Matrix Interpretations on Polyhedral Domains

Johannes Waldmann

F-IMN, HTWK Leipzig, Germany
johannes.waldmann@htwk-leipzig.de

Abstract

We refine matrix interpretations for proving termination and complexity bounds of term rewrite systems we restricting them to domains that satisfy a system of linear inequalities. Admissibility of such a restriction is shown by certificates whose validity can be expressed as a constraint program. This refinement is orthogonal to other features of matrix interpretations (complexity bounds, dependency pairs), but can be used to improve complexity bounds, and we discuss its relation with the usable rules criterion. We present an implementation and experiments.

1998 ACM Subject Classification F.3.1 Specifying and Verifying and Reasoning about Programs

Keywords and phrases Termination of term rewriting, matrix interpretations, constraint programming, linear inequalities

Digital Object Identifier 10.4230/LIPIcs.RTA.2015.318

1 Introduction

To prove termination of a rewrite system, we can give an interpretation of function symbols that defines a well-founded monotone algebra that is compatible with the rules [30]. By restricting the class of interpretations, we get termination proof methods that can be automated, in the sense that the interpretation is determined by a finite set of parameters that can be computed by a program. An early instance is polynomial interpretations [7], where the parameters are the coefficients of polynomials. We are concerned here with vector-valued interpretations, where the parameters are coefficients (for the matrix representations) of multi-linear functions [13, 11]. The domain for these interpretations is $\mathbb{N}^d$, ordered by $x > y \quad \text{iff} \quad x_1 > y_1 \wedge x_2 \geq y_2 \wedge \ldots \wedge x_d \geq y_d$. This order is non-total, and the method can prove non-simple termination.

Several variants and modifications have been investigated, and we list some that are relevant for the present investigation:

- By restricting the shape of matrices, we can prove not just termination, but polynomial derivational complexity [18, 28].
- Vector-valued interpretations can be used to define reduction pairs for termination proofs in the dependency pair framework [2]. The main point here is that monotonicity constraints can be relaxed.
- Instead of vectors and linear functions over $(\mathbb{N}, +, \cdot)$, we can take vectors and linear functions over other semirings, e.g., the arctic semiring $(\mathbb{N} \cup \{-\infty\}, \max, +)$ [14]. In this semiring, monotonicity of operations is different (from $\mathbb{N}$), in a way that this is well-suited to the dependency pair framework.
- Returning to $\mathbb{N}^d$, we may consider different orders on that domain [19].

In the present paper, we modify matrix interpretations in yet another way: we keep the semiring ($\mathbb{N}$) and order $>$ on $\mathbb{N}^d$, but restrict the domain of interpretations by additional linear inequalities. We obtain a convex polyhedral domain $D \subseteq \mathbb{N}^d$. Using such domains is a

26th International Conference on Rewriting Techniques and Applications (RTA'15).
Editor: Maribel Fernández; pp. 318–333
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

standard approach in static analysis of imperative programs that is routinely used in optimizing compilers (cf. the Parma Polyhedral Library [3] for the GNU Compiler Collection). In the context of automated termination analysis of rewrite systems, polyhedral domains were first suggested by Lucas and Meseguer [17]. In the present paper, we substantially extend their idea.

Let us illustrate the method by an example.

▶ **Example 1.** The goal is to prove termination and polynomial derivational complexity of the string rewriting system

$$R = \{fg \to ff, gf \to gg\},$$

where the string fg is an abbreviation for the term $f(g(x))$, etc. We define a domain $D = \{(x_1, x_2, x_3) \in \mathbb{N}^3 \mid x_3 \geq x_2 + 1\}$. This set is non-empty, e.g., $(0, 0, 1) \in D$. Then both

$$\begin{aligned}
[f](x_1, x_2, x_3) &= (x_1 + 2x_2 + 1, 0, x_3 + 1) \\
[g](x_1, x_2, x_3) &= (x_1 \qquad\qquad , x_3, x_3 + 1)
\end{aligned}$$

map D into D. Now we combine interpretations:

$$\begin{aligned}
[fg](x) &= (x_1 + \qquad 2x_3 + 1, 0, x_3 + 2), \\
[ff](x) &= (x_1 + 2x_2 + \qquad 2, 0, x_3 + 2).
\end{aligned}$$

The point is now that $\forall x \in D : [fg](x) > [ff](x)$ even though we don't have a point-wise inequality between corresponding coefficients in the first component: the coefficient of x_2 in $[fg](x)_1$ is zero, and the coefficient of x_2 in $[ff](x)_1$ is two.

By the condition that defines D, we have

$$[fg](x)_1 \geq x_1 + 2x_3 + 1 \geq x_1 + (2x_2 + 2) + 1 > x_1 + 2x_2 + 2 = [ff](x)_1.$$

Additionally, we verify (without using D conditions)

$$[gf](x) = (x_1 + 2x_2 + 1, x_3 + 1, x_3 + 2) > (x_1, x_3 + 1, x_3 + 2) = [gg](x).$$

We also note that $[f]$ and $[g]$ are strictly monotone w.r.t. $>$, since all coefficients are non-negative, and the coefficient of x_1 in the first component is positive.

This proves that $[\cdot]$ is a strictly monotone D-valued interpretation that is strictly compatible with the rewrite system R. So, the interpretation certifies termination of R [11].

Moreover, the coefficient matrices of $[\cdot]$ are upper triangular, so we actually proved polynomial derivational complexity [18]. By closer inspection (there are just two occurrences of 1 on the main diagonals) the complexity is quadratic. This property of R was known before, e.g., CaT [15] proves it via root labelling [22]. Ours seems to be the first "direct" proof.

In the remainder of the paper, we formally define and justify the method (Sections 3 and 4), discuss modifications with respect to derivational complexity (Section 6) and the dependency pair method (Sections 7) with the usable-rules criterion (Section 8) and finally (Section 9) describe an implementation and experiments.

2 Notation and Preliminaries

A ranked signature maps function symbols to arities, e.g., $\Sigma = \{(a,2),(f,1),(g,1)\}$. The size $\|\Sigma\|$ of a signature is $\sum_{(f,k)\in\Sigma)} k$, e.g., $\|\Sigma\| = 4$. We consider terms in $\mathsf{Term}(\Sigma, V)$ with symbols from Σ and variables from V. We denote by $\mathsf{Var}(t)$ the set of variables appearing in a term t, and by $|t|$ the size of the term (the number of its positions). A rewrite rule is a pair $(l,r) \in \mathsf{Term}(\Sigma,V)^2$, written $l \to r$, and a set R of rewrite rules defines a relation $\to_R$ on $\mathsf{Term}(\Sigma)$ in the usual way. We write $\to_1 \circ \to_2$ for the product of relations $\to_1$ and $\to_2$. For a relation $\to$, we denote by $\to^k$ its k-fold product, by $\to^+$ its transitive closure, and by $\to^*$ its transitive reflexive closure. We write $\to_1 / \to_2$ for the relation $\to_2^* \circ \to_1 \circ \to_2^*$. We say a relation $\to$ is well-founded if there is no infinite $\to$-chain. A rewrite system R is called terminating if $\to_R$ is well-founded. A rewrite system R is called terminating relative to a rewrite system S if $\to_R / \to_S$ is well-founded. The derivational complexity $\mathsf{dc}_\to$ of a relation $\to$ on terms describes the length of $\to$-chains as a function of the size of the start term. Formally, $\mathsf{dc}_\to : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ is the function $s \mapsto \sup\{k \mid \exists t_1, t_2 \in \mathsf{Term}(\Sigma) : |t_1| \leq s \wedge t_1 \to^k t_2\}$. This is in fact a function $\mathbb{N} \to \mathbb{N}$ in case $\to$ is terminating and finitely branching. We write dc_R for $\mathsf{dc}_{\to_R}$ and $\mathsf{dc}_{R/S}$ for $\mathsf{dc}_{\to_R/\to_S}$.

