# On the Coverability Problem for Pushdown Vector Addition Systems in One Dimension*

Jérôme Leroux[1], Grégoire Sutre[1], and Patrick Totzke[2]

[1] Univ. Bordeaux & CNRS, LaBRI, UMR 5800, Talence, France
[2] Department of Computer Science, University of Warwick, UK

**Abstract.** Does the trace language of a given vector addition system (VAS) intersect with a given context-free language? This question lies at the heart of several verification questions involving recursive programs with integer parameters. In particular, it is equivalent to the coverability problem for VAS that operate on a pushdown stack. We show decidability in dimension one, based on an analysis of a new model called grammar-controlled vector addition systems.

## 1 Introduction

Pushdown systems are a well-known and natural formalization of recursive programs. Vector addition systems (VAS) are widely used to model concurrent systems and programs with integer variables. Pushdown vector addition systems (pushdown VAS) combine the two: They are VAS extended with a pushdown stack and allow to model, for instance, asynchronous programs [6] and, more generally, programs with recursion and integer variables.

Despite the model's relevance for automatic program verification, most classical model-checking problems are so far only partially solved. Termination and boundedness are decidable but their complexity is open [12]. Coverability and reachability are known to be TOWER-hard [9], but their decidability is open. In fact, reachability and the seemingly simpler coverability problem are essentially the same for pushdown VAS: there is a simple logarithmic-space reduction from reachability to coverability that only adds one extra dimension.

*Contributions.* Our main result is that coverability is decidable for 1-dimensional pushdown VAS. We work with a new grammar-based model called grammar-controlled vector addition systems (GVAS), which amounts to VAS restricted to firing sequences defined by a context-free grammar. In dimension one, this model corresponds to two-stack pushdown systems where one of the two stacks uses a single stack symbol. To prove our main result, we show that it is enough to check finitely many potential certificates of coverability. The latter are parse trees of the context-free grammar annotated with counter information from the 1-dimensional VAS. We truncate these annotated parse trees thanks to an

* This work was partially supported by ANR project REACHARD (ANR-11-BS02-001).

analysis of the asymptotic behaviour of the summary function induced by the 1-dimensional GVAS. Asymptotically-linear summary functions are shown to be effectively Presburger-definable, which makes the above truncation effective.

*Related work.* This paper continues a line of research that investigates the limitations of extending VAS while preserving the decidability of important verification questions, such as reachability, coverability and boundedness.

The coverability and boundedness problems for ordinary VAS are long known to be ExpSpace-complete [14,16] and reachability is decidable [15,8,11]. In recent years, several extensions of VAS have been considered with respect to decidability and complexity of reachability problems. For instance, Reinhardt [17] showed that reachability remains decidable for VAS in which one dimension can be tested for zero. *Branching VAS* introduce split-transitions and can be interpreted as bottom-up or top-down tree acceptors. *Alternating VAS* add a limited form of alternation where only one player is affected by the counters. Coverability and boundedness in these models are 2-ExpTime-complete [5,4], reachability is Tower-hard for branching and undecidable for alternating VAS [10,4].

Closer to this paper is the work of Bouajjani, Habermehl and Mayr [3], who study a model called BPA($\mathbb{Z}$). These are context-free grammars where nonterminals carry an integer parameter that can be evaluated and passed on when applying a production rule. They show how to compute a symbolic representation of the reachability set. Their formalism, like the 1-dimensional GVAS considered here, can model recursive programs with one integer variable. But while BPA($\mathbb{Z}$) allows arbitrary Presburger-definable operations on the variable, it cannot model return values.

Atig and Ganty [1] also study the context-free restriction of the reachability relation in vector addition systems. Instead of restricting the dimension of the VAS, they restrict the context-free language and show that reachability is decidable for the subclass of indexed context-free languages.

*Outline.* We first recall some background and notation for context-free grammars. Section 3 formally introduces grammar-controlled vector addition systems, their coverability problem and the required technology to solve it in dimension one. In Section 4, we show the existence of small certificates. These are subsequently proved to be recursive in two steps. Section 5 shows that, for so-called thin GVAS, the step relation is effectively Presburger-definable. Then, summary functions are shown to be computable by reduction to the thin case in Section 6.

## 2 Preliminaries

We let $\overline{\mathbb{R}} \stackrel{\text{def}}{=} \mathbb{R} \cup \{-\infty, +\infty\}$ denote the extended real number line and use the standard extensions of $+$ and $\leq$ to $\overline{\mathbb{R}}$. Recall that $(\overline{\mathbb{R}}, \leq)$ is a complete lattice. $\overline{\mathbb{Z}} \stackrel{\text{def}}{=} \mathbb{Z} \cup \{-\infty, +\infty\}$ and $\overline{\mathbb{N}} \stackrel{\text{def}}{=} \mathbb{N} \cup \{-\infty, +\infty\}$ denote the (complete) sublattices of extended integers and extended natural numbers, respectively.[3]

[3] Our extension of $\mathbb{N}$ contains $-\infty$ for technical reasons.

*Words.* Let $A^*$ be the set of all finite *words* over the alphabet $A$. The *empty word* is denoted by $\varepsilon$. We write $|w|$ for the *length* of a word $w$ in $A^*$ and $w^k \stackrel{\text{def}}{=} ww \cdots w$ for its $k$-fold concatenation. The *prefix* partial order $\preceq$ over words is defined by $u \preceq v$ if $v = uw$ for some word $w$. We write $u \prec v$ if $u$ is a proper prefix of $v$. A *language* is a subset $L \subseteq A^*$. A language $L$ is said to be *prefix-closed* if $u \preceq v$ and $v \in L$ implies $u \in L$.

*Trees.* A *tree* $T$ is a finite prefix-closed subset of $\mathbb{N}^*$ satisfying the property that if $tj$ is in $T$ then $ti$ in $T$ for all $i < j$. Elements of $T$ are called *nodes*. Its *root* is the empty word $\varepsilon$. An *ancestor* of a node $t$ is a prefix $s \preceq t$. A *child* of a node $t$ in $T$ is a node $tj$ in $T$ with $j$ in $\mathbb{N}$. A node is called a *leaf* if it has no child, and is said to be *internal* otherwise. The *size* of a tree $T$ is its cardinal $|T|$, its *height* is the maximal length $|t|$ for any of its nodes $t \in T$.

*Context-free Grammars.* A *context-free grammar* is a triple $G = (V, A, R)$, where $V$ and $A$ are disjoint finite sets of *nonterminal* and *terminal* symbols, and $R \subseteq V \times (V \cup A)^*$ is a finite set of *production rules*. The *degree* of $G$ is $\delta^G \stackrel{\text{def}}{=} \max\{|\alpha| \mid (X, \alpha) \in R\}$. We write

$$X \vdash \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_k$$

to denote that $(X, \alpha_1), \ldots, (X, \alpha_k) \in R$. For all words $w, w' \in (V \cup A)^*$, the grammar admits a *derivation step* $w \Longrightarrow w'$ if there exist two words $u, v$ in $(V \cup A)^*$ and a production rule $(X, \alpha)$ in $R$ such that $w = uXv$ and $w' = u\alpha v$. Let $\stackrel{*}{\Longrightarrow}$ denote the reflexive and transitive closure of $\Longrightarrow$. The *language* of a word $w$ in $(V \cup A)^*$ is the set $L_w^G \stackrel{\text{def}}{=} \{z \in A^* \mid w \stackrel{*}{\Longrightarrow} z\}$. A nonterminal $X$ is said to be *derivable* from a word $w \in (V \cup A)^*$ if there exists $u, v \in (V \cup A)^*$ such that $w \stackrel{*}{\Longrightarrow} uXv$. A nonterminal $X \in V$ is called *productive* if $L_X^G \neq \emptyset$.

*Parse Trees.* A *parse tree* for a context-free grammar $G = (V, A, R)$ is a tree $T$ equipped with a labeling function $sym : T \to (V \cup A \cup \{\varepsilon\})$ such that $R$ contains the production rule $sym(t) \vdash sym(t0) \cdots sym(tk)$ for every internal node $t$ with children $t0, \ldots, tk$. In addition, each leaf $t \neq \varepsilon$ with $sym(t) = \varepsilon$ is the only child of its parent. Notice that $sym(t) \in V$ for every internal node $t$. A parse tree is called *complete* when $sym(t) \in (A \cup \{\varepsilon\})$ for every leaf $t$. The *yield* of a parse tree $(T, sym)$ is the word $sym(t_1) \cdots sym(t_\ell)$ where $t_1, \ldots, t_\ell$ are the leaves of $T$ in lexicographic order (informally, from left to right). Observe that $S \stackrel{*}{\Longrightarrow} w$, where $S = sym(\varepsilon)$ is the label of the root and $w$ is the yield. Conversely, a parse tree with root labeled by $S$ and yield $w$ can be associated to any derivation $S \stackrel{*}{\Longrightarrow} w$.

## 3 Grammar-Controlled Vector Addition Systems

We first recall the main concepts of vector addition systems. Fix $k \in \mathbb{N}$. A $k$-dimensional *vector addition system* (shortly, $k$-*VAS*) is a finite set $\boldsymbol{A} \subseteq \mathbb{Z}^k$ of *actions*. Its operational semantics is given by the binary *step* relations $\xrightarrow{\boldsymbol{a}}$

over $\mathbb{N}^k$, where $\boldsymbol{a}$ ranges over $\boldsymbol{A}$, defined by $\boldsymbol{c} \xrightarrow{\boldsymbol{a}} \boldsymbol{d}$ if $\boldsymbol{d} = \boldsymbol{c} + \boldsymbol{a}$. The step relations are extended to words and languages as expected: $\xrightarrow{\varepsilon}$ is the identity, $\xrightarrow{z\boldsymbol{a}} \stackrel{\text{def}}{=} \xrightarrow{\boldsymbol{a}} \circ \xrightarrow{z}$ for $z \in \boldsymbol{A}^*$ and $\boldsymbol{a} \in \boldsymbol{A}$, and $\xrightarrow{L} \stackrel{\text{def}}{=} \bigcup_{z \in L} \xrightarrow{z}$ for $L \subseteq \boldsymbol{A}^*$. For every word $z = \boldsymbol{a}_1 \cdots \boldsymbol{a}_k$ in $\boldsymbol{A}^*$, we let $\sum z$ denote the sum $\boldsymbol{a}_1 + \cdots + \boldsymbol{a}_k$. Notice that $\boldsymbol{c} \xrightarrow{z} \boldsymbol{d}$ implies $\boldsymbol{d} - \boldsymbol{c} = \sum z$, for every $\boldsymbol{c}, \boldsymbol{d} \in \mathbb{N}^k$.

