, \text{target}) \subseteq \downarrow L(\mathcal{V}', \text{source}', \text{target}')$?

Theorem (Habermehl, Meyer & Wimmel’10)

Given a labelled VAS $\mathcal{V}$ and configurations $\text{source}$ and $\text{target}$ and its decomposition, one can construct a finite automaton for $\downarrow L(\mathcal{V}, \text{source}, \text{target})$ in polynomial time.

Corollary

The Downwards Language Inclusion is in Ackermann.
A RELATED PROBLEM

**DOWNWARDS LANGUAGE INCLUSION PROBLEM**

**input:** two labelled VAS $\mathcal{V}$ and $\mathcal{V}'$ and configurations source, target, source', target'

**question:** $\downarrow L(\mathcal{V}, \text{source}, \text{target}) \subseteq \downarrow L(\mathcal{V}', \text{source}', \text{target}')$?

**COROLLARY**

The Downwards Language Inclusion is in ACKERMANN.

**THEOREM (Zetzsche’16)**

The Downwards Language Inclusion is ACKERMANN-hard.
PERSPECTIVES

1. complexity gap for VAS reachability
   ▶ **TOWER-hard** [Czerwinski et al.'19]
   ▶ decomposition algorithm: requires $F_\omega = \text{Ackermann}$ time,
     because downward language inclusion is $F_\omega$-hard [Zetzsche’16]

2. reachability in VAS extensions?
   ▶ decidable in VAS with hierarchical zero tests [Reinhardt’08]
   ▶ what about
     ▶ branching VAS
     ▶ unordered data Petri nets
     ▶ pushdown VAS
1. complexity gap for VAS reachability
   ▶ **TOWER-hard** [Czerwinski et al.’19]
   ▶ decomposition algorithm: requires $F_\omega = \text{ACKERMANN}$ time, because downward language inclusion is $F_\omega$-hard [Zetzsche’16]

2. reachability in VAS extensions?
   ▶ decidable in VAS with hierarchical zero tests [Reinhardt’08]
   ▶ what about
     ▶ branching VAS
     ▶ unordered data Petri nets
     ▶ pushdown VAS
1. complexity gap for VAS reachability
   ▶ **TOWER-hard** [Czerwinski et al.’19]
   ▶ decomposition algorithm: requires $F_{\omega} = \text{Ackermann}$ time, because downward language inclusion is $F_{\omega}$-hard [Zetzsche’16]

2. reachability in VAS extensions?
   ▶ decidable in VAS with hierarchical zero tests [Reinhardt’08]
   ▶ what about
     ▶ branching VAS
     ▶ unordered data Petri nets
     ▶ pushdown VAS