An algebra A for signature Σ is given by a domain D_A, and for each k-ary symbol f from Σ, a k-ary function $[f]_A : D_A^k \to D_A$. The algebra then maps each $t \in \mathsf{Term}(\Sigma)$ to an element of D_A, also denoted $[t]_A$, and by extension, each $t \in \mathsf{Term}(\Sigma, V)$ with $|Var(t)| = k$, to a k-ary function $[t]_A : D_A^k \to D_A$. An algebra is monotone w.r.t. an order $>_A$ on its domain if each $[f]_A$ is monotone in each argument: $x_i >_A x_i'$ implies $[f]_A(x_1,\dots,x_i,\dots,x_k) >_A [f]_A(x_1,\dots,x_i',\dots,x_k)$. An algebra with order $>_A$ is well-founded if $>_A$ is well-founded. A Σ-algebra A is compatible with relation $\to$ if $x \to y$ implies $[x]_A > [y]_A$. If A is clear from the context, we write $[t]$ for $[t]_A$, and $>$ for $>_A$, etc.

A d-dimensional matrix interpretation defines an algebra with domain $\mathbb{N}^d$, the order is given by $x > y$ iff $x_1 > y_1 \wedge x_2 \geq y_2 \wedge \ldots \wedge x_d \geq y_d$, and the interpretation of a k-ary symbol f is given by a multi-linear function of shape $[f](x_1,\dots,x_k) = F_0 + \sum_i F_i x_i$, where F_0 is a vector (the absolute part), and $F_1,\dots,F_k$ are matrices (the coefficients for the linear part). Because of this presentation, we think of vectors $x_1,\dots,x_k,F_0$ as column vectors. All coefficients in $F_0,F_1,\dots,F_k$ are nonnegative (because $[f]$ must map into $\mathbb{N}^d$). A matrix interpretation is monotone w.r.t. $>$ if each top left entry of $F_1,\dots,F_k$ is positive. A matrix interpretation is strictly (weakly) compatible with a rule (l,r) with $|\mathsf{Var}(l)\cup\mathsf{Var}(r)| = k$ if the interpretations $[l]$ and $[r]$, which can be written as $[l] = F_0 + \sum_i F_i x_i, [r] = G_0 + \sum_i G_i x_i$, verify $F_0 > G_0$ ($F_0 \geq G_0$) and for all $1 \leq i \leq k$, $F_i \geq G_i$ (here, $\geq$ on matrices is the point-wise extension of $\geq$ on $\mathbb{N}$).

▶ **Example 2.** For $\{f(g(x)) \to f(f(x)), g(f(x)) \to g(g(x))\}$, the interpretation

$$[f](x_1) = \begin{pmatrix}0\\0\\1\end{pmatrix} + \begin{pmatrix}1&1&0\\0&0&0\\0&1&1\end{pmatrix} \cdot x_1, \qquad [g](x_1) = \begin{pmatrix}1\\1\\0\end{pmatrix} + \begin{pmatrix}1&0&2\\0&1&1\\0&0&0\end{pmatrix} \cdot x_1$$

is monotone and compatible with the rules, since

$$[fg](x_1) = \begin{pmatrix}2\\0\\2\end{pmatrix} + \begin{pmatrix}1&1&3\\0&0&0\\0&1&1\end{pmatrix} \cdot x_1 > [ff](x_1) = \begin{pmatrix}0\\0\\2\end{pmatrix} + \begin{pmatrix}1&1&0\\0&0&0\\0&1&1\end{pmatrix} \cdot x_1$$

$$[gf](x_1) = \begin{pmatrix}3\\2\\0\end{pmatrix} + \begin{pmatrix}1&3&2\\0&1&1\\0&0&0\end{pmatrix} \cdot x_1 > [gg](x_1) = \begin{pmatrix}2\\2\\0\end{pmatrix} + \begin{pmatrix}1&0&2\\0&1&1\\0&0&0\end{pmatrix} \cdot x_1$$

We note that it is decidable whether a given d-dimensional matrix interpretation is monotone, and compatible with a given R. The decision procedure is a straight-line program (the control flow does not depend on the data), so we can derive a constraint system from it, and use it to compute a suitable interpretation, once R is given. The constraint language contains inequalities between polynomials (called `QFNIA` in [4]). Because of its intractability, one often restricts unknown numbers to finite ranges, and represents them as bit vectors (using `QFBV` in [4]), in binary, or even unary [8].

3 Interpretations on Polyhedral Domains

We now define the concepts, and illustrate them by formalizing Example 1.

▶ **Definition 3.** A *polyhedral interpretation* A with domain dimension $d \in \mathbb{N}$ and constraint dimension $c \in \mathbb{N}$ for a ranked signature Σ consists of

- (the polyhedral domain) a matrix $C_A \in \mathbb{Q}^{c\times d}$ and a vector $B_A \in \mathbb{Q}^{c\times 1}$, describing the set $D_A = \{x \mid x \in \mathbb{Q}^d, x \geq 0 \wedge C_A x + B_A \geq 0\}$
- (the underlying interpretation) for each $(f, k) \in \Sigma$, a (column) vector $F_0 \in \mathbb{N}^{d\times 1}$, and a list of k square matrices $F_1, \dots, F_k \in \mathbb{N}^{d\times d}$, describing a function $[f]_A : (\mathbb{N}^d)^k \to \mathbb{N}^d : (x_1, \dots, x_k) \mapsto F_0 + \sum_i F_i x_i$

Subscript A is omitted when it can be inferred from the context.

Note that we use rational numbers ($\mathbb{Q}$) for describing the constraints (this fits with the theory of linear algebra that we will need) but natural numbers for the interpretation (this fits with well-foundedness of the domain). In examples, and in our implementation (see Section 9), we will substitute $\mathbb{Z}$ for $\mathbb{Q}$.

▶ **Example 4** (Example 1 continued)**.** The signature is $\Sigma = \{(f, 1), (g, 1)\}$, the domain dimension is $d = 3$, the constraint dimension is $c = 1$, the domain is described by $C = \begin{pmatrix} 0 & -1 & 1 \end{pmatrix}, B = \begin{pmatrix} -1 \end{pmatrix}$, and the underlying interpretation is

$$[f](x) = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, [g](x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Since we will later determine polyhedral interpretations by solving constraint systems, we collect information that helps to determine their size. In particular, we are interested in how many extra constraints we need, compared to the standard matrix method.

▶ **Observation 5.** *A domain description (in Def. 3) contains $c \cdot (d + 1)$ unknowns.*

Properties of a polyhedral interpretation will be derived from the existence of a valid certificate, which contains a part that refers to the domain, and a part that refers to the rewrite system. These certificates are derived from a general principle

▶ **Lemma 6** (Inhomogenous Farkas' Lemma)**.** *([25, 26]) A linear inequality $a^T x \leq p$ is a consequence of a solvable system of inequalities $Ax \leq b$ iff there is some $y \geq 0$ with $a = A^T y$ and $y^T b \leq p$.*

In other words, the conclusion is implied by a nonnegative linear combination of the premises. Our certificates are in fact representations of the coefficients in that linear combination.