The *VAS reachability problem* asks, given a $k$-VAS $\boldsymbol{A}$ and vectors $\boldsymbol{c}, \boldsymbol{d} \in \mathbb{N}^k$, whether $\boldsymbol{c} \xrightarrow{\boldsymbol{A}^*} \boldsymbol{d}$. This problem is known to be ExpSpace-hard [14], but no upper bound has been established yet. The *VAS coverability problem* asks, given a $k$-VAS $\boldsymbol{A}$ and vectors $\boldsymbol{c}, \boldsymbol{d} \in \mathbb{N}^k$, whether $\boldsymbol{c} \xrightarrow{\boldsymbol{A}^*} \boldsymbol{d}'$ for some vector $\boldsymbol{d}' \geq \boldsymbol{d}$. This problem is known to be ExpSpace-complete [14,16].

**Definition 3.1 (GVAS).** *A $k$-dimensional* grammar-controlled vector addition system *(shortly, $k$-*GVAS*) is a context-free grammar $G = (V, \boldsymbol{A}, R)$ with $\boldsymbol{A} \subseteq \mathbb{Z}^k$.*

We give the semantics of GVAS by extending the binary step relations of VAS to words over $V \cup A$. Formally, for every word $w \in (V \cup A)^*$, we let $\xrightarrow{w} \stackrel{\text{def}}{=} \xrightarrow{L}$ where $L = L_w^G$ is the language of $w$. The *GVAS reachability problem* asks, given a $k$-GVAS $G = (V, \boldsymbol{A}, R)$, a nonterminal $S \in V$ and two vectors $\boldsymbol{c}, \boldsymbol{d} \in \mathbb{N}^k$, whether $\boldsymbol{c} \xrightarrow{S} \boldsymbol{d}$. The *GVAS coverability problem* asks, given the same input, whether $\boldsymbol{c} \xrightarrow{S} \boldsymbol{d}'$ for some vector $\boldsymbol{d}' \geq \boldsymbol{d}$. These problems can equivalently be rephrased in terms of VAS that have access to a pushdown stack, called *stack VAS* in [9] and *pushdown VAS* in [12]. Lazić [9] showed a Tower lower bound for these two problems, by simulating bounded Minsky machines. Their decidability remains open. As remarked in [9], GVAS reachability can be reduced to GVAS coverability. Indeed, a simple "budget" construction allows to reduce, in logarithmic space, the reachability problem for $k$-GVAS to the coverability problem for $(k+1)$-GVAS. This induces a hierarchy of decision problems, consisting of, alternatingly, coverability and reachability for growing dimension. The decidability of all these problems is open. This motivates the study of the most simple case: the coverability problem in dimension one, which is the focus of this paper. Our main contribution is the following result.

**Theorem 3.2.** *The coverability problem is decidable for 1-GVAS.*

For the remainder of the paper, we restrict our attention to the dimension one, and shortly write GVAS instead of 1-GVAS. Every GVAS can be effectively normalized, by removing non-productive nonterminals, replacing terminals $a \in \mathbb{Z}$ by words over the alphabet $\{-1, 0, 1\}$, and enforcing, through zero padding (since $\xrightarrow{0}$ is the identity relation), that $|\alpha| \geq 2$ for some production rule $X \vdash \alpha$. So in order to simplify our proofs, we consider w.l.o.g. only GVAS of this simpler form.

*Assumption.* We restrict our attention to GVAS $G = (V, A, R)$ where every $X \in V$ is productive, where $A = \{-1, 0, 1\}$, and of degree $\delta^G \geq 2$.

We associate to a GVAS $G$ and a word $w \in (V \cup A)^*$ the *displacement* $\Delta_w^G \in \overline{\mathbb{Z}}$ and the *summary* function $\sigma_w^G : \overline{\mathbb{N}} \to \overline{\mathbb{N}}$ defined by

$$\Delta_w^G \stackrel{\text{def}}{=} \sup\{\textstyle\sum z \mid z \in L_w^G\} \qquad\qquad \sigma_w^G(n) \stackrel{\text{def}}{=} \sup\{d \mid \exists c \leq n : c \xrightarrow{w} d\}$$

Informally, $\Delta_w^G$ is the "best shift" achievable by a word in $L_w^G$, and $\sigma_w^G(n)$ gives the "largest" number that is reachable via some word in $L_w^G$ starting from $n$ or below. When no such number exists, $\sigma_w^G(n)$ is $-\infty$ (recall that $\sup \emptyset = -\infty$). Since all nonterminals are productive, the language $L_w^G$ is not empty. Therefore, $\Delta_w^G > -\infty$ and $\sigma_w^G(n) > -\infty$ for some $n \in \mathbb{N}$.

*Remark 3.3 (Monotonicity).* For every $w \in (V \cup A)^*$ and $c, d, e \in \mathbb{N}$, $c \xrightarrow{w} d$ implies $c + e \xrightarrow{w} d + e$. Consequently, $\sigma_w^G(n + e) \geq \sigma_w^G(n) + e$ holds for every $w \in (V \cup A)^*$, $n \in \overline{\mathbb{N}}$ and $e \in \mathbb{N}$.

A straightforward application of Parikh's theorem shows that $\Delta_w^G$ is effectively computable from $G$ and $w$. We will provide in Section 6 an effective characterization of $\sigma_w^G$ when the displacement $\Delta_w^G$ is finite. In order to characterize functions $\sigma_w^G$ where the displacement $\Delta_w^G$ is infinite, it will be useful to consider the *ratio* of $w$, defined as

$$\lambda_w^G \stackrel{\text{def}}{=} \liminf_{n \to +\infty} \frac{\sigma_w^G(n)}{n}$$

Notice that $\lambda_w^G \geq 1$. This fact follows from Remark 3.3 and the observation that $\sigma_w^G(n) > -\infty$ for some $n \in \mathbb{N}$. From now on, we just write $L_w$, $\delta$, $\Delta_w$, $\sigma_w$ and $\lambda_w$ when $G$ is clear from the context.

*Example 3.4.* Multiplication by 2 can be expressed as a summary function using the GVAS with production rules $S \vdash -1\, S\, 1\, 1 \mid \varepsilon$. Indeed, for every $c$,

$$\begin{aligned} c \xrightarrow{S} d \quad &\iff \quad \exists n \in \mathbb{N} : c \xrightarrow{(-1)^n (11)^n} d \\ &\iff \quad \exists n \leq c : c \xrightarrow{(-1)^n} c - n \xrightarrow{(11)^n} c + n = d \quad \iff \quad c \leq d \leq 2c \end{aligned}$$

Therefore, $\sigma_S(n) = 2n$ for every $n \in \mathbb{N}$. Observe that $\Delta_S = +\infty$ and $\lambda_S = 2$. □

*Example 3.5.* The Ackermann functions $A_m : \mathbb{N} \to \mathbb{N}$, for $m \in \mathbb{N}$, are defined by induction for every $n \in \mathbb{N}$ by:

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

These functions are expressible as summary functions for the GVAS with nonterminals $X_0, \ldots, X_m$ and with production rules $X_0 \vdash 1$ and $X_i \vdash -1\, X_i\, X_{i-1} \mid 1 X_{i-1}$ for $1 \leq i \leq m$. It is routinely checked that $\sigma_{X_m}(n) = A_m(n)$ for every $n \in \mathbb{N}$. Notice also that $\lambda_{X_0} = 1$, $\lambda_{X_1} = 2$, and $\lambda_{X_m} = +\infty$ for every $m \geq 2$. □

**Lemma 3.6.** *For every two words $u, v \in (V \cup A)^*$, the following properties hold:*

1. *$\Delta_{uv} = \Delta_u + \Delta_v$ and $\sigma_{uv} = \sigma_v \circ \sigma_u$.*
2. *If $u \xRightarrow{*} v$ then $\Delta_u \geq \Delta_v$, $\lambda_u \geq \lambda_v$, and $\sigma_u(n) \geq \sigma_v(n)$ for all $n \in \overline{\mathbb{N}}$.*

An equivalent formulation of the coverability problem is the question whether $\sigma_S(c) \geq d$ holds, given a nonterminal $S \in V$ and two numbers $c, d \in \mathbb{N}$. We solve this problem by exhibiting small certificates for $\sigma_S(c) \geq d$, that take the form of (suitably truncated) annotated parse trees.

## 4 Small Coverability Certificates

To solve the coverability problem, we annotate parse trees in a way that is consistent with the summary functions. A *flow tree* for a GVAS $G$ is a parse tree $(T, \mathit{sym})$ for $G$ equipped with two functions $\mathit{in}, \mathit{out} : T \to \mathbb{N}$, assigning an *input* and an *output* value to each node, and satisfying, for every node $t \in T$, the following *flow conditions*:

1. If $t$ is internal with children $t0, \ldots, tk$, then $\mathit{in}(t0) \leq \mathit{in}(t)$, $\mathit{out}(t) \leq \mathit{out}(tk)$, and $\mathit{in}(t(j+1)) \leq \mathit{out}(tj)$ for every $j = 0, \ldots, k-1$.
2. If $t$ is a leaf then $\mathit{out}(t) \leq \sigma_{\mathit{sym}(t)}(\mathit{in}(t))$.

We shortly write $t : c\#d$ to mean that $(\mathit{in}(t), \mathit{sym}(t), \mathit{out}(t)) = (c, \#, d)$. A flow tree is called *complete* when the underlying parse tree is complete, i.e., when $\mathit{sym}(t) \in (A \cup \{\varepsilon\})$ for every leaf $t$. The following lemmas state useful properties of flow trees that can be shown using the flow conditions and the monotonicity of summary functions (see Remark 3.3). A consequence is that $\sigma_S(c) \geq d$ holds if, and only if, there exists a complete flow tree with root $\varepsilon : cSd$.

