On Boundedness Problems for Pushdown Vector Addition Systems

Jérôme Leroux   Grégoire Sutre   **Patrick Totzke**

September 21, 2015

Vector Addition Systems – Recap

Definition

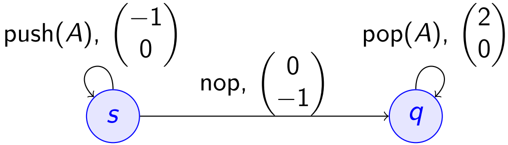
A VAS is a finite set of vectors $\mathbf{a} \in \mathbb{Z}^d$. For $\mathbf{v}, \mathbf{v}' : \mathbb{N}^d$ it has a step

$$\mathbf{v} \xrightarrow{\mathbf{a}} \mathbf{v}' \quad \text{if} \quad \mathbf{v}' = \mathbf{v} + \mathbf{a}.$$

- Equivalent to Petri Nets
  (concurrency, weak counters, event systems)
- Reachability: decidable
  Mayr'81,*Kosaraju'82*, . . . *Leroux and Schmitz'15*
- Coverability, Boundedness: ExpSpace-complete
  Lipton'76, *Rackoff'78*
- Most Games/Equivalences undecidable (e.g. Bisimulation)
  Jančar'95

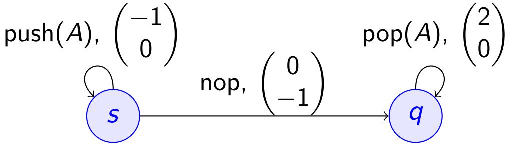
Pushdown Vector Addition Systems

. . . are products of VAS with pushdown automata.



Pushdown Vector Addition Systems

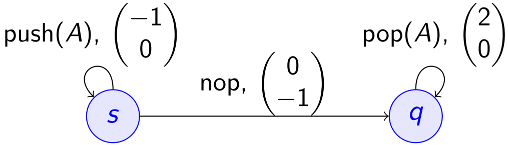
. . . are products of VAS with pushdown automata.



$s, \bot, \begin{pmatrix} 2 \\ 1 \end{pmatrix}$

Pushdown Vector Addition Systems

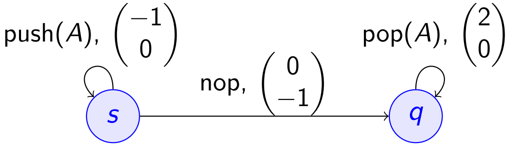
...are products of VAS with pushdown automata.



$$s, \perp, \begin{pmatrix}2\\1\end{pmatrix} \longrightarrow\longrightarrow s, AA\perp, \begin{pmatrix}0\\1\end{pmatrix}$$

Pushdown Vector Addition Systems

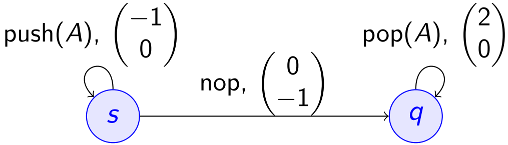
... are products of VAS with pushdown automata.



$$s, \bot, \begin{pmatrix}2\\1\end{pmatrix} \;\longrightarrow\longrightarrow\; s, AA\bot, \begin{pmatrix}0\\1\end{pmatrix} \;\longrightarrow\; q, AA\bot, \begin{pmatrix}0\\0\end{pmatrix}$$

Pushdown Vector Addition Systems

...are products of VAS with pushdown automata.

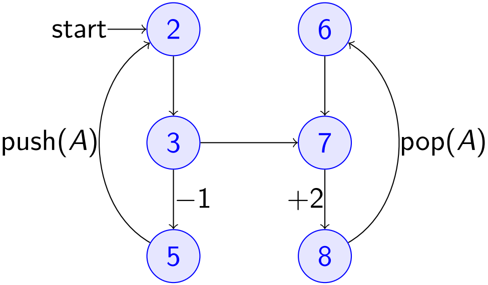


$$s, \bot, \begin{pmatrix}2\\1\end{pmatrix} \;\longrightarrow\longrightarrow\; s, AA\bot, \begin{pmatrix}0\\1\end{pmatrix} \;\longrightarrow\; q, AA\bot, \begin{pmatrix}0\\0\end{pmatrix} \;\longrightarrow\longrightarrow\; q, \bot, \begin{pmatrix}4\\0\end{pmatrix}$$