Given a polyhedral interpretation A for signature Σ, we ask for $(f, k) \in \Sigma$ whether $[f]_A : D^k \to D$. We have the representation $[f](x_1, \dots, x_k) = F_0 + \sum_i F_i x_i$, with $F_0 \geq$

$0, \ldots, F_i \geq 0$, and we know that the arguments are from the domain: $\forall i : x_i \in D$, that is, $\forall i : x_i \geq 0 \wedge Cx_i + B \geq 0$. We combine all these assumptions, and collect them in a matrix (with $k \cdot (d + c)$ rows, $k \cdot d + 1$ columns)

$$\begin{pmatrix} I & 0 & \ldots & 0 & 0 \\ 0 & I & & 0 & 0 \\ \vdots & & \ddots & & \\ 0 & 0 & \ldots & I & 0 \\ C & 0 & \ldots & 0 & B \\ 0 & C & & 0 & B \\ \vdots & & \ddots & & \\ 0 & 0 & \ldots & C & B \end{pmatrix} \tag{1}$$

Do we have $[f](x_1, \ldots, x_k) \in D$? In other words, do these inequalities imply $C(F_0 + \sum_i F_i x_i) + B \geq 0$? (Note that $[f](x_1, \ldots, x_k) \geq 0$ is already implied by $F_i \geq 0$ and $x_i \geq 0$.) In our matrix notation, the conclusion is

$$\begin{pmatrix} CF_1 & CF_2 & \ldots & CF_k & CF_0 + B \end{pmatrix}$$

with c rows, $k \cdot d + 1$ columns. For each of the c inequalities (rows) from the conclusion, Lemma 6 gives one coefficient per each of the $k \cdot (c+d)$ assumptions. In total, we have $kc(c+d)$ coefficients, and we can arrange them as matrices $V_1, \ldots, V_k \in \mathbb{Q}_+^{c \times d}, W_1, \ldots, W_k \in \mathbb{Q}_+^{c \times c}$ and get

$$V_1 + W_1 C = CF_1 \wedge \ldots \wedge V_k + W_k C = CF_k \wedge \sum_i W_i B \leq CF_0 + B.$$

Since $V_i \geq 0$ we can simplify the equations to inequalities, obtaining

▶ **Lemma 7.** *The function $(x_1, \ldots, x_k) \mapsto F_0 + \sum_i F_i x_i$ with $F_i \geq 0$ maps $D^k \to D$ if and only if there exist $W_i \in \mathbb{Q}_+^{c \times c}$ with*

$$W_1 C \leq CF_1 \wedge \ldots \wedge W_k C \leq CF_k \wedge \sum_i W_i B \leq CF_0 + B.$$

Now we consider compatibility of the interpretation A with a rewrite rule (l, r) with $|\mathsf{Var}(l) \cup \mathsf{Var}(r)| = k$. Then the difference of interpretations $[l]_A - [r]_A$ is a linear function $\Delta : (x_1, \ldots, x_k) \mapsto \Delta_0 + \sum_i \Delta_i x_i$. When $x_i \in D$, we want $\Delta(x_1, \ldots, x_k) > 0$ or ≥ 0 (strict or weak compatibility). So, the conclusion (d rows) is

$$\begin{pmatrix} \Delta_1 & \Delta_2 & \ldots & \Delta_k & \Delta_0 \end{pmatrix},$$

resp. Δ_0' in the last component, where Δ_0' is obtained from Δ_0 by decreasing the first component by 1. Again by Lemma 6, there are coefficients $T_i, U_i \in \mathbb{Q}_+^{d \times c}$ with

$$T_1 + U_1 C = \Delta_1 \wedge \ldots \wedge T_k + U_k C = \Delta_k \wedge \sum_i U_i B \leq \Delta_0,$$

and we simplify (since $T_i \geq 0$) all equations to inequalities, and obtain

▶ **Lemma 8.** *A polyhedral interpretation A is strictly (weakly, respectively) compatible with a rewrite rule (l, r) with $|\mathsf{Var}(l) \cup \mathsf{Var}(r)| = k$ and $([l] - [r])(x_1, \ldots, x_k) = \Delta_0 + \sum_i \Delta_i x_i$ if and only if there exist matrices $U_i \in \mathbb{Q}_+^{d \times c}$ such that*

$$U_1 C \leq \Delta_1 \wedge \ldots \wedge U_k C < \Delta_k \wedge \sum_i U_i B \leq \Delta_0 \quad (\leq \Delta_0, \textit{resp.})$$

4 Certificates for Polyhedral Interpretations

We use one direction of Lemmata 7 and 8 to define certificates for properties of polyhedral interpretations.

▶ **Definition 9.** A *domain certificate* for a polyhedral interpretation consists of

- a vector $n \in \mathbb{Q}_+^d$ which is called valid if $Cn + B \geq 0$.
- for each $(f, k) \in \Sigma$ with $[f](x_1, \dots, x_k) = F_0 + \sum_i F_i x_i$, matrices $W_1, \dots W_k \in \mathbb{Q}_+^{c \times c}$ which are called valid if
  - $\forall 1 \leq i \leq k : CF_i \geq W_i C$ and $CF_0 + B \geq (\sum_i W_i)B$

▶ **Observation 10.** *The domain certificate contains $d + \|\Sigma\| \cdot c^2$ unknowns. Validity of the domain certificate can be checked with $1 + |\Sigma|$ matrix multiplications in $(c \times d) \cdot (d \times 1)$, $|\Sigma|$ matrix multiplications in $(c \times c) \cdot (c \times 1)$, $\|\Sigma\|$ matrix multiplications in $(c \times d) \cdot (d \times d)$, $|\Sigma|$ matrix multiplications in $(c \times c) \cdot (c \times d)$.*

There are also additions and comparisons, but their cost is dominated by multiplication.

▶ **Example 11** (Example 4 continued). We take $n = \begin{pmatrix}0 & 0 & 1\end{pmatrix}^T$ which is valid since $Cn+B = \begin{pmatrix}0 & -1 & 1\end{pmatrix}\begin{pmatrix}0 & 0 & 1\end{pmatrix}^T - 1 = 0$, and for both f and g, the choice $W_1 = (0)$ is valid since $CF_1 \geq 0$ and $CG_1 \geq 0$ and $CF_0 + B = CG_0 + B = 0$.

▶ **Lemma 12.** *The following statements are equivalent:*

- *polyhedral interpretation A has a valid domain certificate,*
- *D_A is non-empty, and for each $(f, k) \in \Sigma$, the function $[f]_A$ maps D_A^k into D_A.*

Proof. This follows from Lemma 7. Additionally, we give an explicit computation that shows one direction of the equivalence: For $y = [f](x_1, \dots, x_k) = F_0 + \sum_i F_i x_i$, we have $y \in D$ by the chain of inequalities $Cy + B = C(F_0 + \sum F_i x_i) + B = CF_0 + \sum_i CF_i x_i + B \geq CF_0 + \sum_i W_i C x_i + B \geq CF_0 - \sum W_i B + B \geq 0$. ◀

▶ **Definition 13.** A *compatibility certificate* for polyhedral interpretation A w.r.t. rewriting system R contains, for each rule $(l, r) \in R$ with $|\mathsf{Var}(l) \cup \mathsf{Var}(r)| = k$ and $([l]_A - [r]_A)(x_1, \dots, x_k) = \Delta_0 + \sum_i \Delta_i x_i$, matrices $U_1, \dots, U_k \in \mathbb{Q}_+^{d \times c}$, which are called valid if

- $\forall i : \Delta_i \geq U_i C$ and
- - $\Delta_0 \geq \sum_i U_i B$ (then the certificate is called *weak* for that rule)
  - or $\Delta_0 > \sum_i U_i B$ (then the certificate is called *strict* for that rule)

For the following, we need notation $\|R\| = \sum\{|\mathsf{Var}(l) \cup \mathsf{Var}(r)| \mid (l, r) \in R\}$.