**Lemma 4.1.** *It holds that $\sigma_\#(c) \geq d$ for every node $t : c\#d$ of a flow tree.*

**Lemma 4.2.** *Let $S \in V$ and $c, d \in \mathbb{N}$. If $\sigma_S(c) \geq d$ then there exists a complete flow tree with root $\varepsilon : bSe$ such that $b \leq c$ and $e \geq d$.*

We will need to compare flow trees. Let the *rank* of a flow tree $(T, \mathit{sym}, \mathit{in}, \mathit{out})$ be the pair $(|T|, \sum_{t \in T} \mathit{in}(t) + \mathit{out}(t))$. The lexicographic order $\preceq_{\text{lex}}$ over $\mathbb{N}^2$ is used to compare ranks of flow trees. A complete flow tree $(T, \mathit{sym}, \mathit{in}, \mathit{out})$ is called *optimal* if there exists no complete flow tree $(T', \mathit{sym}', \mathit{in}', \mathit{out}')$ of strictly smaller rank such that $\mathit{in}'(\varepsilon) \leq \mathit{in}(\varepsilon)$, $\mathit{sym}(\varepsilon) = \mathit{sym}(\varepsilon)$, and $\mathit{out}'(\varepsilon) \geq \mathit{out}(\varepsilon)$. Optimal flow trees enjoy the following important properties, stated formally below. Firstly, they are *tight*, meaning that the inequalities in the first flow condition are in fact equalities. Secondly, they are *balanced*, meaning that the input value of each node is never too large compared to its output value.

**Lemma 4.3.** *For every internal node $t$ in an optimal complete flow tree, we have $\mathit{in}(t0) = \mathit{in}(t)$, $\mathit{in}(t1) = \mathit{out}(t0)$, …, $\mathit{in}(tk) = \mathit{out}(t(k{-}1))$, and $\mathit{out}(t) = \mathit{out}(tk)$, where $t0, \ldots, tk$ are the children of $t$.*

**Lemma 4.4.** *For every node $t$ in an optimal complete flow tree, it holds that $\mathit{in}(t) \leq \mathit{out}(t) + \delta^{|V|}$.*

Next, we show how to truncate flow trees while preserving enough information to decide that the *in* and *out* labelings satisfy the flow conditions. Our truncation is justified by the following lemma.

**Lemma 4.5.** *Let $X \in V$ and $n \in \mathbb{N}$. If $\lambda_X = +\infty$ and there is a derivation $X \xRightarrow{*} uXv$ such that $\sigma_u(n) > n$, then it holds that $\sigma_X(n) = +\infty$.*

**Definition 4.6 (Certificates).** *A* certificate *is a flow tree* $(T, sym, in, out)$ *in which every leaf* $t$ *with* $\lambda_{sym(t)} = +\infty$ *has a proper ancestor* $s \prec t$ *such that* $sym(s) = sym(t)$ *and* $in(s) < in(t)$*.*

Notice that every complete flow tree is a certificate. We now prove the existence of small certificates. Let $S \in V$ and $c, d \in \mathbb{N}$ such that $\sigma_S(c) \geq d$. We introduce the set $\mathcal{T}$ of all complete flow trees with root $\varepsilon : bSe$ satisfying $b \leq c$ and $e \geq d$. By Lemma 4.2, the set $\mathcal{T}$ is not empty. Let us pick $(T, sym, in, out)$ in $\mathcal{T}$ among those of least rank. By definition, the root $\varepsilon$ of $T$ satisfies $in(\varepsilon) \leq c$ and $out(\varepsilon) = d$. Notice that the complete flow tree $T$ is optimal. Let us introduce the set $U$ of all nodes $t \in T$ such that every proper ancestor $s \prec t$ satisfies the following condition:

$$\text{For every ancestor } r \preceq s,\ sym(r) = sym(s) \implies in(r) \geq in(s) \tag{1}$$

By definition, the set $U$ is a nonempty and prefix-closed subset of $T$. The following fact derives from Lemma 4.1 and the property that $T$ is a complete flow tree.

*Fact 4.7.* The tree $U$, equipped with the restrictions to $U$ of the functions *sym*, *in* and *out*, is a certificate.

Our next step is to bound the height of $U$ as well as the input and output values of its nodes. We will use the following properties, that are easily derived from the definition of $U$, the optimality of $T$, and Lemmas 4.3 and 4.4.

*Fact 4.8.* Let $r$ and $s$ be nodes in $U$ such that $r \prec s$.

1. If $s$ is internal in $U$ and $sym(r) = sym(s)$ then $out(s) < out(r)$, and
2. If $s$ is a child of $r$ then $out(s) \leq out(r) + (\delta - 1)\delta^{|V|}$.

Consider a leaf $t$ in $U$. For each $i$ in $\{0, \ldots, |t|\}$, let $t_i$ denote the unique prefix $t_i \preceq t$ with length $|t_i| = i$, and let $(\#_i, d_i) = (sym(t_i), out(t_i))$. Note that $d_0 = out(\varepsilon) = d$. Fact 4.8 entails that for every $i, j$ with $0 \leq i, j < |t|$,

$$d_{i+1} \leq d_i + \delta^{|V|+1} \qquad \text{and} \qquad (i < j \wedge \#_i = \#_j) \implies d_i > d_j \tag{2}$$

Let $m_i = \max\{d_0, \ldots, d_i\}$ for all $i \in \{0, \ldots, |t|\}$. According to Equation (2), increasing pairs $m_i < m_{i+1}$ may occur in the sequence $m_0, \ldots, m_{|t|}$ only when $\#_{i+1} \notin \{\#_0, \ldots, \#_i\}$ or $i + 1 = |t|$. So there are at most $|V|$ such increasing pairs. Moreover, for each increasing pair $m_i < m_{i+1}$, the increase $m_{i+1} - m_i$ is bounded by $\delta^{|V|+1}$. We derive that $d_i \leq m_{|t|} \leq d + |V| \cdot \delta^{|V|+1} < d + \delta^{2|V|+1}$ for all $i$ with $0 \leq i \leq |t|$, since $\delta \geq 2$ by assumption. It follows from Equation (2) that each nonterminal in $V$ appears at most $d + \delta^{2|V|+1}$ times in the sequence $(\#_i)_{0 \leq i < |t|}$. By the pigeonhole principle, we get that $|t| \leq |V| \cdot (d + \delta^{2|V|+1})$. We have thus shown that for every node $t \in U$,

$$|t| \leq d \cdot |V| + \delta^{3|V|+1} \qquad \text{and} \qquad in(t) + out(t) \leq 2d + \delta^{2|V|+3} \tag{3}$$

This concludes the proof of the "only if" direction of the following proposition. The "if" direction follows from Lemma 4.1, since every certificate is a flow tree.

**Proposition 4.9.** *For every $S \in V$ and $c, d \in \mathbb{N}$, it holds that $\sigma_S(c) \geq d$ if, and only if, there exists a certificate with root $\varepsilon : bSd$ for some $b \leq c$ and whose nodes $t$ satisfy Equation (3).*

The above proposition leads to a simple procedure to solve the coverability problem, as we only need to enumerate finitely many potential certificates. Checking whether an annotated parse tree is a certificate reduces to (a) the question whether a given nonterminal $X$ has an infinite ratio, and (b) the coverability question $\sigma_X(c) \geq d$ for nonterminals $X$ with finite ratio. Both questions will be shown to be decidable in Section 6 by reduction to the subclass of thin GVAS, which is the focus of the next section.

## 5 Semilinearity of the Step Relations for Thin GVAS

We turn to reachability relations in a particular subclass of GVAS called *thin*. A context-free grammar is said to be *thin*[4] if $\alpha \in A^*VA^*$ for every production rule $X \vdash \alpha$ such that $X$ is derivable from $\alpha$. Recall that *Presbuger arithmetic* is the first-order theory of the natural numbers with addition. It is well-known that *semilinear sets* coincide with the sets definable in Presburger arithmetic [7].

**Theorem 5.1.** *For every nonterminal symbol $S$ of a thin GVAS, the relation $\xrightarrow{S}$ is effectively definable in Presburger arithmethic.*

Our argument goes by a reduction to the reachability problem for 2-dimensional vector addition systems, and uses the following result.

**Theorem 5.2 ([13]).** *Let $\boldsymbol{A}$ be a 2-VAS and $\Pi \subseteq \boldsymbol{A}^*$ be a regular language over its actions. The relation $\xrightarrow{\Pi}$ is effectively definable in the Presburger arithmetic.*

Let us call a GVAS $G = (V, A, R)$ *simple* if for every production rule $X \vdash \alpha$, either $X$ is not derivable from $\alpha$, or $\alpha \in AVA$. Clearly, every simple GVAS is thin. Conversely, every thin GVAS can be transformed into an equivalent simple GVAS by replacing production rules in $V \times A^*VA^*$ by finitely many new rules in $V \times AVA$. See Lemma D.1 in Appendix D for details. Consequently, it suffices to show the claim of Theorem 5.1 for simple GVAS only.

We show by induction on $|V|$ that $\xrightarrow{S}$ is effectively definable in Presburger arithmethic for every simple thin GVAS $G = (V, A, R)$, and for every nonterminal $S \in V$. Naturally, if $|V|$ is empty the proof is immediate. Assume the induction is proved for a number $h \in \mathbb{N}$, and let us consider a simple thin GVAS $G = (V, A, R)$ with $|V| = h + 1$, and a nonterminal $S \in V$.

Notice that $\boldsymbol{A} \overset{\text{def}}{=} \{-1, 0, 1\}^2$ is a vector addition system. We consider the finite, directed graph with set of nodes $V$ that contains an $(a, -b)$-labeled edge

[4] Thinness entails that for any derivation $S \overset{*}{\Longrightarrow} w$, the number of nonterminals in $w$ is bounded by $\delta^{|V|}$. This entails that parse trees of thin GVAS are of bounded width. Thin GVAS are thus a subclass of the *finite-index* grammars of [1].

from $X$ to $Y$ for every production rule $X \vdash aYb$ in $R$. To each nonterminal $X \in V$, we assotiate the regular language $\Pi_X$ of words recognized by this finite graph starting from $S$ and reaching $X$. By Theorem 5.2, $\xrightarrow{\Pi_X}$, the regular restriction of the reachability set of $\boldsymbol{A}$, is effectively definable in Presburger arithmetic.

As a next ingredient, let $\Gamma_X$ be the finite set of words $\alpha \in (V \cup A)^*$ such that $X \vdash \alpha$ is a production rule and $X$ is not derivable from $\alpha$. We observe that $L_\alpha^G$ is equal to the language of $\alpha$ in the simple grammar $G'$, obtained from $G$ by removing the nonterminal $X$ and all production rules where $X$ occurs. By induction, and since $\xrightarrow{a}$ are trivially Presburger-definable for terminals $a \in A$, we deduce that $\xrightarrow{\alpha}$ is effectively Presburger-definable as a compositon of Presburger relations. Because $\Gamma_X$ is finite, we deduce that $\xrightarrow{\Gamma_X} = \bigcup_{\alpha \in \Gamma_X} \xrightarrow{\alpha}$, is definable in the Presburger arithmetic as a finite disjunction of Presburger relations.

This following Lemma 5.3 concludes Theorem 5.1.