Pushdown Vector Addition Systems

. . . are products of VAS with pushdown automata. They can for example model recursive prorams with variables over $\mathbb{N}$.

```
x ← n
procedure DOUBLEX
    if (⋆ ∧ x > 0) then
        x ← (x − 1)
        DOUBLEX
    end if
    x ← (x + 2)
end procedure
```



Pushdown Vector Addition Systems

- Reachability $=$ Coverability $(=$ State-Reachability$)$
  TOWER-hard *Lazic'13*

Pushdown Vector Addition Systems

- Reachability d dim. $=$ Coverability $d+1$ dim.
  TOWER-hard *Lazic'13*

Pushdown Vector Addition Systems

- Reachability d dim. $=$ Coverability $d+1$ dim.
  TOWER-hard *Lazic'13*
- Coverability in 1 dim. is decidable *Leroux, Sutre, and T.'15*

Pushdown Vector Addition Systems

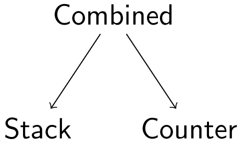
- Reachability d dim. $=$ Coverability $d+1$ dim.
  TOWER-hard *Lazic'13*
- Coverability in 1 dim. is decidable *Leroux, Sutre, and T.'15*
- Boundedness: decidable with Hyper-Ackermannian bounds
  Leroux, Praveen, and Sutre'14

Theorem [LSP'14]

If a PVAS configuration $(p, \perp, n)$ is bounded then the cardinality of the reachability set is at most $F_{\omega^d \cdot |Q|}(d \cdot n)$.

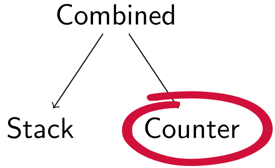
Pushdown Vector Addition Systems

- Reachability d dim. $=$ Coverability $d+1$ dim.
  TOWER-hard *Lazic'13*
- Coverability in 1 dim. is decidable *Leroux, Sutre, and T.'15*
- Boundedness: decidable with Hyper-Ackermannian bounds
  Leroux, Praveen, and Sutre'14
- Counter-, Stack-, and Combined Boundedness Problems.



Pushdown Vector Addition Systems

- Reachability d dim. $=$ Coverability $d+1$ dim.
  TOWER-hard *Lazic'13*
- Coverability in 1 dim. is decidable *Leroux, Sutre, and T.'15*
- Boundedness: decidable with Hyper-Ackermannian bounds
  Leroux, Praveen, and Sutre'14
- Counter-, Stack-, and Combined Boundedness Problems.



The following is in EXPTIME.

1-PVAS Counter-Boundedness

Given: 1-dim. PVAS, initial configuration (p, w, a).

Question: is $\{b \mid (p,w,a) \xrightarrow{*} (p', w', b)\}$ infinite?

Another Perspective

Definition (Context-free Controlled VAS)

a VAS $\mathbf{A} \subseteq \mathbb{Z}^d$ together with a context-free language $\mathcal{L} \subseteq A^*$.
There is a step $\mathbf{s} \longrightarrow \mathbf{t}$ between $\mathbf{s}, \mathbf{t} \in \mathbb{N}^d$ if

$$\mathbf{a_1 a_2} \ldots \mathbf{a_k} \in \mathcal{L} \quad \text{and} \quad \mathbf{s} \xrightarrow{\mathbf{a_1}} \xrightarrow{\mathbf{a_2}} \cdots \xrightarrow{\mathbf{a_k}} \mathbf{t}.$$

Another Perspective

Definition (Context-free Controlled VAS)

a VAS $\mathbf{A} \subseteq \mathbb{Z}^d$ together with a context-free language $\mathcal{L} \subseteq A^*$.
There is a step $\mathbf{s} \longrightarrow \mathbf{t}$ between $\mathbf{s}, \mathbf{t} \in \mathbb{N}^d$ if

$$\mathbf{a_1 a_2} \ldots \mathbf{a_k} \in \mathcal{L} \quad \text{and} \quad \mathbf{s} \xrightarrow{\mathbf{a_1}}\xrightarrow{\mathbf{a_2}} \cdots \xrightarrow{\mathbf{a_k}} \mathbf{t}.$$

Theorem

For Cf-Controlled VAS, Coverability (and Reachability) logspace reduces to Boundedness.