▶ **Observation 14.** *The compatibility certificate contains $\|R\| \cdot d \cdot c$ unknowns. Validity of the compatibility certificate can be checked with $\|R\|$ matrix multiplications in $(d \times c) \cdot (c \times d)$, and $|R|$ matrix multiplications in $(d \times c) \cdot (c \times 1)$, assuming Δ_i are already given.*

▶ **Example 15** (Example 4 continued). For rule (fg, ff), we compute

$$[fg](x) = \begin{pmatrix}1 & 0 & 2\\0 & 0 & 0\\0 & 0 & 1\end{pmatrix} x + \begin{pmatrix}1\\0\\2\end{pmatrix}, \quad [ff](x) = \begin{pmatrix}1 & 2 & 0\\0 & 0 & 0\\0 & 0 & 1\end{pmatrix} x + \begin{pmatrix}2\\0\\2\end{pmatrix}, \quad \Delta^{(fg,ff)} = \begin{pmatrix}0 & -2 & 2\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} x + \begin{pmatrix}-1\\0\\0\end{pmatrix}.$$

A valid strict certificate for this rule is $U_1^{(fg,ff)} = \begin{pmatrix}2 & 0 & 0\end{pmatrix}^T$, since

$$U_1 C = \begin{pmatrix}2\\0\\0\end{pmatrix}\begin{pmatrix}0 & -1 & 1\end{pmatrix} = \begin{pmatrix}0 & -2 & 2\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \leq \Delta_1, \; U_1 B = \begin{pmatrix}2\\0\\0\end{pmatrix}(-1) = \begin{pmatrix}-2\\0\\0\end{pmatrix} < \Delta_0$$

For rule (gf, gg), we compute

$$[gf](x) = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, \quad [gg](x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}, \quad \Delta^{(gf,gg)} = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

and a valid strict certificate is $U_1^{(gf,gg)} = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T$.

▶ **Lemma 16.** *These statements are equivalent:*

- *polyhedral interpretation A has a valid compatibility certificate w.r.t. rewrite system R*
- *A is strictly compatible with the rules of R for which the certificate is strict, and weakly compatible with the rules for which the certificate is weak.*

Proof. This follows from Lemma 8. Additionally, we give an explicit computation for one direction. For $(l, r) \in R$, we have $([l]_A - [r]_A)(x_1, \ldots, x_k) = \Delta_0 + \sum_i \Delta_i x_i \geq \Delta_0 + \sum_i U_i C x_i \geq \Delta_0 - \sum_i U_i B$ which is ≥ 0 or > 0. ◀

From previous observations, and assuming that matrix multiplication in $(a \times b) \cdot (b \times c)$ can be done with $O(a \cdot b \cdot c)$ elementary operations, we obtain the following, which is the basis for our implementation, see Section 9.

▶ **Theorem 17.** *The validity of the certificate of a polyhedral interpretation with domain dimension d and constraint dimension c for a rewrite system R over signature Σ is decidable. The certificate can be represented by $d + \|\Sigma\|c^2 + \|R\|cd$ unknowns and $O(\|\Sigma\|cd^2 + |\Sigma|c^2 d + \|R\|cd^2 + |R|cd)$ elementary constraints.*

5 Polyhedral Interpretations for Termination and Complexity

Polyhedral interpretations can be used for proofs of termination:

▶ **Theorem 18.** *If a polyhedral interpretation has a valid domain certificate, and a strict compatibility certificate for a rewrite system R, and a weakly compatibility certificate for a rewrite system S, and the underlying interpretation is monotone, then R is terminating relative to S.*

Proof. The polyhedral interpretation defines a well-founded monotone algebra on a subset of $(\mathbb{N}^d, >)$ that is compatible with $\to_R / \to_S$. ◀

Compared to the standard matrix method, we kept the order, restricted the domain, and changed the test for compatibility: we can now use properties of the polyhedral domain, and thus ease the requirement of comparing coefficients of $[l]_A > [r]_A$ point-wise.

▶ **Example 19.** [1] We apply the method to problem `Ex16_Luc06_C` from the TPDB [21].

[1] `http://nfa.imn.htwk-leipzig.de/termcomp/show_job_pair/41269410`

$$\begin{aligned}
active(f(X,X)) \to mark(f(a,b)), \quad active(b) \to mark(a),\\
active(f(X1,X2)) \to f(active(X1),X2), \quad f(mark(X1),X2) \to mark(f(X1,X2)),\\
proper(f(X1,X2)) \to f(proper(X1),proper(X2)),\\
proper(a) \to ok(a), \quad proper(b) \to ok(b),\\
f(ok(X1),ok(X2)) \to ok(f(X1,X2)),\\
top(mark(X)) \to top(proper(X)), \quad top(ok(X)) \to top(active(X))
\end{aligned}$$

we remove $active(b) \to mark(a)$ and then use interpretation

$$\begin{aligned}
mark \mapsto \begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}1&0\\0&1\end{pmatrix}\cdot x_1, f \mapsto \begin{pmatrix}1\\2\end{pmatrix} + \begin{pmatrix}1&3\\0&0\end{pmatrix}\cdot x_1 + \begin{pmatrix}1&0\\0&0\end{pmatrix}\cdot x_2,\\
a \mapsto \begin{pmatrix}1\\0\end{pmatrix}, b \mapsto \begin{pmatrix}0\\1\end{pmatrix}, ok \mapsto \begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}1&0\\0&1\end{pmatrix}\cdot x_1, top \mapsto \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}1&2\\1&1\end{pmatrix}\cdot x_1,\\
active \mapsto \begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}1&0\\0&1\end{pmatrix}\cdot x_1, proper \mapsto \begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}1&0\\0&1\end{pmatrix}\cdot x_1
\end{aligned}$$

on a domain restricted by $\begin{pmatrix}-1\end{pmatrix} + \begin{pmatrix}1&1\end{pmatrix}\cdot x \geq 0$. Equivalently, $x_1 + x_2 \geq 1$, so just $(0,0)^T$ is excluded from the domain. Rule $active(f(X,X)) \to mark(f(a,b))$ is interpreted by

$$[\text{lhs}] = \begin{pmatrix}1\\2\end{pmatrix} + \begin{pmatrix}2&3\\0&0\end{pmatrix}\cdot x_1, \quad [\text{rhs}] = \begin{pmatrix}2\\2\end{pmatrix} + \begin{pmatrix}0&0\\0&0\end{pmatrix}\cdot x_1,$$

Note the absolute parts are not decreasing: $(1\ 2)^T \not> (2\ 2)^T$. This rule has a strong compatibility certificate $(2\ 0)^T$. In effect, we add twice the domain constraint, to prove the decrease in the first component. Starting from the right-hand side: $2 \leq 2+2(-1+x_{11}+x_{12}) = x_{11}+x_{12} < 1+2x_{11}+3x_{12}$. Then other rules can be removed by the standard matrix method.