**Lemma 5.3.** *For for all $c, d \in \mathbb{N}$, $c \xrightarrow{S} d$ if, and only if, the following relation holds:*

$$\phi_S(c,d) \stackrel{def}{=} \bigvee_{X \in V} \exists c', d' \in \mathbb{N} \quad (c,d) \xrightarrow{\Pi_X} (c',d') \wedge c' \xrightarrow{\Gamma_X} d' \tag{4}$$

*Proof.* Assume that $c \xrightarrow{S} d$. It means that there exists $w \in L_S$ such that $c \xrightarrow{w} d$. Since $w \in A^*$, we deduce that a sequence of derivation steps from $S$ that produces $w$ must necessarily derive at some point a nonterminal symbol $X$ with a production rule $X \vdash \alpha$ such that $\alpha \in A^*$, and in particular $\alpha \in \Gamma_X$. By considering the first time a derivation step $X \xRightarrow{\alpha}$ with $\alpha \in \Gamma_X$ occurs, we deduce a sequence $X_0, \dots, X_k$ of nonterminal symbols with $X_0 = S$, a sequence $r_1, \dots, r_k$ of production rules $r_j \in R$ of the form $X_{j-1} \vdash a_j X_j b_j$ with $a_j, b_j \in A$, a production rule $r_{k+1} \in R$ of the form $X_k \vdash \alpha$ where $\alpha \in \Gamma_{X_k}$, and a word $w' \in L_\alpha$ such that $w = a_1 \dots a_k w' b_k \dots b_1$. Since $c \xrightarrow{w} d$, it follows that there exist $c', d' \in \mathbb{N}$ such that $c \xrightarrow{a_1 \dots a_k} c' \xrightarrow{w'} d' \xrightarrow{b_k \dots b_1} d$. Thus $(c,d) \xrightarrow{\pi} (c',d')$ with $\pi \stackrel{\text{def}}{=} (a_1, -b_1) \dots (a_k, -b_k)$. It follows that $\phi_S(c,d)$ holds. Conversely, if $\phi_S(c,d)$ holds, by reversing the previous proof steps, if follows that $c \xrightarrow{S} d$. A detailed proof is given in Appendix D. □

## 6 Computation of Summaries for Bounded Ratios

In this section, we show that the summary function $\sigma_X$ is effectively computable when the ratio $\lambda_X$ is finite. In addition, the question whether $\lambda_X$ is finite is shown to be decidable. These results are ultimately obtained by reduction to the thin GVAS case. We first consider nonterminals with finite displacements.

The next lemma follows from the observation that if the maximal displacement of a nonterminal is finite, then it can already be achieved by a short word.

**Lemma 6.1.** *Let $S \in V$ be a nonterminal with $\Delta_S < +\infty$. Then it holds that $\sigma_S(n) = n + \Delta_S$ for every $n \in \overline{\mathbb{N}}$ such that $n \geq \delta^{|V|}$.*

**Proposition 6.2.** *For every nonterminal $S \in V$ with $\Delta_S < +\infty$, the function $\sigma_S$ is effectively computable.*

The following lemma will be useful in our reduction below.

**Lemma 6.3.** *Let $X \in V$ be a nonterminal. If there is a derivation $X \xRightarrow{*} uXv$ such that $\Delta_{uv} = +\infty$ then it holds that $\lambda_X = +\infty$.*

We will now show that summaries are computable for nonterminals with finite ratio. The main idea is to transform the given GVAS into an equivalent *thin* GVAS, by hard-coding the effect of nonterminals with finite displacement. This is effective due to Proposition 6.2. Computability of $\lambda_X$ and $\sigma_X$ then follows from Theorem 5.1. The following ad-hoc notion of equivalence is sufficient for this purpose. Crucially, it has no requirement for nonterminals with infinite ratio.

Two GVAS $G = (V, A, R)$ and $G' = (V', A', R')$ are called *equivalent* if firstly $V = V'$, secondly $\lambda_X^G = \lambda_X^{G'}$ for every nonterminal $X$, and thirdly $\sigma_X^G = \sigma_X^{G'}$ for every nonterminal $X$ *with finite ratio.*

*Unfoldings.* For our first transformation, assume a nonterminal $X \in V$ with $\Delta_X^G < +\infty$. The *unfolding of* $X$ is the GVAS $H = (V, A, R')$ where $R'$ is obtained from $R$ by removing all production rules $X \vdash \alpha$ and instead adding, for every $0 \leq i \leq \delta^{|V|}$ with $j = \sigma_X^G(i) > -\infty$, a rule $X \vdash (-1)^i(1)^j$.

Observe that the language $L_X^H$ is finite, and that $H$ can be computed from $G$ and $X$ because $\sigma_X^G$ is computable by Proposition 6.2.

*Fact 6.4.* The unfolding of $X$ is equivalent to $G$.

*Expansions.* Our second transformation completely inlines a given nonterminal with finite language. Given a nonterminal $Y \in V$ with $L_Y^G$ finite, the *expansion of* $Y$ is the GVAS $H = (V, A, R')$ where $R'$ is obtained from $R$ by replacing each production rule $X \vdash \alpha_0 Y \alpha_1 \cdots Y \alpha_k$, with $Y$ not occurring in $\alpha_0 \cdots \alpha_k$, by the rules $X \vdash \alpha_0 z_1 \alpha_1 \cdots z_k \alpha_k$ where $z_1, \ldots, z_k \in L_Y^G$. Note that $H$ can be computed from $G$ and $Y$. Obviously, languages are preserved by this transformation, i.e., $L_w^G = L_w^H$ for every $w$ in $(V \cup A)^*$. The following fact follows.

*Fact 6.5.* The expansion of $Y$ is equivalent to $G$.

*Abstractions.* Our last transformation simplifies a given nonterminal with infinite ratio, in such a way that its ratio remains infinite. Given a nonterminal $X \in V$ with $\lambda_X^G = +\infty$, the *abstraction of* $X$ is the GVAS $H = (V, A \cup \{1\}, R')$ where $R'$ is obtained from $R$ by removing all production rules $X \vdash \alpha$ and replacing them by the two rules $X \vdash 1X \mid \varepsilon$. Note that $H$ can be computed from $G$ and $X$.

*Fact 6.6.* The abstraction of $X$ is equivalent to $G$.

We now show how to effectively transform a GVAS into an equivalent thin GVAS. As a first step, we hard-code the effect of nonterminals with finite displacement into the production rules, using unfoldings and expansions described above. By Facts 6.4 and 6.5, this results in an equivalent GVAS. Moreover, it now holds that every nonterminal $Y$ occurring on the right handside $\alpha$ of some production rule $X \vdash \alpha$ has $\Delta_Y = +\infty$. Let $(V, A, R)$ be the constructed GVAS and assume that it is not already thin. This means that there exists a production rule $X \vdash \alpha$ with $\alpha \notin A^*VA^*$ such that $X$ is derivable from $\alpha$. So $X \overset{*}{\Longrightarrow} uXv$ for some words $u, v$ in $(V \cup A)^*$ such that $uv$ contains some nonterminal $Y$. As $Y$ occurs on the right handside of the initial production rule, it must have an infinite displacement. From Lemma 3.6 we thus get that also $\Delta_{uv} = +\infty$, and Lemma 6.3 lets us conclude that $\lambda_X = +\infty$. Therefore, by Fact 6.6, we may replace $G$ by the abstraction of $X$. Observe that this strictly decreases the number of production rules violating the condition for the system to be thin and at the same time it preserves the property that $\Delta_Y = +\infty$ for every $Y \in V$ occurring in the right handside a production rule. By iterating this abstraction process, we obtain a thin GVAS that is equivalent to the GVAS that we started with. We have thus shown the following proposition. Its corollary follows from Theorem 5.1, and states the missing ingredients for the proof of the coverability problem.

**Proposition 6.7.** *For every GVAS $G$, there exists an effectively constructable thin GVAS that is equivalent to $G$.*

**Corollary 6.8.** *The question whether $\lambda_X < +\infty$ holds for a given GVAS $G$ and a given nonterminal $X$, is decidable. Moreover, if $\lambda_X < +\infty$ then the function $\sigma_X$ is effectively computable.*

*Proof (of Theorem 3.2).* Thanks to Proposition 4.9, it suffices to check finitely many candidate certificates, each consisting of a parse tree $(T, \mathit{sym})$ of bounded height and labeling functions $\mathit{in}, \mathit{out} : T \to \mathbb{N}$ with bounded values. It remains to show that it is possible to verify that a given candidate is in fact a certificate. For this, it needs to satisfy the two flow conditions from page 6 and moreover, every leaf $t$ with $\lambda_{\mathit{sym}(t)} = +\infty$ must have some ancestor $s \prec t$ with $\mathit{sym}(s) = \mathit{sym}(t)$ and $\mathit{in}(s) < \mathit{in}(t)$.

The first flow condition can easily be verified locally. By Corollary 6.8, it is possible to check if $\lambda_{\mathit{sym}(t)} < +\infty$ for every leaf $t$ and therefore verify the third condition. In order to verify the second flow condition, it suffices to check that $\sigma_{\mathit{sym}(t)}(\mathit{in}(t)) \geq \mathit{out}(t)$ holds for all leaves with finite ratio $\lambda_{\mathit{sym}(t)} < +\infty$. This is effective due to Corollary 6.8. Indeed, if none of the above checks fail then it follows from Lemma 4.5 that $\sigma_{\mathit{sym}(t)}(\mathit{in}(t)) \geq \mathit{out}(t)$ necessarily holds also for the remaining leaves $t$ with $\lambda_{\mathit{sym}(t)} = +\infty$ (see Lemma E.3 in Appendix E for details). This means that the candidate satisfies the second flow condition and therefore all requirements for a certificate. □

## 7 Conclusion

The decidability of the coverability problem for pushdown VAS is a long-standing open question with applications for program verification. In this paper, we proved that coverability is decidable for 1-dimensional pushdown VAS. We reformulated the problem to the equivalent coverability problem for 1-dimensional grammar-controlled vector addition systems, and analyzed their behaviour in terms of structural properties of derivation trees.

An NP lower complexity bound can be shown by reduction from the Subset Sum problem. A closer inspection of our approach allows to derive an ExpSpace upper bound, using recent results by Blondin et al. [2] on 2-dimensional VAS reachability. The exact complexity is open, and so is the decidability of the problem for larger dimensions.