Another Perspective

Definition (Context-free Controlled VAS)

a VAS $\mathbf{A} \subseteq \mathbb{Z}^d$ together with a context-free language $\mathcal{L} \subseteq A^*$.
There is a step $\mathbf{s} \longrightarrow \mathbf{t}$ between $\mathbf{s}, \mathbf{t} \in \mathbb{N}^d$ if

$$\mathbf{a_1 a_2} \ldots \mathbf{a_k} \in \mathcal{L} \quad \text{and} \quad \mathbf{s} \xrightarrow{\mathbf{a_1}}\xrightarrow{\mathbf{a_2}} \cdots \xrightarrow{\mathbf{a_k}} \mathbf{t}.$$

Theorem

For Cf-Controlled VAS, Coverability (and Reachability) logspace reduces to Boundedness.

Observation

Relevant for the PVAS boundedness problem is the *trace* language $\{w \in \mathbf{A}^* \mid (p_0, \perp) \xrightarrow{w}\}$ defined by the PDA.

Another Perspective

Definition (Context-free Controlled VAS)

a VAS $\mathbf{A} \subseteq \mathbb{Z}^d$ together with a context-free language $\mathcal{L} \subseteq A^*$.
There is a step $\mathbf{s} \longrightarrow \mathbf{t}$ between $\mathbf{s}, \mathbf{t} \in \mathbb{N}^d$ if

$$\mathbf{a_1 a_2 \dots a_k} \in \mathcal{L} \quad \text{and} \quad \mathbf{s} \xrightarrow{\mathbf{a_1}} \xrightarrow{\mathbf{a_2}} \cdots \xrightarrow{\mathbf{a_k}} \mathbf{t}.$$

Theorem

For Cf-Controlled VAS, Coverability (and Reachability) logspace reduces to Boundedness.

Observation

Relevant for the PVAS boundedness problem is the *trace* language $\{w \in \mathbf{A}^* \mid (p_0, \bot) \xrightarrow{w}\}$ defined by the PDA.

prefix-closed!

Another Perspective

Definition (Context-free Controlled VAS)

a VAS $\mathbf{A} \subseteq \mathbb{Z}^d$ together with a context-free language $\mathcal{L} \subseteq A^*$.
There is a step $\mathbf{s} \longrightarrow \mathbf{t}$ between $\mathbf{s}, \mathbf{t} \in \mathbb{N}^d$ if

$$\mathbf{a_1 a_2 \ldots a_k} \in \mathcal{L} \quad \text{and} \quad \mathbf{s} \xrightarrow{\mathbf{a_1}}\xrightarrow{\mathbf{a_2}} \cdots \xrightarrow{\mathbf{a_k}} \mathbf{t}.$$

Theorem

For Cf-Controlled VAS, Coverability (and Reachability) logspace reduces to Boundedness.

Observation

Relevant for the PVAS boundedness problem is the *trace* language $\{w \in \mathbf{A}^* \mid (p_0, \bot) \xrightarrow{w}\}$ defined by the PDA.

Main Theorem

Boundedness of 1-dim VAS controlled by a prefix-closed language is in ExpTime.

Another Perspective

given as GfG

Definition (Context-free Controlled VAS)

a VAS $\mathbf{A} \subseteq \mathbb{Z}^d$ together with a context-free language $\mathcal{L} \subseteq A^*$.
There is a step $\mathbf{s} \xrightarrow{X} \mathbf{t}$ between $\mathbf{s}, \mathbf{t} \in \mathbb{N}^d$ if

$$X \overset{*}{\Longrightarrow} \mathbf{a_1 a_2 \ldots a_k} \quad \text{and} \quad \mathbf{s} \xrightarrow{\mathbf{a_1}} \xrightarrow{\mathbf{a_2}} \cdots \xrightarrow{\mathbf{a_k}} \mathbf{t}.$$

Theorem

For Cf-Controlled VAS, Coverability (and Reachability) logspace reduces to Boundedness.

Observation

Relevant for the PVAS boundedness problem is the *trace* language $\{w \in \mathbf{A}^* \mid (p_0, \bot) \xrightarrow{w}\}$ defined by the PDA.

Main Theorem

Boundedness of 1-dim VAS controlled by a prefix-closed language is in EXPTIME.

Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.

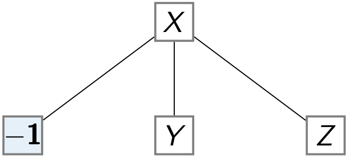
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.