This termination problem was solved in the 2014 competition only by AProVE [12], using back-transformation to CSR `QTRSToCSRProof`. [2]

The previous example suggests that polyhedral interpretations are strictly more powerful than standard matrix interpretations (even when these are combined with other methods), but we currently have no proof. At least we can be sure that they are not less powerful:

▶ **Observation 20.**
- *For any domain and constraint dimension: A polyhedral interpretation with domain constraint $B = 0, C = 0$ is a standard matrix interpretation.*
- *A polyhedral interpretation with constraint dimension 0 is a standard matrix interpretation.*

Proof. First part: take $n = 0, W_i = 0, U_i = 0$ and verify that the compatibility condition reduces to $\Delta_0 \geq 0 (> 0, \text{resp.})$ and $\Delta_i \geq 0$. Second part: The matrices in the domain certificate have extension 0×0, the matrices in the compatibility certificate have extension $d \times 0$, so they are zero matrices, and the first part applies. ◀

Even if we do have constraints, we can ignore them, to obtain a statement on derivational complexity. Recall that the height function dc_A of a well-founded monotone Σ-algebra A on $(D_A, >_A)$ is the function $s \mapsto \sup\{\mathsf{dc}_{>_A}([t]_A) \mid t \in \mathsf{Term}(\Sigma), |t| \leq s\}$.

[2] `http://nfa.imn.htwk-leipzig.de/termcomp/display_proof/26921465`

▶ **Lemma 21.** *If a polyhedral interpretation A is monotone and strictly compatible with a rewrite system R, then dc_R is bounded by the height dc_A of the matrix interpretation that underlies A.*

Proof. Each $\to_R$-chain is mapped to a $>$-chain in the polyhedral domain D, which is also a $>$-chain in $\mathbb{N}^d$. ◀

We mention two consequences.

The original matrix method is limited because matrix products grow at most exponentially: with matrix interpretations, it is impossible to reduce a termination problem by "removing a rule" that is used more than exponentially often. E.g., no matrix interpretation can remove a rule from $\{ab \to bca, cb \to bbc\}$ [13], and polyhedral constraints will not change that.

If we have a polyhedral interpretation A that is compatible with R and where the underlying matrix interpretation grows polynomially only, then we have a proof that dc_R is polynomially bounded. The introductory Example 1 already applies this. We will show in the next section that we can do better in some cases, by not ignoring the information in the constraints.

6 Improving Polynomial Growth Bounds

We show that polyhedral constraints can serve to lower a bound for polynomial growth of a matrix interpretation.

Recall that for each component of a vector valued interpretation we can assign a degree of growth, and the degree of the interpretation is the degree of the first component.

If the interpretation uses upper triangular matrices (of dimension d), the degree of the i-th component is at most $d + 1 - i$, and there is a refinement where the degree can be reduced further if all matrices have zero diagonal entries at index (i, i).

Now polyhedral constraints for triangular matrices in some cases bound the i-th component from above by some positive linear combination of components with higher indices, that is, of lower degree.

As a special case, the very last component could be bounded by a constant, as in the following example.

▶ **Example 22.** `TRS/secret06/jambox/5` [3] The rewrite system

$$\{a(a(y,0),0) \to y, c(c(y)) \to y, c(a(c(c(y)),x)) \to a(c(c(c(a(x,0)))),y)\}$$

has a compatible polyhedral interpretation that uses upper triangular matrices

$$0 \mapsto \begin{pmatrix}0\\0\\0\\0\end{pmatrix}, c \mapsto \begin{pmatrix}0\\0\\0\\2\end{pmatrix} + \begin{pmatrix}1&1&0&1\\0&1&2&0\\0&0&1&1\\0&0&0&0\end{pmatrix} \cdot x_1, a \mapsto \begin{pmatrix}1\\0\\0\\0\end{pmatrix} + \begin{pmatrix}1&0&0&0\\0&1&0&1\\0&0&1&0\\0&0&0&1\end{pmatrix} \cdot x_1 + \begin{pmatrix}1&2&2&2\\0&1&4&4\\0&0&1&1\\0&0&0&0\end{pmatrix} \cdot x_2$$

with domain restriction $\begin{pmatrix}2\end{pmatrix} + \begin{pmatrix}0&0&0&-1\end{pmatrix} \cdot x \geq 0$. The restriction means that $x_4 \leq 2$. (By inspecting the interpretation, the domain for x_4 is found to be $\{0, 2\}$.) There are 4 non-zero positions on the main diagonal, so the naive degree bound is 4. The domain constraint says

[3] `http://nfa.imn.htwk-leipzig.de/termcomp/show_job_pair/40307443`

that x_4 is bounded by 2, so the degree of x_3 is at most linear, x_2 at most quadratic, and x_1 at most cubic.

The states $3, 4$ of the underlying automaton are in fact an unambiguous component, so degree 3 would have been detected by the method from [28], which is however more expensive to implement.

This example was not solved in the 2014 complexity competition.

▶ **Example 23.** For system `ExProp7_Luc06_GM` (not solved in 2014 complexity competition), we find a compatible polyhedral upper triangular interpretation [4] with constraint $\begin{pmatrix}0 & 0 & -1 & 1\end{pmatrix} x + 2 \geq 0$. That is $x_4 + 2 \geq x_3$, so the degree of x_3 is linear (not quadratic), and this reduces the degree estimate for the interpretation as a whole.

7 Polyhedral Constraints and the Dependency Pair Method

The dependency pair (DP) method [2] can use reduction pairs that come from matrix interpretations [11]. We briefly recap the notation. Given Σ, the marked signature $\Sigma^\#$ is $\{(f^\#, k) \mid (f, k) \in \Sigma\}$. We use sort symbols $\{O, \#\}$ and say that $(f, k) \in \Sigma$ has type $O^k \to O$, while $f^\#$ has type $O^k \to \#$. For $t = f(t_1, \ldots, t_k) \in \mathsf{Term}(\Sigma, V)$, we write $t^\#$ for $f^\#(t_1, \ldots, t_k) \in \mathsf{Term}(\Sigma \cup \Sigma^\#, V)$. The root symbol of a term t is $\mathsf{root}(t)$. The set of defined symbols of a rewrite system R over Σ is $\mathsf{Def}(R) = \{\mathsf{root}(l) \mid (l, r) \in R\}$. The dependency pairs of R are $\mathsf{DP}(R) = \{(l^\#, s^\#) \mid (l, r) \in R, \mathsf{root}(s) \in \mathsf{Def}(R), s \trianglelefteq r, s \ntrianglelefteq l\}$. Termination of R is then proved by a reduction pair $(>, \geq)$ where $\mathsf{DP}(R) \subseteq >$ and $R \subseteq \geq$.

In this context, a d-dimensional matrix interpretation defines an extended monotone algebra by interpreting sort O by $(\mathbb{N}^d, \geq)$, sort $\#$ by $(\mathbb{N}^1, >)$, and function symbols by multilinear functions (respecting the sorts) as before.

For this basic version of the DP method with matrix interpretations, we can apply polyhedral constraints with the following (inessential) modifications:

- We do not need strict monotonicity, so the top-left entries of matrices are unconstrained.
- We restrict the domain for sort O only (not $\#$). That is, we need domain certificates only for symbols from Σ (not $\Sigma^\#$).
- Compatibility certificates are needed for all rules. For rules from R, this is done as before, and for rules from $\mathsf{DP}(R)$, the target domain is $\mathbb{N}^1$, so the matrices U_i in the compatibility certificates have extension $(1 \times c)$.