## References

1. Atig, M.F., Ganty, P.: Approximating Petri net reachability along context-free traces. In: FSTTCS. pp. 152–163 (2011)
2. Blondin, M., Finkel, A., Göller, S., Haase, C., McKenzie, P.: Reachability in two-dimensional vector addition systems with states is PSPACE-complete. In: LICS (2015), to appear
3. Bouajjani, A., Habermehl, P., Mayr, R.: Automatic verification of recursive procedures with one integer parameter. TCS 295, 85–106 (2003)
4. Courtois, J., Schmitz, S.: Alternating vector addition systems with states. In: MFCS. pp. 220–231 (2014)
5. Demri, S., Jurdzinski, M., Lachish, O., Lazic, R.: The covering and boundedness problems for branching vector addition systems. JCSS 79(1), 23–38 (2013)
6. Ganty, P., Majumdar, R.: Algorithmic verification of asynchronous programs. ACM Trans. Progr. Lang. Syst. 34(1), 6:1–6:48 (2012)
7. Ginsburg, S., Spanier, E.H.: Semigroups, Presburger formulas and languages. Pacific J. Math. 16(2), 285–296 (1966)
8. Kosaraju, S.R.: Decidability of reachability in vector addition systems (preliminary version). In: STOC. pp. 267–281 (1982)
9. Lazic, R.: The reachability problem for vector addition systems with a stack is not elementary. CoRR abs/1310.1767 (2013)
10. Lazic, R., Schmitz, S.: Non-elementary complexities for branching VASS, MELL, and extensions. In: CSL/LICS (2014)
11. Leroux, J.: Vector addition system reachability problem: a short self-contained proof. In: POPL. pp. 307–316 (2011)
12. Leroux, J., Praveen, M., Sutre, G.: Hyper-ackermannian bounds for pushdown vector addition systems. In: CSL/LICS (2014)
13. Leroux, J., Sutre, G.: On flatness for 2-dimensional vector addition systems with states. In: CONCUR. pp. 402–416 (2004)
14. Lipton, R.J.: The reachability problem requires exponential space. Tech. Rep. 63, Yale University (Jan 1976)
15. Mayr, E.W.: An algorithm for the general Petri net reachability problem. In: STOC. pp. 238–246 (1981)
16. Rackoff, C.: The covering and boundedness problems for vector addition systems. TCS 6(2), 223–231 (1978)
17. Reinhardt, K.: Reachability in Petri nets with inhibitor arcs. ENTCS 223(0), 239–264 (2008)

## A Elementary Parse Trees

Let $G = (V, A, R)$ be a context-free grammar. A parse tree $(T, sym)$ for $G$ is called *elementary*, if it contains no two nodes $s \prec t$ with $sym(s) = sym(t)$. A flow tree (see Section 4) shall be called *elementary* when the underlying parse tree is elementary.

*Remark A.1.* If the degree $\delta$ of $G$ is nonzero, then every elementary parse tree has at most $\delta^{|V|}$ leaves.

## B Proofs for Section 3

**Lemma 3.6.** *For every two words $u, v \in (V \cup A)^*$, the following properties hold:*

1. $\Delta_{uv} = \Delta_u + \Delta_v$ *and* $\sigma_{uv} = \sigma_v \circ \sigma_u$.
2. *If* $u \xRightarrow{*} v$ *then* $\Delta_u \geq \Delta_v$, $\lambda_u \geq \lambda_v$, *and* $\sigma_u(n) \geq \sigma_v(n)$ *for all* $n \in \overline{\mathbb{N}}$.

*Proof.* Let $u, v \in (V \cup A)^*$. For the proof of part 1), recall that $L_u$ and $L_v$ are non-empty, since all nonterminals are productive. We derive from the definition of the displacement that:

$$\begin{aligned} \Delta_u + \Delta_v &= \sup\{\textstyle\sum z \mid z \in L_u\} + \sup\{\textstyle\sum z \mid z \in L_v\} \\ &= \sup\{\textstyle\sum z_u + \sum z_v \mid z_u \in L_u \wedge z_v \in L_v\} \\ &= \sup\{\textstyle\sum z_u z_v \mid z_u \in L_u \wedge z_v \in L_v\} \\ &= \sup\{\textstyle\sum z \mid z \in L_{uv}\} && [L_{uv} = L_u L_v] \\ &= \Delta_{uv} \end{aligned}$$

Let $n \in \overline{\mathbb{N}}$ and let us show that $\sigma_{uv}(n) = \sigma_v \circ \sigma_u(n)$. Assume that $c \xrightarrow{uv} d$ with $c \leq n$. There exists $c'$ such that $c \xrightarrow{u} c' \xrightarrow{v} d$. Observe that $c' \leq \sigma_u(n)$. It follows from the definition of $\sigma_v$ that $d \leq \sigma_v(\sigma_u(n))$. We have shown that $\sigma_{uv}(n) \leq \sigma_v \circ \sigma_u(n)$. Conversely, suppose that $c' \xrightarrow{v} d$ with $c' \leq \sigma_u(n)$. By definition of $\sigma_u(n)$, there exists $c \leq n$ and $d' \geq c'$ such that $c \xrightarrow{u} d'$. We get that $c \xrightarrow{u} d' \xrightarrow{v} d''$ for some $d'' \geq d$. Observe that $d'' \leq \sigma_{uv}(n)$. It follows that $d \leq \sigma_{uv}(n)$. We have shown that $\sigma_v \circ \sigma_u(n) \leq \sigma_{uv}(n)$.

We now prove point 2. Assume that $u \xRightarrow{*} v$, and let $n \in \overline{\mathbb{N}}$. Observe that $L_u \supseteq L_v$. Therefore, it holds that $\{\sum z \mid z \in L_u\} \supseteq \{\sum z \mid z \in L_v\}$ and that $\{d \mid \exists c \leq n : c \xrightarrow{u} d\} \supseteq \{d \mid \exists c \leq n : c \xrightarrow{v} d\}$. The first inclusion entails that $\Delta_u \geq \Delta_v$, and the second inclusion entails that $\sigma_u(n) \geq \sigma_v(n)$. The last assertion, namely $\lambda_u \geq \lambda_v$, follows from the fact that $\sigma_u(n) \geq \sigma_v(n)$ for all $n \in \mathbb{N}$. ◻

## C Proofs for Section 4

**Lemma 4.1.** *It holds that $\sigma_{\#}(c) \geq d$ for every node $t : c\#d$ of a flow tree.*

*Proof.* Let $(T, sym, in, out)$ be a flow tree. We prove the claim by structural induction on $T$. For leaf nodes $t$, the claim holds by the second flow requirement. For internal nodes $t : cXd$, assume that the claim holds for the children $t0, \ldots, tk$ of $t$. Suppose that $tj : c_j \#_j d_j$ for all $j$ with $0 \leq j \leq k$. Since $X \Longrightarrow \#_0 \cdots \#_k$, Lemma 3.6 implies that $\sigma_X(n) \geq \sigma_{\#_k} \circ \cdots \circ \sigma_{\#_0}(n)$ for all $n \in \mathbb{N}$. By the first flow requirement, it holds that $c_0 \leq c$, $c_1 \leq d_0, \ldots, c_k \leq d_{k-1}$, and $d \leq d_k$. We derive from the monotonicity of summary functions (see Remark 3.3) that

$$\begin{aligned}
\sigma_X(c) &\geq \sigma_{\#_k} \circ \cdots \circ \sigma_{\#_0}(c_0) && [c \geq c_0] \\
&\geq \sigma_{\#_k} \circ \cdots \circ \sigma_{\#_1}(c_1) && [\sigma_{\#_0}(c_0) \geq d_0 \geq c_1] \\
&\geq \sigma_{\#_k}(c_k) && [\sigma_{\#_j}(c_j) \geq d_j \geq c_{j+1}] \\
&\geq d && [\sigma_{\#_k}(c_k) \geq d_k \geq d]
\end{aligned}$$

By induction, we conclude that the lemma holds for every node of $T$. ☐

**Lemma 4.2.** *Let $S \in V$ and $c, d \in \mathbb{N}$. If $\sigma_S(c) \geq d$ then there exists a complete flow tree with root $\varepsilon : bSe$ such that $b \leq c$ and $e \geq d$.*

*Proof.* Assume that $\sigma_S(c) \geq d$. This means that there exists $e \geq d$ such that $c \xrightarrow{S} e$, which in turn means that there exists $w \in L_S$ such that $c \xrightarrow{w} e$. Since $w \in L_S$, there exists a derivation $S \overset{*}{\Longrightarrow} w$, hence, a complete parse tree with root labeled by $S$ and yield $w$. This parse tree, together with the fact that $c \xrightarrow{w} e$, induces a complete flow tree with root $\varepsilon : cSe$. ☐

**Lemma 4.3.** *For every internal node $t$ in an optimal complete flow tree, we have $in(t0) = in(t)$, $in(t1) = out(t0)$, $\ldots$, $in(tk) = out(t(k-1))$, and $out(t) = out(tk)$, where $t0, \ldots, tk$ are the children of $t$.*

*Proof.* The first flow condition requires $in(t0) \leq in(t)$, $in(t1) \leq out(t0)$, $\ldots$, $in(tk) \leq out(t(k-1))$, and $out(t) \leq out(tk)$, for every internal node $t$ with children $t0, \ldots, tk$. For the converse inequalities, assume that $in(t0) < in(t)$ (the other cases are analogous). Then, changing the labeling of the node $t$ using $in(t) := in(t0)$ provides a complete flow tree of strictly smaller rank, contrary to the optimality of $T$. ☐

**Lemma 4.4.** *For every node $t$ in an optimal complete flow tree, it holds that $in(t) \leq out(t) + \delta^{|V|}$.*

*Proof.* Let $(T, sym, in, out)$ be an optimal complete flow tree. We only prove the lemma for the root $\varepsilon : c\#d$, since every subtree of an optimal complete flow tree is also an optimal complete flow tree. Let $t_1, \ldots, t_\ell$, with $t_i : c_i a_i d_i$, denote the leaves of $T$ in lexicographic order (informally, from left to right).

We first show that $c - d \leq \ell$. Note that $a_1, \ldots, a_\ell$ are in $(A \cup \{\varepsilon\})$ since $(T, sym)$ is a complete parse tree. It holds that $A \subseteq \{-1, 0, 1\}$ by assumption. We derive that $\sigma_{a_i}(d_i + 1) \geq d_i$ for all $i$ with $1 \leq i \leq \ell$. The optimality of $T$ entails that $c_i \leq d_i + 1$. Indeed, if $c_i > d_i + 1$ for some $i$ then we would obtain a complete flow tree of lesser rank by changing the labeling of the node $t_i$ using

$in(t_i) := d_i + 1$. This would contradict the optimality of $T$. By Lemma 4.3, it holds that $c_1 = c$ and $d_\ell = d$. It also follows from Lemma 4.3 that $d_i = c_{i+1}$ for all $i$ with $1 \leq i < \ell$. We get that $c - d = c_1 - d_\ell = (c_1 - d_1) + \cdots + (c_\ell - d_\ell) \leq \ell$.

