Improved Matrix interpretation ⋆

Pierre Courtieu, Gladys Gbedo, and Olivier Pons

CÉDRIC – CNAM, Paris, France

Abstract. We present a new technique to prove termination of Term Rewriting Systems, with full automation. A crucial task in this context is to find suitable well-founded orderings. A popular approach consists in interpreting terms into a domain equipped with an adequate well-founded ordering. In addition to the usual interpretations: natural numbers or polynomials over integer/rational numbers, the recently introduced matrix based interpretations have proved to be very efficient regarding termination of string rewriting and of term rewriting. In this spirit we propose to interpret terms as polynomials over integer matrices. Designed for term rewriting, our generalisation subsumes previous approaches allowing for more orderings without increasing the search space. Thus it performs better than the original version. Another advantage is that, interpreting terms to actual polynomials of matrices, it opens the way to matrix non linear interpretations. This result is implemented in the CiME3 rewriting toolkit.

1 Introduction

The property of termination, well-known to be undecidable, is fundamental in many aspects of computer science and logic. It is crucial in the proof of programs correctness, it underlies induction proofs, etc. Despite its non-decidability, many heuristics have been proposed to provide automation for termination proofs. In particular, many heuristics have been defined in the framework of term rewriting systems (TRS). All of them require, possibly after several transformations of the initial termination problem, to search a well-founded ordering satisfying some properties. Among the different kinds of orderings, polynomial interpretations [19, 4, 6] and recursive path ordering [8] are the most used.

More recently matrix interpretation introduced in the context of string rewriting [16] and adapted to term rewriting system by Endrullis et al. in [11] has proved to be very efficient. They interpret term into vectors associating to each symbol a linear mapping with matrix coefficients. We propose a generalization of this method interpreting term into matrix and associating to each symbol an actual matrix polynomial. Our generalization subsumes the previous methods and allows for more matrices and more orderings. In particular it allows for more systems to be proved to be terminating without increasing the bounds for coefficients or the size of matrices.

Due to the monotonicity requirement for interpretations, the original matrix interpretations are restricted to matrices with a strictly positive upper left coefficient, and the associated strict ordering only considers the upper coefficient on vectors. We propose

⋆ Work partially supported by A3PAT project of the French ANR (ANR-05-BLAN-0146-01).

weaker limitation still preserving monotonicity. We require for each matrices to have a fixed sub-matrix with no null columns. The strict ordering only consider coefficients corresponding to this sub-matrix. In this framework the original matrix interpretation is a particular case where the sub-matrix is reduced to the upper left coefficients.

Section 2 recalls preliminary notions on term rewriting systems, termination criteria, usual orderings and presents the matrix interpretation. It also introduces our model of presentation of termination proof as an inference tree [5]. Section 3 presents the extension we propose and the proof of its correctness. Section 4 describes the proof search and Section 5 presents severals examples. Section 6 illustrate the efficiency of our method on the *termination problems database (TPDB)* and show how it improves previous methods. Finally we present future work and conclude in Section 7.

2 Preliminaries

2.1 Term rewriting systems

We assume that the reader is familiar with basic concepts of term rewriting [9, 3] and termination. We recall the usual notions, and give our notations.

Terms— A *signature* Σ is a finite set of *symbols* with fixed arities. Let X be a countable set of *variables*; $T(\Sigma, X)$ denotes the set of finite *terms* on Σ and X. $\Lambda(t)$ is the symbol at the root position in term t. We write $t|_p$ for the subterm of t at position p and $t[u]_p$ for term t where $t|_p$ is replaced by u. *Substitutions* are mappings from variables to terms and $t\sigma$ denotes the application of a substitution σ to a term t.

Monotonicity— A function $f : D^n \rightarrow D$ on a domain D is *monotonic* with respect to a relation R on D iff $\forall d_1, d_2 \in D, \forall 1 \leq i \leq n : \forall a_1, \ldots a_{i-1}, a_{i+1}, \ldots, a_n \in D, d_1 \; R \; d_2 \Rightarrow f(a_1, \ldots, a_{i-1}, d_1, a_{i+1}, \ldots, a_n) \; R \; f(a_1, \ldots, a_{i-1}, d_2, a_{i+1}, \ldots, a_n)$. We say that a *relation on terms* R is monotonic if all function symbols are monotonic with respect to R.

Rewriting— A *term rewriting system* (TRS for short) over a signature Σ is a set S of *rewrite rules* $l \rightarrow r$ with $l, r \in T(\Sigma, X)$. In this work we only consider *finite* systems. A TRS S defines a monotonic relation $\rightarrow_S$ closed under substitution (aka a *rewrite relation*) in the following way: $s \rightarrow_S t$ (*s reduces to t*) if there is a position p such that $s|_p = l\sigma$ and $t = s[r\sigma]_p$ for a rule $l \rightarrow r \in S$ and a substitution σ. We shall omit systems and positions that are clear from the context. We denote the reflexive-transitive closure of a relation $\rightarrow$ by $\rightarrow^\star$. Symbols occurring at root position in the left-hand sides of rules in S are said to be *defined*, the others are said to be *constructors*. We denote $\rightarrow_1 . \rightarrow_2$ the relation defined by $x \rightarrow_1 . \rightarrow_2 y$ iff $\exists z, x \rightarrow_1 z \rightarrow_2 y$ where $\rightarrow_1$ and $\rightarrow_2$ are two relations.