The DP method allows for many enhancements, which we do not discuss here, but use in examples.

▶ **Example 24.** We consider the termination problem `TRS_Standard/Various_04/11`

$$\{f(0, 1, x) \to f(h(x), h(x), x), h(0) \to 0, h(g(x, y)) \to y\}$$

[4] `http://nfa.imn.htwk-leipzig.de/termcomp/show_job_pair/40858473`

After DP transformation, we look at the SCC that contains $\{f^{\#}(0,1,x) \to f^{\#}(h(x),h(x),x)\}$, and apply the matrix interpretation [5]

$$0 \mapsto \begin{pmatrix}1\\1\end{pmatrix}, 1 \mapsto \begin{pmatrix}0\\2\end{pmatrix}, h \mapsto \begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} \cdot x_1, g \mapsto \begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0 & 0\\0 & 0\end{pmatrix} \cdot x_1 + \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} \cdot x_2,$$

$$f \mapsto \begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}0 & 0\\0 & 0\end{pmatrix} \cdot x_1 + \begin{pmatrix}0 & 0\\0 & 0\end{pmatrix} \cdot x_2 + \begin{pmatrix}0 & 0\\0 & 0\end{pmatrix} \cdot x_3,$$

$$f^{\#} \mapsto \begin{pmatrix}0\end{pmatrix} + \begin{pmatrix}1 & 0\end{pmatrix} \cdot x_1 + \begin{pmatrix}0 & 1\end{pmatrix} \cdot x_2 + \begin{pmatrix}1 & 1\end{pmatrix} \cdot x_3$$

with constraint $\begin{pmatrix}2\end{pmatrix} + \begin{pmatrix}-1 & -1\end{pmatrix} \cdot x \geq 0$. The interpretation of $f^{\#}(0,1,x) \to f^{\#}(h(x),h(x),x)$ is

$$[\text{lhs}] = \begin{pmatrix}3\end{pmatrix} + \begin{pmatrix}1 & 1\end{pmatrix} \cdot x_1, \quad [\text{rhs}] = \begin{pmatrix}0\end{pmatrix} + \begin{pmatrix}2 & 2\end{pmatrix} \cdot x_1.$$

We can verify that adding the domain constraint to the value of the right-hand side gives $(2) + (1\ 1) \cdot x_1$ which is in the proper point-wise relation to the left-hand side. This problem was solved in the 2014 termination competition only by Mu-Term [16] and AProVE [12], using innermost rewriting and narrowing. [6]

▶ **Example 25.** For `TRS_Standard/Endrullis_06/pair2simple2`

$$\{p(a(x_0), p(a(a(a(x_1))), x_2)) \to p(a(x_2), p(a(a(b(x_0))), x_2))\}$$

we find an interpretation [7]

$$b \mapsto \begin{pmatrix}0\\0\\0\end{pmatrix} + \begin{pmatrix}0 & 0 & 1\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \cdot x_1, a \mapsto \begin{pmatrix}0\\0\\1\end{pmatrix} + \begin{pmatrix}0 & 1 & 0\\0 & 0 & 3\\0 & 0 & 0\end{pmatrix} \cdot x_1,$$

$$p \mapsto \begin{pmatrix}0\\0\\0\end{pmatrix} + \begin{pmatrix}1 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \cdot x_1 + \begin{pmatrix}1 & 1 & 0\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \cdot x_2,$$

$$p^{\#} \mapsto \begin{pmatrix}0\end{pmatrix} + \begin{pmatrix}0 & 0 & 0\end{pmatrix} \cdot x_1 + \begin{pmatrix}1 & 0 & 1\end{pmatrix} \cdot x_2$$

with polyhedral constraint dimension two: $\begin{pmatrix}2\\1\end{pmatrix} + \begin{pmatrix}0 & -1 & 1\\0 & 0 & -1\end{pmatrix} \cdot x \geq 0.$

Rule $p(a(x_0), p(a(a(a(x_1))), x_2)) \to p(a(x_2), p(a(a(b(x_0))), x_2))$ is interpreted by

$$[\text{lhs}] = \begin{pmatrix}3\\0\\0\end{pmatrix} + \begin{pmatrix}0 & 1 & 0\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \cdot x_1 + \begin{pmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \cdot x_2 + \begin{pmatrix}1 & 1 & 0\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \cdot x_3$$

$$[\text{rhs}] = \begin{pmatrix}0\\0\\0\end{pmatrix} + \begin{pmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \cdot x_1 + \begin{pmatrix}0 & 0 & 0\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \cdot x_2 + \begin{pmatrix}1 & 2 & 0\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix} \cdot x_3,$$

and the (weak) compatibility certificate is $\begin{pmatrix}0 & 0\\0 & 0\\0 & 0\end{pmatrix}, \begin{pmatrix}0 & 0\\0 & 0\\0 & 0\end{pmatrix}, \begin{pmatrix}1 & 1\\0 & 0\\0 & 0\end{pmatrix}$

This problem was solved in the 2014 competition with recursive path order [8] and it seems there is no compatible standard matrix interpretation.

---

[5] `http://nfa.imn.htwk-leipzig.de/termcomp/show_job_pair/40515447`
[6] `http://nfa.imn.htwk-leipzig.de/termcomp/show_job_pair/26924062`
[7] `http://nfa.imn.htwk-leipzig.de/termcomp/show_job_pair/40849733`
[8] `http://nfa.imn.htwk-leipzig.de/termcomp/show_job_pair/26919296`

8 Polyhedral Constraints and the DP Method with Usable Rules

The DP method with the "usable rules" extension [2] requires the following change. It is required that the reduction pair is C_ϵ-compatible (for a fresh function symbol C, we need $[C](x,y) \geq x \wedge [C](x,y) \geq y$.) For a reduction pair that comes from an interpretation, this means that its domain must allow to construct least upper bounds.

Let us make explicit how this works for the standard matrix method: the domain is $\mathbb{N}^d$, the (weak) order is component-wise $\geq$, the least upper bound $\sup(x,y)$ of $x = (x_1,\dots,x_d), y = (y_1,\dots,y_d)$ is $(\max(x_1,y_1),\dots,\max(x_n,y_n))$, so $\sup(x,y) \in \mathbb{N}^d$ trivially.

Now consider some polyhedral domain $D \subseteq \mathbb{N}^d$. The order we use on D is exactly the order on $\mathbb{N}^d$ as before, so the least upper bound is the same as well. But it is not always true that $x, y \in D$ implies $\sup(x,y) \in D$.

▶ **Example 26.** In dimension 2, consider the constraint $x_1 + x_2 \leq 1$. This means $D = \{(0,0),(0,1),(1,0)\}$. Then $\sup((0,1),(1,0)) = (1,1) \notin D$.

Indeed we would obtain erroneous termination statements when using the "usable rules" extension with polyhedral constraints on domains that are not sup-closed. [9]

We now give a sufficient condition for a polyhedral domain to allow sup.