$\boxed{X}$

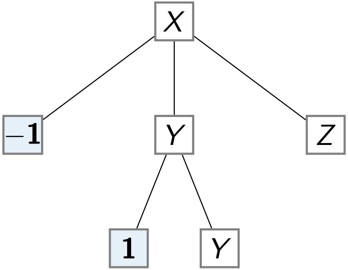
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



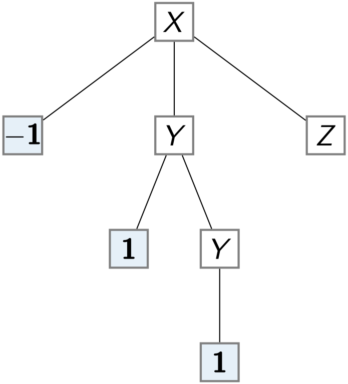
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



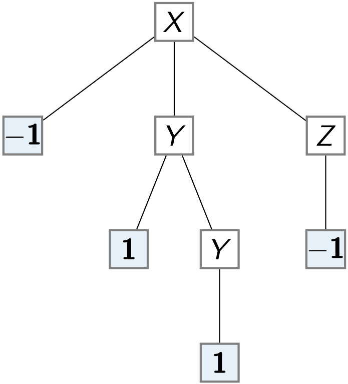
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



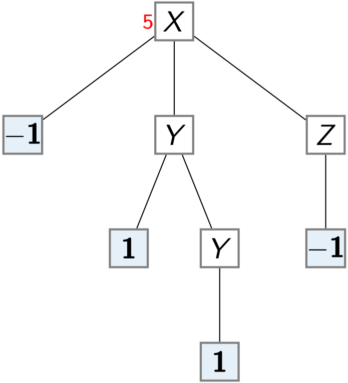
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



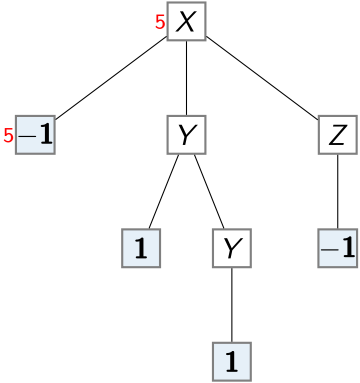
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



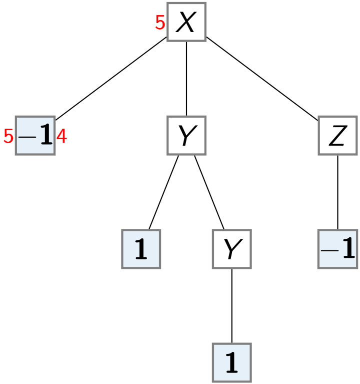
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



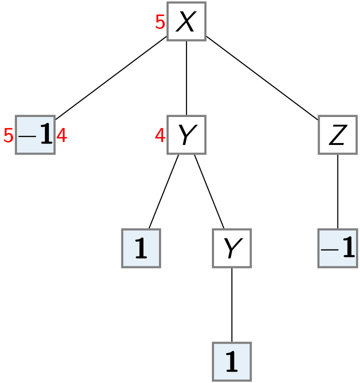
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



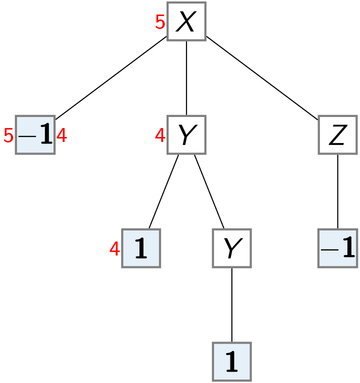
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



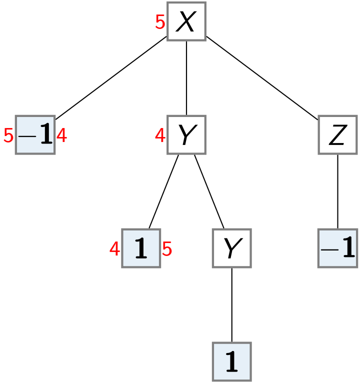
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