We now prove that $c \leq d + \delta^{|V|}$. Assume towards a contradiction that $c > d + \delta^{|V|}$. It follows that $T$ has $\ell > \delta^{|V|}$ leaves. We derive from Remark A.1 that $(T, sym)$ is not elementary. By iteratively collapsing[5] nodes $s \prec t$ with $sym(s) = sym(t)$, we obtain a complete and elementary parse tree $(T', sym')$ with $|T'| < |T|$. The root labeling is preserved by this transformation, that is $sym'(\varepsilon) = \#$. Since $(T', sym')$ is elementary, it contains at most $\delta^{|V|}$ leaves. Therefore, it induces a complete flow tree $(T', sym', in', out')$ satisfying $in'(\varepsilon) = d + \delta^{|V|}$ and $out'(\varepsilon) \geq d$. We obtain that, $in'(\varepsilon) \leq in(\varepsilon)$, $sym(\varepsilon) = sym(\varepsilon)$, and $out'(\varepsilon) \geq out(\varepsilon)$. This contradicts the optimality of $T$. □

**Lemma 4.5.** *Let $X \in V$ and $n \in \mathbb{N}$. If $\lambda_X = +\infty$ and there is a derivation $X \stackrel{*}{\Longrightarrow} uXv$ such that $\sigma_u(n) > n$, then it holds that $\sigma_X(n) = +\infty$.*

*Proof.* Assume that $\lambda_X = +\infty$ and that there exists $u, v \in (V \cup A)^*$ such that $X \stackrel{*}{\Longrightarrow} uXv$ and $\sigma_u(n) > n$. Since every nonterminal is productive, there exists $b \in \mathbb{N}$ such that $\sigma_v(b) \geq 0$. By Remark 3.3, we derive that $(\sigma_v)^k(m + kb) \geq m$ for every $k, m \in \mathbb{N}$. Similarly, since $\sigma_u(n) \geq n + 1$, we get from Remark 3.3 that $(\sigma_u)^k(n) \geq n + k$ for every $k \in \mathbb{N}$. Define $\lambda = b + 1$. Since $\lambda < \lambda_X = +\infty$, there exists $m_0 \in \mathbb{N}$ such that $\sigma_X(m) \geq \lambda \cdot m$ for all $m \geq m_0$. For every $k \in \mathbb{N}$ with $k \geq m_0$, it holds that $X \stackrel{*}{\Longrightarrow} u^k X v^k$, which entails, by monotonicity of the summary functions, that

$$
\begin{array}{rcll}
\sigma_X(n) & \geq & \sigma_{u^k X v^k}(n) & \text{[Lemma 3.6]} \\
& = & \sigma_{v^k} \circ \sigma_X \circ \sigma_{u^k}(n) & \text{[Lemma 3.6]} \\
& \geq & \sigma_{v^k} \circ \sigma_X(n + k) & \\
& \geq & \sigma_{v^k}(\lambda \cdot (n + k)) & \\
& = & \sigma_{v^k}(\lambda \cdot n + k + kb) & [\lambda = b + 1] \\
& \geq & \lambda \cdot n + k &
\end{array}
$$

We have thus shown that $\sigma_X(n) \geq k$ for every $k \in \mathbb{N}$ with $k \geq m_0$. We conclude that $\sigma_X(n) = +\infty$. □

The two following facts are part of the proof of Proposition 4.9. Recall that, in the context of this proof, $(T, sym, in, out)$ is a complete flow tree that is optimal, and that $U$ is the set of all nodes $t \in T$ such that every proper ancestor $s \prec t$ satisfies Equation (1), which is copied below:

$$\text{For every ancestor } r \preceq s,\ sym(r) = sym(s) \implies in(r) \geq in(s)$$

*Fact 4.7.* The tree $U$, equipped with the restrictions to $U$ of the functions $sym$, $in$ and $out$, is a certificate.

[5] Collapsing two nodes $s \prec t$ consists in replacing the subtree rooted in $s$ by the subtree rooted in $t$.

*Proof.* It follows from $U \subseteq T$ and Lemma 4.1 that $U$ is a flow tree. Let us show that every leaf of $U$ satisfies the condition of Definition 4.6. Let $t$ be a leaf of $U$ such that $\lambda_{sym(t)} = +\infty$. Since $(T, sym)$ is a complete parse tree, every leaf $u$ of $T$ verifies $sym(u) \in (A \cup \{\varepsilon\})$, hence, $\lambda_{sym(u)} = 1$. It follows that $t$ has a child $u$ in $T$. But $u \notin U$ as otherwise $t$ would be internal in $U$. So there exists a proper ancestor $s \prec u$ that violates Equation (1). Since $t$ itself is in $U$, we get that $s = t$. We derive that there exists an ancestor $r$ of $s = t$ such that $sym(r) = sym(t)$ and $in(r) < in(t)$. $\square$

*Fact 4.8.* Let $r$ and $s$ be nodes in $U$ such that $r \prec s$.

1. If $s$ is internal in $U$ and $sym(r) = sym(s)$ then $out(s) < out(r)$, and
2. If $s$ is a child of $r$ then $out(s) \leq out(r) + (\delta - 1)\delta^{|V|}$.

*Proof.* Let us start with the first assertion. By contradiction, assume that $s$ is internal in $U$, $sym(r) = sym(s)$ and $out(s) \geq out(r)$. Since $s$ is internal in $U$, $s$ is the proper ancestor of some node in $U$, hence, $s$ verifies Equation (1). We derive that $in(s) \leq in(r)$. Observe that the subtree of $T$ rooted in $r$ contains more nodes than the subtree of $T$ rooted in $s$. It follows that the subtree of $T$ rooted in $r$ is not optimal, which contradicts the optimality of $T$. The second assertion is easily derived from Lemmas 4.3 and 4.4, the observation that $r$ has at most $\delta$ children, and the fact that $T$ is optimal. $\square$

## D Proofs for Section 5

**Lemma D.1.** *For every thin GVAS $G = (V, A, R)$ one can contruct a simple GVAS $G' = (V', A', R')$ such that $V \subseteq V'$ and $L_S^G = L_S^{G'}$ for all $S \in V$.*

*Proof.* We assume that $0 \in A$. Let us consider a production rule $X \vdash \alpha$ with $\alpha = a_1 \ldots a_i Y b_j \ldots b_1$ where $Y \in V$, and $a_1 \ldots, a_i, b_j, \ldots, b_1$ is a sequence of terminal symbols in $A$. We let $m \geq 1$ be a positive integer such that $i, j \leq m$. Define $a_{i+1}, \ldots, a_m$ and $b_m, \ldots, b_{j+1}$ to be 0, and introduce fresh nonterminal symbols $X_1, \ldots, X_{m-1}$. The production rule $X \vdash \alpha$ is then replaced by the production rules $X_{j-1} \vdash a_j X_j b_j$ where $1 \leq j \leq m$, $X_0 \stackrel{\text{def}}{=} X$, and $X_m \stackrel{\text{def}}{=} Y$. Just observe that such a transformation let the language $L_S$ unchanged. $\square$

**Lemma 5.3.** *For for all $c, d \in \mathbb{N}$, $c \xrightarrow{S} d$ if, and only if, the following relation holds:*

$$\phi_S(c, d) \stackrel{\text{def}}{=} \bigvee_{X \in V} \exists c', d' \in \mathbb{N} \quad (c, d) \xrightarrow{\Pi_X} (c', d') \wedge c' \xrightarrow{\Gamma_X} d' \tag{4}$$

*Proof.* To see this, fix any two numbers $c, d \in \mathbb{N}$. Assume first that $c \xrightarrow{S} d$. It means that there exists a word $w \in L_S$ such that $c \xrightarrow{w} d$. Since $w$ is a word over the terminal symbols, we deduce that a sequence of derivation steps from $S$ that produces $w$ must necessarily derive at some point a nonterminal symbol $X$ with a production rule $X \vdash \alpha$ such that $\alpha \in A^*$, and in particular $\alpha \in \Gamma_X$.

By considering the first time that a derivation step $X \xRightarrow{\alpha}$ with $\alpha \in \Gamma_X$ occurs, we deduce that all the previous derivation steps replace nonterminal symbols by words in $AVA$. We extract a sequence $X_0, \dots, X_k$ of nonterminal symbols with $X_0 = S$, a sequence $r_1, \dots, r_k$ of production rules $r_j \in R$ of the form $X_{j-1} \vdash a_j X_j b_j$ with $a_j, b_j \in A$, a production rule $r_{k+1} \in R$ of the form $X_k \vdash \alpha$ where $\alpha \in \Gamma_{X_k}$, and a word $w' \in L_\alpha$ such that:

$$w = a_1 \dots a_k w' b_k \dots b_1 \tag{5}$$

Since $c \xrightarrow{w} d$, we derive that there exists a sequence $c_0 \dots c_k \in \mathbb{N}$ and a sequence $d_k, \dots, d_0 \in \mathbb{N}$ satisfying the following relation.

$$c = c_0 \xrightarrow{a_1} c_1 \cdots \xrightarrow{a_k} c_k \xrightarrow{w'} d_k \xrightarrow{b_k} d_{k-1} \cdots \xrightarrow{b_1} d_0 = d \tag{6}$$

This is true if, and only if, in the 2-VAS $\boldsymbol{A}$, there exists a path

$$(c, d) = (c_0, d_0) \xrightarrow{(a_1, -b_1)} (c_1, d_1) \cdots \xrightarrow{(a_k, -b_k)} (c_k, d_k) \tag{7}$$

Let $c' \stackrel{\text{def}}{=} c_k$, $d' \stackrel{\text{def}}{=} d_k$, and $X \stackrel{\text{def}}{=} X_k$. Observe that $\pi \stackrel{\text{def}}{=} (a_1, -b_1) \dots (a_k, -b_k)$ is a word in $\Pi_X$ such that $(c, d) \xrightarrow{\pi} (c', d')$. Moreover, from $c' \xrightarrow{w'} d'$ we get that $c' \xrightarrow{\Gamma_X} d'$. Together this means that $\phi_S(c, d)$ is true.