▶ **Theorem 27.** *Let $C \in \mathbb{Q}^{c\times d}, B \in \mathbb{Q}^{c\times 1}$ describe a domain $D = \{x \mid x \geq 0, Cx + B \geq 0\} \subseteq \mathbb{Q}^d$. If each row of C contains at most one negative entry, then $x, y \in D$ implies $\sup(x,y) \in D$.*

Proof. We analyze the i-th constraint, given by row $c = C_i$, and entry $b = B_i$. We need to show $c \cdot \sup(x,y) + b \geq 0$. We can write $c = c^+ + c^-$ where $c^+ = \sup(c,0)$ and $c^- = \inf(c,0)$. (all entries in c^+ are ≥ 0, all entries in c^- are ≤ 0). Multiplication by c^+ is monotone: if $x \leq z$, then $c^+x \leq c^+z$. If c has no negative entry, then $c = c^+$, so multiplication by c is monotone, and we have $c\cdot\sup(x,y)+b \geq cx+b \geq 0$. Assume c has one negative entry $c_k < 0$. Without loss of generality, we have $x_k \geq y_k$ (if not, swap x with y). Then $c^-\cdot\sup(x,y) = c^-x$. We have $c\cdot\sup(x,y)+b = c^+\cdot\sup(x,y)+c^-\cdot\sup(x,y)+b \geq c^+x+c^-x+b = cx+b \geq 0$. ◀

The condition in Theorem 27 is easily implemented in a constraint program.

9 Implementation and Experiments

We extended to the implementation of matrix interpretations in `matchbox` [29] by adding polyhedral constraints. The constraint system already has unknowns for the interpretation, and we added unknowns for the domain constraint and certificate, and for the compatibility certificates. The constraint program already computes the interpretation of rules (with compression) [5], and we added the validity constraints for domain and compatibility.

To prove termination of a rewrite system, we have (as usual) one parameter $d \in \mathbb{N}$, for the dimension of the interpretation domain, and now an extra parameter $c \in \mathbb{N}$: the number of inequalities (the height of the C matrix). We found that already $c = 1$ is often helpful, see most examples in this paper.

According to Sections 3 and 4, unknowns should be from $\mathbb{Q}$ for the domain constraint, and $\mathbb{Q}_+$ for certificates. Since we don't know of a competitive constraint solver over $\mathbb{Q}$, we restrict to $\mathbb{Z}$ for the domain constraint, and $\mathbb{N}$ for certificates. Experiments suggest that

[9] `http://nfa.imn.htwk-leipzig.de/termcomp/show_job_pair/40422582`

most domain constraints use small numbers, so we further restrict to $\{-1, 0, 1\}$ for C (not for B).

The constraint system consists of (in)equalities between polynomials, so it is expressible in the QFNRA (QFNIA, resp.) logic [4]. With a bit-blasting approach in mind, we can also use QFBV (bit vectors). Our implementation allows us to choose between Boolector [6] as a QFBV-solver, or built-in bit-blasting, and then MiniSat [10] as a SAT-solver.

The following data is typical of how polyhedral constraints increase the size of the constraint systems:

▶ **Example 28** (Example 22 continued)**.** For `TRS/secret06/jambox/5`, with domain dimension 4, and bit width 4, matchbox' built-in bit-blaster was applied. The number of variables/clauses is: for constraint dimension 0 (the original matrix method): 31614/40256, for constraint dimension 1: 39867/67064.

We have two remarks on BV-solving/bit-blasting:

Overflow is forbidden in our context, but allowed in the QFBV standard. So each arithmetical operation (`add`, `mul`) is immediately followed by computing the overflow and asserting that it is false. We use functions `saddo`, `smulo` provided in Boolector's API.

We need signed numbers. Among the unknowns, just C and B may contain negative numbers, but signed numbers will propagate into the validity constraints. We note that while we have additions of signed numbers, all multiplications have at most one signed factor.

In both cases (no overflow, some signs are known), a constraint solver could exploit this information statically. We especially think that "non-overflowing arithmetic" would be a useful addition to the QFBV-standard.

We were running our implementation on the termination and complexity problems of the 2014 termination competition (TPDB version 8). The main purpose was to extract interesting examples, used in this paper. These examples show that there are several cases where polyhedral constraints allow a matrix termination proof where none was given in the last competition, or only proofs that use other methods.

We checked the effect that polyhedral constraints have when added to a base version of matchbox with arctic and natural matrix interpretations. We observe different behaviour in less than 10 % of the benchmarks.

We also compared Boolector and bit-blasting/MiniSat back-ends. Our conclusion is that Boolector wins by a small margin.

A web page with that presents our experimental data is provided. [10]

10 Discussion

Related work. Polyhedral constraints for interpretations in termination proofs where first suggested by Lucas and Meseguer [17]. Our contribution is to provide an actual implementation that handles the case where interpretation and domain constraints are unknown, and extensions (to complexity analysis, dependency pairs, and usable rules).

We mentioned that polyhedral analysis of imperative programs, especially loops, is a standard method. The difference to analysis of rewriting systems is: the semantics of the imperative program is literally given by the numerical values and operations appearing in the program text (e.g., in `int y = 2*x+1`, the symbol 2 denotes the number 2, and the symbol `+` denotes addition). For rewrite systems, a priori there is no semantics, so it has

[10] `http://www.imn.htwk-leipzig.de/~waldmann/etc/polyhedral/`

to be determined during the analysis (e.g., it will be defined via an interpretation). This means that for polyhedral domains for rewrite systems, we cannot use directly the methods developed for imperative programs.

Certification. It seems straightforward (in principle) to integrate polyhedral domains for matrix interpretations into the CeTA certification framework [23]. To check validity of domain and compatibility certificates, we just need to verify the calculations from Section 4, while a certified proof of Farkas' Lemma (Section 3) is not needed.

Challenges in Rewriting. There are two long-standing challenges: find a matrix interpretation that gives a tight complexity bound for $\{a^2b^2 \to b^3a^3\}$ (z001), and for $\{a^2 \to bc, b^2 \to ac, c^2 \to ab\}$ (z086). For both rewrite systems, matrix interpretations with exponential growth are known, while the growth of rewrite sequences is known to be polynomial (z001 by matchbounds, z086 by a manual proof [1]). Can we prove polynomial derivational complexity via matrix interpretations on a polyhedral domain? So far, we did not succeed—using several days of CPU time.

Extensions: Order. It may be interesting to analyze different orders on polyhedral domains, as Neurauter et al. [20] did for the full standard domain $\mathbb{N}^d$. We might get more termination proofs, or better complexity bounds. It is to be expected that monotonicity, which is now easy (top left coefficient ≥ 1), needs to be replaced with something more elaborate, that requires a certificate.

Extensions: Negative Coefficients. Can we allow negative coefficients in interpretations (in $F_0, F_1, \ldots, F_k$)? We then need to make sure that $x \geq 0$ is respected (by extra domain certificates), and require additional "monotonicity certificates".

Extensions: Domain. We can perhaps even drop the $x \geq 0$ restriction, This implies changes in other certificates, and requires an extra "well-foundedness certificate" that shows that values for the first component of interpretations are bounded from below.

Extensions: Semiring. Another direction for extension is to choose a different underlying semiring, e.g., arctic or tropical, and apply results from tropical linear algebra. At least in principle, it is clear what to do: a polyhedral domain [9] is described as $\{x \mid A_1 \cdot x + b_1 \leq x \leq A_2 \cdot x + b_2\}$ (since addition is not invertible, we cannot move the right-hand-side x to the left) and certificates must be constructed accordingly.

Acknowledgments. I am grateful to Alfons Geser, Dieter Hofbauer, and anonymous reviewers for discussions and comments, and to René Thiemann and Harald Zankl for help with their software. The Starexec platform [24] and Starexec-Presenter [27] software proved to be very helpful for running experiments and analyzing data. Armin Biere made available Boolector [6] for use as an alternative solver back-end.