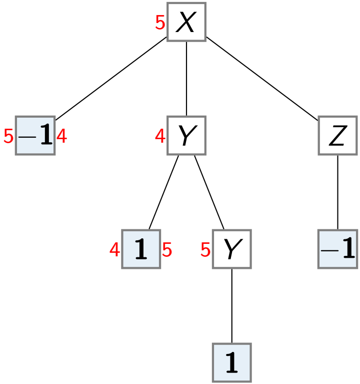
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



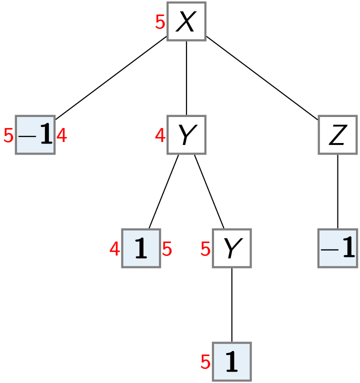
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



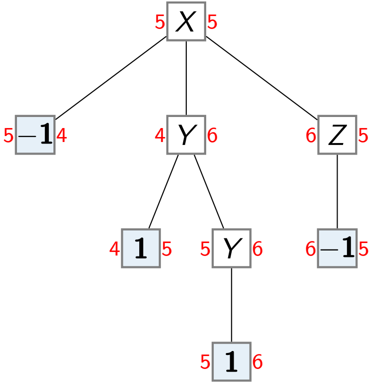
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



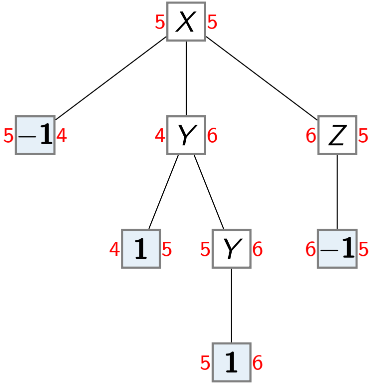
Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



Flow Trees

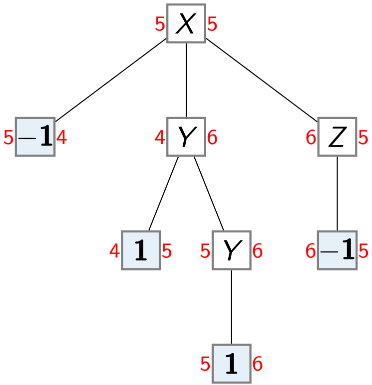
A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



$_a\boxed{X}_b$ means $a \xrightarrow{X} b' \geq b$; $\quad {}_{-\infty}\boxed{X}_b$ means $\exists a \in \mathbb{N}.\ a \xrightarrow{X} b' \geq b$.

Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.

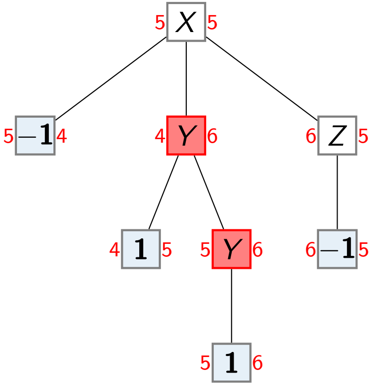


$T \sqsubseteq T'$ if

1. $|T| < |T'|$ or
2. $|T| = |T'|$ and $\sum$(labels $> -\infty$) on T is smaller than on T'

Flow Trees

A derivation tree with *consistent* in/out labels in $\mathbb{Z} \cup \{-\infty\}$.



$T \sqsubseteq T'$ if

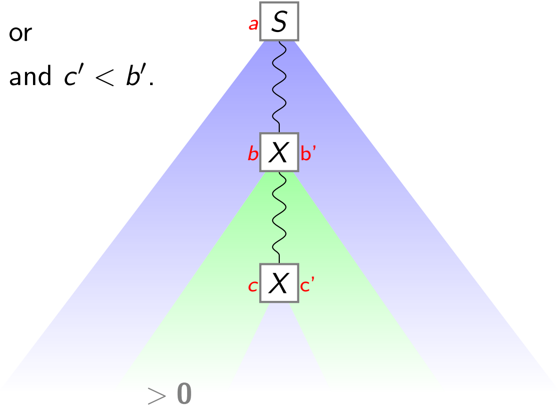
1. $|T| < |T'|$ or
2. $|T| = |T'|$ and $\sum$(labels $> -\infty$) on T is smaller than on T'

Certificates

Definition

A *certificate* is a flow tree with a node $_b\boxed{X}_{b'}$ and a descendant $_c\boxed{X}_{c'}$ such that

1. $b < c$ or
2. $b = c$ and $c' < b'$.