Conversely, assume that $\phi_S(c, d)$ holds. Since $\psi_S(c, d)$ is a finite disjunction, there exist $X \in V$ and $c, d, c', d' \in \mathbb{N}$ such that $(c, d) \xrightarrow{\Pi_X} (c', d')$ and $c' \xrightarrow{\Gamma_X} d'$. Let us consider a word $\pi \in \Pi_X$ of the form $\pi = (a_1, -b_1) \dots (a_k, -b_k)$ such that $(c, d) \xrightarrow{\pi} (c', d')$. We also introduce a word $\alpha \in \Gamma_X$ such that $c' \xrightarrow{\alpha} d'$. This last relation shows that there exists $w' \in L_G(\alpha)$ such that $c' \xrightarrow{w'} d'$. From $(c, d) \xrightarrow{\pi} (c', d')$ we derive a sequence $(c_0, d_0), \dots, (c_k, d_k)$ of pairs in $\mathbb{N} \times \mathbb{N}$ such that $(c_k, d_k) = (c', d')$ and such that relation (7) and thus (6) hold. Hence, $c \xrightarrow{w} d$ where $w$ is the word satisfying (5). Since $w \in L_S$, it follows that $c \xrightarrow{S} d$. $\square$

## E Proofs for Section 6

By definition of the displacement, if $\Delta_S < +\infty$, then there exists a word $w \in L_S$ such that $\Delta_S = \sum w$. The following lemma provides a way to bound the length of such a word $w$.

**Lemma E.1.** *For every nonterminal $S \in V$ with $\Delta_S < +\infty$, there is a complete elementary parse tree with root labeled by $S$ and yield $w \in A^*$ such that $\Delta_S = \sum w$.*

*Proof.* Since $\Delta_S < +\infty$, there exists a complete parse tree with root labeled by $S$ and yield $w \in A^*$ such that $\sum w = \Delta_S$. Let $(T, sym)$ be such a parse tree with the fewest possible number of nodes and assume towards a contradiction that $T$ is not elementary. This means there exists $s \prec t$ in $T$ and $X \in V$ such that $sym(s) = X = sym(t)$. The subtree rooted in $s$ provides a derivation

$X \xRightarrow{*} uXv$ for two words $u, v$ in $A^*$. Notice that if $\sum u + \sum v > 0$ then $\Delta_X = +\infty$. Then, Lemma 3.6 implies that $\Delta_S \geq \Delta_{uXv} = \Delta_u + \Delta_X + \Delta_v = +\infty$, which contradicts the assumption of the lemma. Therefore, $\sum u + \sum v \leq 0$. By collapsing the subtree $\{t' \in T \mid s \preceq t' \wedge t \not\preceq t'\}$, we get a new parse tree $(T', sym')$ with $|T'| < |T|$, $sym'(\varepsilon) = S$ and yield $w' \in A^*$ satisfying $\sum w' = \sum w - (\sum u + \sum v) \geq \sum w \geq \Delta_S$. Since clearly, $w' \in L_S$, by definition of the displacement it holds that $\sum w' \leq \Delta_S$ and therefore that $\sum w' = \Delta_S$. This contradicts our assumed minimality of $T$. Hence $T$ is elementary. $\square$

The corollary below follows from Lemma E.1 and the observation (Remark A.1) that the yield of an elementary parse tree is a word of length bounded by $\delta^{|V|}$.

**Corollary E.2.** *For every nonterminal $S \in V$ with $\Delta_S < +\infty$, and for every $c \geq \delta^{|V|}$ there exists an elementary complete flow tree with root $\varepsilon : cSd$ such that $d = c + \Delta_S$.*

*Proof.* According to Lemma E.1, there exists a complete elementary parse tree $(T, sym)$ with root labeled by $S$ and yield $w \in A^*$ such that $\Delta_S = \sum w$. Since this tree is elementary, it has no more than $\delta^{|V|}$ leaves. Hence, $|w| \leq \delta^{|V|} \leq c$, which entails that $c \xrightarrow{w} c + \Delta_S$. It is routinely checked that the parse tree $(T, sym)$ induces an elementary complete flow tree with root $\varepsilon : cSd$, where $d = c + \Delta_S$. $\square$

**Lemma 6.1.** *Let $S \in V$ be a nonterminal with $\Delta_S < +\infty$. Then it holds that $\sigma_S(n) = n + \Delta_S$ for every $n \in \overline{\mathbb{N}}$ such that $n \geq \delta^{|V|}$.*

*Proof.* Observe that $\sigma_S(n) \leq n + \Delta_S$ holds for every $S \in V$ and $n \in \overline{\mathbb{N}}$. The remaining inequality follows from Corollary E.2 and Lemma 4.1. $\square$

**Proposition 6.2.** *For every nonterminal $S \in V$ with $\Delta_S < +\infty$, the function $\sigma_S$ is effectively computable.*

*Proof.* Let $S \in V$ with $\Delta_S < +\infty$, and let $c \in \mathbb{N}$. Observe that $\sigma_S(c) \leq c + \Delta_S$. Therefore, the computation of $\sigma_S(c)$ reduces to the question whether $\sigma_S(c) \geq d$, given $d \in \mathbb{N}$. To decide the latter, we show that $\sigma_S(c) \geq d$ if, and only if, there exists a complete flow tree with root $\varepsilon : bSe$ satisfying $b \leq c$ and $e \geq d$, and of height bounded by $h \stackrel{\text{def}}{=} |V| \cdot (\delta^{|V|} + 1)$. The "if" direction follows from Lemma 4.1 and the monotonicity of the summary function $\sigma_S$. For the "only if" direction, assume that $\sigma_S(c) \geq d$. By Lemma 4.2, there exists a complete flow tree with root $\varepsilon : bSe$ satisfying $b \leq c$ and $e \geq d$. Pick one, say $(T, sym, in, out)$, that contains the least number of nodes $t \in T$ with $|t| > h$. We show that, in fact, $T$ contains no such node. Since $\Delta_S < +\infty$, we derive from Lemma 3.6 that $\Delta_{sym(r)} < +\infty$ for every node $r \in T$. Now, consider a leaf $t$ in $T$. Assume, towards a contradiction, that $|t| > h$. The main observation is that for every two nodes $r, s \in T$,

$$r \prec s \prec t \wedge sym(r) = sym(s) \implies in(r) \neq in(s) \tag{8}$$

For if this were not the case, then

– either $out(r) \leq out(s)$, in which case we could replace the subtree rooted in $r$ by the subtree rooted in $s$, contradicting the minimality assumption on $T$.
– or $out(r) > out(s)$, which would entail, with the same reasoning as in the proof of Lemma E.1, that $\Delta_{sym(r)} = +\infty$, which is impossible.

By the pigeonhole principle, it follows from Equation (8) that there exists an ancestor $s \prec t$ such that $|s| \leq |V| \cdot \delta^{|V|}$ and $in(s) \geq \delta^{|V|}$. The height of the subtree rooted in $s$ is strictly larger than $|V|$, since $t$ is in it. Because $\Delta_{sym(s)} < +\infty$ we can use Corollary E.2 and replace, without violating the flow conditions, the subtree rooted in $s$ by a complete flow tree of height at most $|V|$. This contradicts the minimality assumption on $T$.

The observation that $in(t)$ and $out(t)$ are both bounded by $in(\varepsilon)+\delta^h$ for every node $t$ of a complete flow tree of height $h$ concludes the proof the proposition. □

**Lemma 6.3.** *Let $X \in V$ be a nonterminal. If there is a derivation $X \xRightarrow{*} uXv$ such that $\Delta_{uv} = +\infty$ then it holds that $\lambda_X = +\infty$.*

*Proof.* Assume that $X \xRightarrow{*} uXv$ with $\Delta_{uv} = +\infty$. Let $\lambda \in \mathbb{R}$ with $\lambda \geq 1$, and let us show that $\lambda_X \geq \lambda$. It is routinely checked that, since $\Delta_{uv} = +\infty$, there exists $\mu \in \{\sum z \mid z \in L_u\}$ and $\nu \in \{\sum z \mid z \in L_v\}$ such that $\lambda\mu + \nu \geq 0$ and $\mu + \nu \geq 1$. Observe that $\Delta_u \geq \mu$, $\Delta_X \geq 0$ and $\Delta_v \geq \nu$. Therefore, there exists $m \in \mathbb{N}$ such that $\sigma_u(m) \geq m + \mu$, $\sigma_X(m) \geq m$ and $\sigma_v(m) \geq m + \nu$. It follows from Remark 3.3 that these inequalities hold for all $n \geq m$ as well. Let $n, k \in \mathbb{N}$ such that $n \geq m$ and $n + k\mu \geq m$. Note that $n + k\mu + k\nu \geq m$ since $\mu + \nu \geq 1$. Since $X \xRightarrow{*} u^k X v^k$, we get, by monotonicity of the summary functions, that

$$\begin{aligned}
\sigma_X(n) &\geq \sigma_{v^k} \circ \sigma_X \circ \sigma_{u^k}(n) && \text{[Lemma 3.6]} \\
&\geq \sigma_{v^k} \circ \sigma_X(n + k\mu) \\
&\geq \sigma_{v^k}(n + k\mu) \\
&\geq n + k\mu + k\nu \\
&\geq n + k \cdot \max\{1, \mu(1 - \lambda)\} && [\mu + \nu \geq 1 \ \wedge \ \lambda\mu + \nu \geq 0]
\end{aligned}$$

If $\mu \geq 0$ then, for every $k \in \mathbb{N}$, it holds that $n + k\mu \geq m$, hence, $\sigma_X(n) \geq n + k$. We derive that $\sigma_X(n) = +\infty$ for every $n \geq m$, which entails that $\lambda_X = +\infty$. Otherwise, $\mu < 0$. Take $k = \lfloor \frac{n-m}{-\mu} \rfloor$ and let $r = n - m + k\mu$. Observe that $0 \leq r \leq -\mu - 1$. Since $n + k\mu \geq m$, we get that $\sigma_X(n) \geq n - k\mu(\lambda - 1)$ from the above inequalities. We derive that $\sigma_X(n) \geq \lambda n + (\lambda - 1)(\mu + 1 - m)$ for every $n \geq m$, which entails that $\lambda_X \geq \lambda$. □

We now show that the transformations used in our reduction to thin GVAS are indeed correct, i.e., produce equivalent systems. Recall that two GVAS $G = (V, A, R)$ and $G' = (V', A', R')$ are called *equivalent* if firstly $V = V'$, secondly $\lambda_X^G = \lambda_X^{G'}$ for every nonterminal $X$, and thirdly $\sigma_X^G = \sigma_X^{G'}$ for every nonterminal $X$ *with finite ratio*.

*Fact 6.4.* The unfolding of $X$ is equivalent to $G$.