References

1 Sergei I. Adian. Upper bound on the derivational complexity in some word rewriting system. *Doklady Mathematics*, 80(2):679–683, 2009.

2 Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs. *Theor. Comput. Sci.*, 236(1-2):133–178, 2000.

3 Roberto Bagnara, Patricia M. Hill, and Enea Zaffanella. The Parma Polyhedra Library: Toward a Complete Set of Numerical Abstractions for the Analysis and Verification of Hardware and Software Systems. *CoRR*, abs/cs/0612085, 2006.

4 Clark Barrett, Aaron Stump, and Cesare Tinelli. The Satisfiability Modulo Theories Library (SMT-LIB). `http://smt-lib.org/`, 2010–.

5 Alexander Bau, Markus Lohrey, Eric Nöth, and Johannes Waldmann. Compression of rewriting systems for termination analysis. In Femke van Raamsdonk, editor, *24th International Conference on Rewriting Techniques and Applications, RTA 2013, June 24-26, 2013, Eindhoven, The Netherlands*, volume 21 of *LIPIcs*, pages 97–112. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

6 Armin Biere and Robert Brummayer. Boolector. `http://fmv.jku.at/boolector/`, 2008–.

7 Ahlem Ben Cherifa and Pierre Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Sci. Comput. Program.*, 9(2):137–159, 1987.

8 Michael Codish, Yoav Fekete, Carsten Fuhs, Jürgen Giesl, and Johannes Waldmann. Exotic semi-ring constraints. In Pascal Fontaine and Amit Goel, editors, *10th International Workshop on Satisfiability Modulo Theories, SMT 2012, Manchester, UK, June 30 - July 1, 2012*, volume 20 of *EPiC Series*, pages 88–97. EasyChair, 2012.

9 Mike Develin and Josephine Yu. Tropical polytopes and cellular resolutions. *Experimental Mathematics*, 16(3):277–291, 2007.

10 Niklas Een and Niklas Sörensson. Minisat. http://minisat.se/, 2003–.

11 Jörg Endrullis, Johannes Waldmann, and Hans Zantema. Matrix interpretations for proving termination of term rewriting. *J. Autom. Reasoning*, 40(2-3):195–220, 2008.

12 Jürgen Giesl, Marc Brockschmidt, Fabian Emmes, Florian Frohn, Carsten Fuhs, Carsten Otto, Martin Plücker, Peter Schneider-Kamp, Thomas Ströder, Stephanie Swiderski, and René Thiemann. Proving Termination of Programs Automatically with AProVE. In Stéphane Demri, Deepak Kapur, and Christoph Weidenbach, editors, *Automated Reasoning - 7th International Joint Conference, IJCAR 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 19-22, 2014. Proceedings*, volume 8562 of *Lecture Notes in Computer Science*, pages 184–191. Springer, 2014.

13 Dieter Hofbauer and Johannes Waldmann. Termination of string rewriting with matrix interpretations. In Frank Pfenning, editor, *Term Rewriting and Applications, 17th International Conference, RTA 2006, Seattle, WA, USA, August 12-14, 2006, Proceedings*, volume 4098 of *Lecture Notes in Computer Science*, pages 328–342. Springer, 2006.

14 Adam Koprowski and Johannes Waldmann. Max/plus tree automata for termination of term rewriting. *Acta Cybern.*, 19(2):357–392, 2009.

15 Martin Korp and Harald Zankl. Cat (complexity and termination). `http://cl-informatik.uibk.ac.at/software/cat/`, 2008–.

16 Salvador Lucas. mu-term: A tool for proving termination of context-sensitive rewriting. In Vincent van Oostrom, editor, *Rewriting Techniques and Applications, 15th International Conference, RTA 2004, Aachen, Germany, June 3-5, 2004, Proceedings*, volume 3091 of *Lecture Notes in Computer Science*, pages 200–209. Springer, 2004.

17 Salvador Lucas and José Meseguer. Models for logics and conditional constraints in automated proofs of termination. In Gonzalo A. Aranda-Corral, Jacques Calmet, and Francisco J. Martín-Mateos, editors, *Artificial Intelligence and Symbolic Computation - 12th International Conference, AISC 2014, Seville, Spain, December 11-13, 2014. Proceedings*, volume 8884 of *Lecture Notes in Computer Science*, pages 9–20. Springer, 2014.

18 Georg Moser, Andreas Schnabl, and Johannes Waldmann. Complexity analysis of term rewriting based on matrix and context dependent interpretations. In Ramesh Hariharan, Madhavan Mukund, and V. Vinay, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2008, December 9-11,*

2008, Bangalore, India, volume 2 of *LIPIcs*, pages 304–315. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2008.

19 Friedrich Neurauter and Aart Middeldorp. Revisiting matrix interpretations for proving termination of term rewriting. In Manfred Schmidt-Schauß, editor, *Proceedings of the 22nd International Conference on Rewriting Techniques and Applications, RTA 2011, May 30 - June 1, 2011, Novi Sad, Serbia*, volume 10 of *LIPIcs*, pages 251–266. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

20 Friedrich Neurauter, Harald Zankl, and Aart Middeldorp. Revisiting matrix interpretations for polynomial derivational complexity of term rewriting. In Christian G. Fermüller and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 17th International Conference, LPAR-17, Yogyakarta, Indonesia, October 10-15, 2010. Proceedings*, volume 6397 of *Lecture Notes in Computer Science*, pages 550–564. Springer, 2010.

21 Albert Rubio, Claude Marche, Hans Zantema, and René Thiemann. Termination problems data base (tpdb). `http://termination-portal.org/wiki/TPDB`, 2003–.

22 Christian Sternagel and Aart Middeldorp. Root-labeling. In Andrei Voronkov, editor, *Rewriting Techniques and Applications, 19th International Conference, RTA 2008, Hagenberg, Austria, July 15-17, 2008, Proceedings*, volume 5117 of *Lecture Notes in Computer Science*, pages 336–350. Springer, 2008.

23 Christian Sternagel, René Thiemann, Sarah Winkler, and Harald Zankl. CeTA - A Tool for Certified Termination Analysis. *CoRR*, abs/1208.1591, 2012.

24 Aaron Stump, Tyler Jensen, and Cesare Tinelli. Starexec. `https://www.starexec.org/`, 2013–.

25 Sergej N. Tschernikow. *Lineare Ungleichungen*. Berlin, 1971.

26 Aard Varks. Farkas' lemma. `http://aardvarks.bandcamp.com/album/farkas-lemma`, 1996.

27 Stefan von der Krone. Stareexec-presenter. `https://github.com/stefanvonderkrone/star-exec-presenter`, 2014–.

28 Johannes Waldmann. Polynomially bounded matrix interpretations. In Christopher Lynch, editor, *Proceedings of the 21st International Conference on Rewriting Techniques and Applications, RTA 2010, July 11-13, 2010, Edinburgh, Scottland, UK*, volume 6 of *LIPIcs*, pages 357–372. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.

29 Johannes Waldmann, Alexander Bau, and Eric Noeth. Matchbox termination prover. http://github.com/jwaldmann/matchbox/, 2014–.

30 Hans Zantema. Termination of term rewriting by interpretation. In Michaël Rusinowitch and Jean-Luc Remy, editors, *Conditional Term Rewriting Systems, Third International Workshop, CTRS-92, Pont-à-Mousson, France, July 8-10, 1992, Proceedings*, volume 656 of *Lecture Notes in Computer Science*, pages 155–167. Springer, 1992.