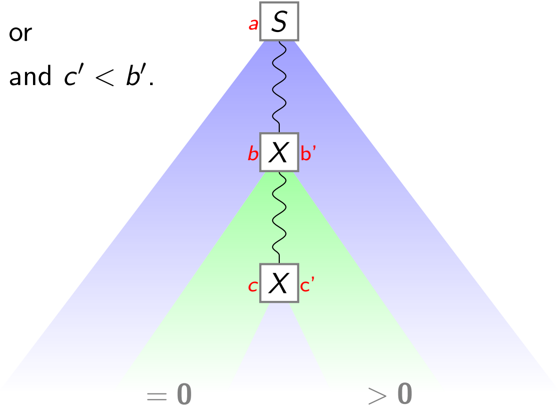


Certificates

Definition

A *certificate* is a flow tree with a node $_{b}\boxed{X}_{b'}$ and a descendant $_{c}\boxed{X}_{c'}$ such that

1. $b < c$ or
2. $b = c$ and $c' < b'$.



Certificates (cont.)

Theorem

$\{a' \mid a \xrightarrow{S} a'\}$ *is infinite iff there is a certificate with root* $(\leq a)$ $\boxed{S}$.

Unboundedness $\implies$ Certificate:

- $a \xrightarrow{S} b$ for sufficiently large b

Certificates (cont.)

Theorem
$\{a' \mid a \xrightarrow{S} a'\}$ *is infinite iff there is a certificate with root* $(\leq a)$ $\boxed{S}$.

Unboundedness $\implies$ Certificate:

- $a \xrightarrow{S} b$ for sufficiently large b
- a minimal flow tree must have long branch

Certificates (cont.)

Theorem

$\{a' \mid a \xrightarrow{S} a'\}$ is infinite iff there is a certificate with root $_{(\leq a)}\boxed{S}$.

Unboundedness $\implies$ Certificate:

- $a \xrightarrow{S} b$ for sufficiently large b
- a minimal flow tree must have long branch
- wqo $(\leq, =)$ on (input $\times V$) implies matching nodes with $b \leq c$

Certificates (cont.)

Theorem
$\{a' \mid a \xrightarrow{S} a'\}$ *is infinite iff there is a certificate with root* $(\leq a)\boxed{S}$.

Unboundedness $\Longrightarrow$ Certificate:

- $a \xrightarrow{S} b$ for sufficiently large b
- a minimal flow tree must have long branch
- wqo $(\leq, =)$ on (input $\times V$) implies matching nodes with $b \leq c$
- minimality excludes $b = c \wedge c' > b'$

Certificates (cont.)

Theorem
$\{a' \mid a \xrightarrow{S} a'\}$ *is infinite iff there is a certificate with root* $(\leq a)\boxed{S}$.

Unboundedness $\implies$ Certificate:

- $a \xrightarrow{S} b$ for sufficiently large b
- a minimal flow tree must have long branch
- wqo $(\leq, =)$ on (input $\times V$) implies matching nodes with $b \leq c$
- minimality excludes $b = c \wedge c' > b'$

Unboundedness $\impliedby$ Certificate:

- yield is $uvwxy \in \mathcal{L}$ with $\sum v \geq 0$ and $\sum v + \sum x > 0$

Certificates (cont.)

Theorem

$\{a' \mid a \xrightarrow{S} a'\}$ *is infinite iff there is a certificate with root* $(\leq a)\boxed{S}$.

Unboundedness $\implies$ Certificate:

- $a \xrightarrow{S} b$ for sufficiently large b
- a minimal flow tree must have long branch
- wqo $(\leq, =)$ on $(\text{input} \times V)$ implies matching nodes with $b \leq c$
- minimality excludes $b = c \wedge c' > b'$

Unboundedness $\impliedby$ Certificate:

- yield is $uvwxy \in \mathcal{L}$ with $\sum v \geq 0$ and $\sum v + \sum x > 0$
- All $uv^n wx^n$ are in $\mathcal{L}$ and executable.

Certificates (cont.)

Theorem
$\{a' \mid a \xrightarrow{S} a'\}$ *is infinite iff there is a certificate with root* $(\leq a)\boxed{S}$.