*Proof.* Recall that the *unfolding* of a nonterminal $X$ with $\Delta_X^G < +\infty$, is the GVAS $H = (V, A, R')$ where $R'$ is obtained from $R$ by removing all production rules $X \vdash \alpha$ and instead adding, for every $0 \leq i \leq \delta^{|V|}$ with $j = \sigma_X^G(i) > -\infty$, a rule $X \vdash (-1)^i(1)^j$.

We first prove that $\sigma_X^G = \sigma_X^H$. First note that $\sigma_X^G(-\infty) = \sigma_X^H(-\infty) = -\infty$ and $\sigma_X^G(+\infty) = \sigma_X^H(+\infty) = +\infty$. Let $n \in \mathbb{N}$. By definition of $H$, we get that $\sigma_X^H(n) = \max\{n - i + \sigma_X^G(i) \mid 0 \leq i \leq \delta^{|V|} \wedge i \leq n\}$. It follows from Remark 3.3 that $\sigma_X^H(n) = n - m + \sigma_X^G(m)$ where $m = \min\{\delta^{|V|}, n\}$. If $n \leq \delta^{|V|}$ then we immediately get that $\sigma_X^H(n) = \sigma_X^G(n)$. Otherwise, $n > \delta^{|V|}$ and $\sigma_X^H(n) = n - \delta^{|V|} + \sigma_X^G(\delta^{|V|})$. We derive from Lemma 6.1 that $\sigma_X^H(n) = \sigma_X^G(n)$.

We now prove that $\sigma_S^G = \sigma_S^H$ for every nonterminal $S$. Let $c, d \in \mathbb{N}$. Assume that $\sigma_S^G(c) \geq d$. By Lemma 4.2, there exists a complete flow tree $(T, \mathit{sym}, \mathit{in}, \mathit{out})$ for $G$ with root $\varepsilon : cSd$. Let $U$ denote the set of all nodes $t \in T$ such that every proper ancestor $s \prec t$ verifies $\mathit{sym}(s) \neq X$. By definition, the set $U$ is a nonempty and prefix-closed subset of $T$. Moreover, $\mathit{sym}(t) \neq X$ for each internal node $t$ of $U$, and $\mathit{sym}(t) \in (\{X\} \cup A)$ for each leaf $t$ of $U$. It follows that $U$ is a flow tree for $H$, since $\sigma_\#^G = \sigma_\#^H$ for every $\# \in (\{X\} \cup A)$. Note that the root of $U$ also satisfies $\varepsilon : cSd$. We derive from Lemma 4.1 that $\sigma_S^H(c) \geq d$.

Conversely, the same reasoning as above shows that $\sigma_S^H(c) \geq d$ implies $\sigma_S^G(c) \geq d$. We have thus shown that $\sigma_S^G(c) \geq d \Leftrightarrow \sigma_S^H(c) \geq d$, for every $c, d \in \mathbb{N}$. It follows that $\sigma_S^G = \sigma_S^H$. By definition of the ratio, we also get that $\lambda_S^G = \lambda_S^H$. □

*Fact 6.6.* The abstraction of $X$ is equivalent to $G$.

*Proof.* Recall that the the *abstraction* of a nonterminal $X \in V$ with $\lambda_X^G = +\infty$, is the GVAS $H = (V, A \cup \{1\}, R')$ where $R'$ is obtained from $R$ by removing all production rules $X \vdash \alpha$ and replacing them by the two rules $X \vdash 1X \mid \varepsilon$.

Let $D_X$ denote the set of nonterminals $S \in V$ such that $X$ is derivable from $S$ in $G$. Note that $D_X$ is also the set of nonterminals $S \in V$ such that $X$ is derivable from $S$ in $H$. Recall that $\lambda_X^G = +\infty$. By definition of $H$, it holds that $\lambda_X^H = +\infty$. It follows from Lemma 3.6 that $\lambda_S^G = \lambda_S^H = +\infty$ for every $S \in D_X$.

Now consider a nonterminal $S \notin D_X$. It is readily seen that $G$ and $H$ have the same derivations $S \xRightarrow{*} w$ starting from $S$. Therefore, $L_S^G = L_S^H$. It follows that $\sigma_S^G = \sigma_S^H$. By definition of the ratio, we also get that $\lambda_S^G = \lambda_S^H$. The observation that every nonterminal with finite ratio is in $V \setminus D_X$ concludes the proof. □

**Corollary 6.8.** *The question whether $\lambda_X < +\infty$ holds for a given GVAS $G$ and a given nonterminal $X$, is decidable. Moreover, if $\lambda_X < +\infty$ then the function $\sigma_X$ is effectively computable.*

*Proof.* By Proposition 6.7, it is enough show the claim for thin GVAS. Let us consider a thin GVAS $G = (V, A, R)$ and a nonterminal $X \in V$. By Theorem 5.1, the relation $\xrightarrow{X}$ is effectively definable in Presburger arithmetic. Therefore, so is the set $\Sigma_X(n) \stackrel{\text{def}}{=} \{d \mid \exists c \leq n : c \xrightarrow{X} d\}$, for any given $n \in \overline{\mathbb{N}}$. We derive that its supremum $\sigma_X(n) = \sup \Sigma_X(n)$ is computable.

We now prove that the question whether $\lambda_X < +\infty$ is decidable. Since the relation $\xrightarrow{X}$ is effectively definable in Presburger arithmethic, it is effectively semilinear [7]. This means that we can compute a finite family $\{(\boldsymbol{b}_i, \boldsymbol{P}_i)\}_{i \in I}$ of vectors $\boldsymbol{b}_i$ in $\mathbb{N}^2$ and finite subsets $\boldsymbol{P}_i$ of $\mathbb{N}^2$, with $\boldsymbol{P}_i = \{\boldsymbol{p}_i^1, \ldots, \boldsymbol{p}_i^{\ell_i}\}$, such that $\xrightarrow{X} = \bigcup_{i \in I} \left(\boldsymbol{b}_i + \mathbb{N}\boldsymbol{p}_i^1 + \cdots + \mathbb{N}\boldsymbol{p}_i^{\ell_i}\right)$. We consider two cases.

- If there exists $i \in I$ and a vector $\boldsymbol{p}$ in $\bigcup_{i \in I} \boldsymbol{P}_i$ such that $\boldsymbol{p}(1) = 0$ and $\boldsymbol{p}(2) > 0$, then $\boldsymbol{b}_i(1) \xrightarrow{X} (\boldsymbol{b}_i(2) + k\boldsymbol{p}(2))$ for every $k \in \mathbb{N}$. It follows that $\sigma_X(\boldsymbol{b}_i(1)) = +\infty$, which entails, by monotonicity of $\sigma_X$, that $\lambda_X = +\infty$.
- Otherwise, there exists $\lambda \in \mathbb{R}$ with $\lambda \geq 1$ such that $\boldsymbol{p}(2) \leq \lambda \boldsymbol{p}(1)$ for every vector $\boldsymbol{p}$ in $\bigcup_{i \in I} \boldsymbol{P}_i$. Define $b = \max\{\boldsymbol{b}_i(2) \mid i \in I\}$. It is routinely checked that $d \leq \lambda c + b$ for every $c, d$ with $c \xrightarrow{X} d$. We derive that $\sigma_X(n) \leq \lambda n + b$ for every $n \in \mathbb{N}$, which implies that $\lambda_X \leq \lambda$.

We have shown that $\lambda_X = +\infty$ if, and only if, there exists $\boldsymbol{p}$ in $\bigcup_{i \in I} \boldsymbol{P}_i$ with $\boldsymbol{p}(1) = 0$ and $\boldsymbol{p}(2) > 0$. The latter condition is decidable, and so is the former. ☐

**Lemma E.3.** *Let $(T, sym)$ be a parse tree and let $in, out : T \to \mathbb{N}$. Then $(T, sym, in, out)$ is a certificate if the three following conditions hold:*

*(i) All internal nodes satisfy the first flow condition,*
*(ii) Every leaf $t \in T$ with $\lambda_{sym(t)} < +\infty$ satisfies the second flow condition, and*
*(iii) Every leaf $t \in T$ with $\lambda_{sym(t)} = +\infty$ has a proper ancestor $s \prec t$ such that $sym(s) = sym(t)$ and $in(s) < in(t)$.*

*Proof.* Assume that $(i)$–$(iii)$ hold. We only need to show that every leaf of $T$ satisfies the second flow condition. By contradiction, assume that $T$ contains a leaf $t$ with $out(t) \not\leq \sigma_{sym(t)}(in(t))$. It follows from $(ii)$ and $(iii)$ that $\lambda_{sym(t)} = +\infty$ and that $t$ has a proper ancestor $s \prec t$ such that $sym(s) = sym(t)$ and $in(s) < in(t)$. Let $t_1, \ldots, t_\ell$, with $t_i : c_i \#_i d_i$, denote the leaves of the subtree of $T$ rooted in $s$, in lexicographic order (informally, from left to right). Obviously, $t = t_k$ for some $k$ in $\{1, \ldots, \ell\}$. We may suppose, without loss of generality, that $t_1, \ldots, t_{k-1}$ satisfy the second flow condition. This means that $d_i \leq \sigma_{\#_i}(c_i)$ for all $i$ with $1 \leq i < k$. Since every internal node satisfies the first flow condition, it holds that $in(s) \geq c_1$ and $d_i \geq c_{i+1}$ for all $i$ with $1 \leq i < k$. We derive from the monotonicity of summary functions that

$$\begin{aligned} \sigma_{\#_1 \cdots \#_{k-1}}(in(s)) &= \sigma_{\#_{k-1}} \circ \cdots \circ \sigma_{\#_1}(in(s)) && \text{[Lemma 3.6]} \\ &\geq \sigma_{\#_{k-1}} \circ \cdots \circ \sigma_{\#_1}(c_1) && [in(s) \geq c_1] \\ &\geq c_k && [\sigma_{\#_i}(c_i) \geq d_i \geq c_{i+1}] \\ &> in(s) && [c_k = in(t) > in(s)] \end{aligned}$$

Define $u = \#_1 \cdots \#_{k-1}$, $X = sym(s) = \#_k$, and $v = \#_{k+1} \cdots \#_\ell$. Recall that $t_1, \ldots, t_\ell$ are the leaves, in lexicographic order, of the subtree of $T$ rooted in $s$. Therefore, we have the derivation $X \xRightarrow{*} uXv$. We obtain from Lemma 4.5 that $\sigma_X(in(s)) = +\infty$. Since $in(t) \geq in(s)$, we get that $\sigma_X(in(t)) = +\infty$, which contradicts our assumption that $out(t) \not\leq \sigma_X(in(t))$. ☐