Unboundedness $\Longrightarrow$ Certificate:

- $a \xrightarrow{S} b$ for sufficiently large b
- a minimal flow tree must have long branch
- wqo $(\leq, =)$ on (input $\times V$) implies matching nodes with $b \leq c$
- minimality excludes $b = c \wedge c' > b'$

Unboundedness $\Longleftarrow$ Certificate:

- yield is $uvwxy \in \mathcal{L}$ with $\sum v \geq 0$ and $\sum v + \sum x > 0$
- All $uv^n wx^n$ are in $\mathcal{L}$ and executable.
- Prefix-closedness of $\mathcal{L}$ implies uv^n and $uv^n wx^n \in \mathcal{L}$.

Certificates (cont.)

Theorem

$\{a' \mid a \xrightarrow{S} a'\}$ *is infinite iff there is a certificate with root* $(\leq a)$ $\boxed{S}$.

Unboundedness $\implies$ Certificate:

- $a \xrightarrow{S} b$ for sufficiently large b
- a minimal flow tree must have long branch
- wqo $(\leq, =)$ on (input $\times V$) implies matching nodes with $b \leq c$
- minimality excludes $b = c \wedge c' > b'$

Unboundedness $\impliedby$ Certificate:

- yield is $uvwxy \in \mathcal{L}$ with $\sum v \geq 0$ and $\sum v + \sum x > 0$
- All $uv^n wx^n$ are in $\mathcal{L}$ and executable.
- Prefix-closedness of $\mathcal{L}$ implies uv^n and $uv^n wx^n \in \mathcal{L}$.

Bounding $\sqsubset$-minimal Certificates

(maybe on blackboard if time)

Theorem

Let $G = (V, \mathbf{A}, R, S)$ be a CfG generating a prefix-closed language over $\mathbf{A} = \{-1, 0, 1\}$ and $n \in \mathbb{N}$ an initial value. Then $\{m \mid n \xrightarrow{S} m\}$ is infinite iff it admits a certificate with height and all input/output values bounded by $n + 4^{4(|V|+1)}$.

Conclusion

Discussed here

- Pushdown VAS; Boundedness of counter/stack/both
- Cf-controlled VAS; Flow Trees
- prefix-closed control $\sim$ counter-Boundedness
- Counter-Boundedness in 1-PVAS is in $\textsc{ExpTime}$

Open Problems

- Decidability of PVAS Reachability (even in dim 1)
- is Boundedness reducible to Reachability in Cf-C-VAS?
- Complexity of 1-PVAS counter-Boundedness (NP– $\textsc{ExpTime}$)
- Complexity of 1-PVAS Coverability (NP– $\textsc{ExpSpace}$)

Conclusion

Discussed here

- Pushdown VAS; Boundedness of counter/stack/both
- Cf-controlled VAS, Flow Trees
- prefix-closed control ∼ Counter-Boundedness
- Counter-Boundedness in 1-PVAS is in EXPTIME

Open Problem

- Decidability of PVAS Reachability (even in dim 1)
- is Boundedness reducible to Reachability in Cf-C VAS?
- Complexity of 1-PVAS counter-Boundedness (NP– EXPTIME)
- Complexity of 1-PVAS Coverability (NP– EXPSPACE)

Additional Stuff

Weak Computation of Ackermann Functions A_m

$$A_m(n) \overset{\text{def}}{=} \begin{cases} n+1 & \text{if } m = 0 \\ A_{m-1}^{n+1}(1) & \text{if } m > 0 \end{cases}$$

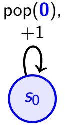
Weak Computation of Ackermann Functions A_m

$$A_m(n) \stackrel{\text{def}}{=} \begin{cases} n+1 & \text{if } m = 0 \\ A_{m-1}^{n+1}(1) & \text{if } m > 0 \end{cases}$$

$$\begin{aligned} A_0(n) &= n+1 \\ A_1(n) &= n+2 \\ A_2(n) &= 2n+2 \\ A_3(n) &= 2^n - 1 \\ &\vdots \end{aligned}$$

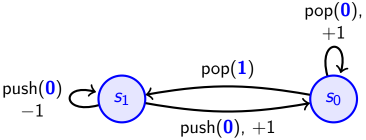
Weak Computation of Ackermann Functions A_m

$$A_m(n) \stackrel{\text{def}}{=} \begin{cases} n+1 & \text{if } m=0 \\ A_{m-1}^{n+1}(1) & \text{if } m>0 \end{cases}$$



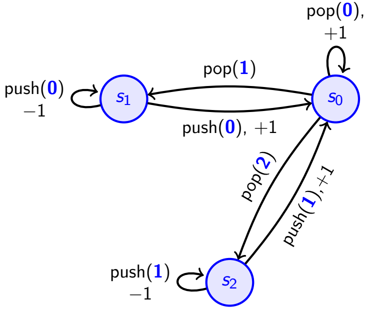
Weak Computation of Ackermann Functions A_m

$$A_m(n) \stackrel{\text{def}}{=} \begin{cases} n+1 & \text{if } m = 0 \\ A_{m-1}^{n+1}(1) & \text{if } m > 0 \end{cases}$$



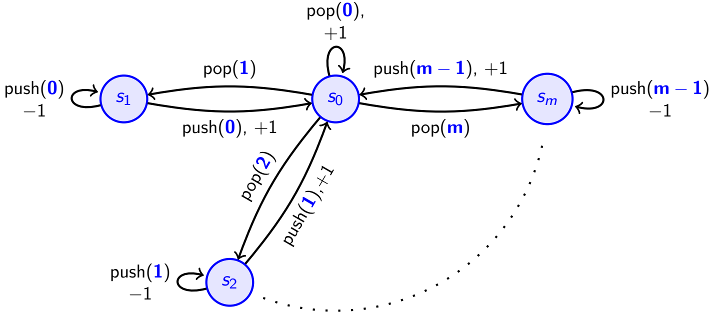
Weak Computation of Ackermann Functions A_m

$$A_m(n) \stackrel{\text{def}}{=} \begin{cases} n+1 & \text{if } m=0 \\ A_{m-1}^{n+1}(1) & \text{if } m>0 \end{cases}$$



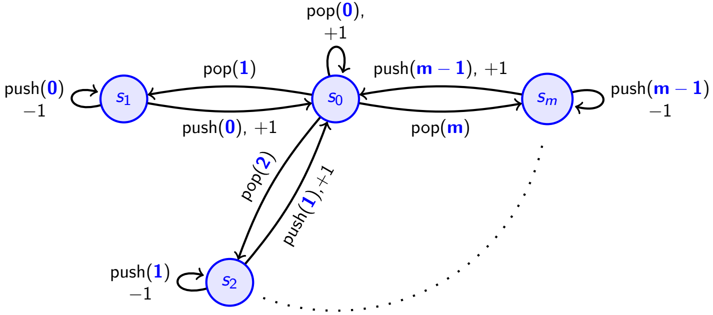
Weak Computation of Ackermann Functions A_m

$$A_m(n) \stackrel{\text{def}}{=} \begin{cases} n+1 & \text{if } m = 0 \\ A_{m-1}^{n+1}(1) & \text{if } m > 0 \end{cases}$$



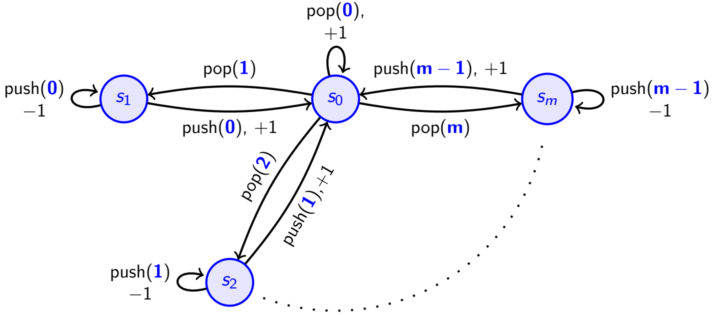
Weak Computation of Ackermann Functions A_m

$$A_m(n) \stackrel{\text{def}}{=} \begin{cases} n+1 & \text{if } m = 0 \\ A_{m-1}^{n+1}(1) & \text{if } m > 0 \end{cases}$$



Weak Computation of Ackermann Functions A_m

$$A_m(n) \stackrel{\text{def}}{=} \begin{cases} n+1 & \text{if } m=0 \\ A_{m-1}^{n+1}(1) & \text{if } m>0 \end{cases}$$



$$(s_0, \mathbf{m}\bot, n) \xrightarrow{*} (s_0, \bot, A_m(n))$$

If $(s_0, \mathbf{m}\bot, n) \xrightarrow{*} (s_0, \bot, n')$ then $n' \leq A_m(n)$