Ordering— Termination proofs usually make use of orderings and *ordering pairs* [18]. We use a slightly restricted definition of ordering pair but it does not interfere with the topic of this work. An ordering pair is a pair $(\geq, >)$ of relations over $T(\mathcal{F}, X)$ such that: 1) $\geq$ is a quasi-ordering, i.e. reflexive and transitive, 2) $>$ is a strict ordering, i.e. irreflexive and transitive, and 3) $\geq \cdot > = >$.

An ordering $>$ is *well-founded* (denoted by $\mathsf{WF}(>)$) if there is no infinite strictly decreasing sequence $t_1 > t_2 > \ldots$. An ordering pair $(\geq, >)$ is *well-founded* (denoted by $\mathsf{WF}(\geq, >)$ if its strict ordering is well-founded. An ordering $<$ is stable by substitution if $\forall\sigma\forall t\forall u, t < u \Rightarrow t\sigma < u\sigma$. An ordering pair is *stable* if $>$ and $\geq$ are stable by substitution. If a strict ordering $>$ is monotonic we call it *strictly monotonic* (denoted $\mathsf{SM}(>)$). An ordering pair $(\geq, >)$ is *weakly monotonic* (denoted by $\mathsf{WM}(\geq, >)$ if $\geq$ is monotonic and *strictly monotonic*(denoted by $\mathsf{SM}(\geq, >)$) if $>$ is monotonic.

Termination— A term is S-*strongly normalizable* if it cannot reduce infinitely many times for $\rightarrow_S$. A rewrite relation $\rightarrow_S$ *terminates* if any term is S-strongly normalizable, which we denote $\mathsf{SN}(\rightarrow_S)$. In such case we may say that S terminates. A termination criterion due to Manna and Ness [3] states that it is sufficient to find a stable and well-founded strictly monotonic ordering $>$ such that for all rule $l \rightarrow r \in S$, $l > r$. This is stated in the rule MN below.

Moreover, it is also well known that the lexicographic combination of two well-founded relations is well-founded. This is stated in the rule LEX below. An effective termination criterion using this property is described in [13]. It allows to prove the so-called *relative* termination of a relation of the form $\rightarrow_{S_1}^* . \rightarrow_{S_2}$ by finding a strictly monotonic, stable and well-founded ordering pair $(\geq, >)$ for which all rules of S_1 decrease for $\geq$ and all rules of S_2 decrease for $>$. This is stated in the rule LEX_{AX} below.

Dependency pairs— The set of *unmarked* dependency pairs [2] of a TRS S, denoted $\mathsf{DP}(S)$ is defined as $\{\langle u, v\rangle \mid u \rightarrow t \in S \text{ and } t|_p = v \text{ and } \Lambda(v) \text{ is defined}\}$. Let $\mathscr{D}$ be a set of dependency pairs, a dependency chain in $\mathscr{D}$ is a *sequence* of dependency pairs $\langle u_i, v_i\rangle$ with a substitution σ such that $\forall i,\ v_i\sigma \xrightarrow[S]{\neq\Lambda\ \star} u_{i+1}\sigma$. Remark that to enhance this technique, implementations may distinguish the root symbols of dependency pairs (by means of marks). We will omit the details of this technique as it is not crucial in this work. Given a TRS S and a set of dependency pairs $\mathscr{D}$, $s \xrightarrow[S]{\neq\Lambda\ \star} u\sigma \xrightarrow[\langle u,v\rangle\in\mathscr{D}]{\Lambda} v\sigma \equiv t$ is denoted by $s \twoheadrightarrow_{\mathscr{D},S} t$. The main theorem of dependency pairs of [2] is the following: Let S be a TRS, $\twoheadrightarrow_{\mathsf{DP}(S),S}$ terminates if and only if $\rightarrow_S$ terminates. This is stated in the inference rule DP below. An effective technique for proving that $\twoheadrightarrow_{\mathscr{D},S}$ terminates consists in discovering a stable and well-founded *weakly monotonic* ordering pair $(\geq, >)$ for which $S \subseteq \geq$ and $\mathscr{D} \subseteq >$. This is stated in the rule DP_{AX} below.

Termination proofs— The algorithms of an automated termination prover is usually presented as popularised by the APROVE *processors* [15]. It transforms recursively problems into equivalent sets of sub-problems until each sub-problem can be directly solved by a suitable well-founded ordering (pair). We call *criterion* a transformation of a well-foundation problem p into a set of new problems $p_1 \ldots p_n$ such that p is well-founded iff $p_1 \ldots p_n$ are. Following the idea introduced in [5, 7] we model a termination proof by an inference tree where inference rules are criteria possibly guarded by a parameter (an ordering) and conditions. Guard conditions are properties that are not proved by inference trees but must be checked when applying rules. The termination criteria described above are summarized by the rules below[1]. Rules MN, LEX_{AX} and DP_{AX} are axioms of the inference system. In automated termination provers, these orderings are typically found by constraint solvers. In particular, term interpretation is

[1] Refer to [7] for a detailed presentation of more criteria in a similar framework

a well-known method to define such orderings.

$$\text{MN}(>)\ \frac{}{\mathsf{SN}(\rightarrow_S)}\ \begin{array}{l}\mathsf{WF}(>) \wedge \mathsf{SM}(>)\\ \wedge \forall l \rightarrow r \in S, l > r\end{array}$$

$$\textsc{Lex}\ \frac{\mathsf{SN}(\rightarrow_{S_1})\quad \mathsf{SN}(\rightarrow^*_{S_1} \cdot \rightarrow_{S_2})}{\mathsf{SN}(\rightarrow_{S_1 \cup S_2})} \qquad \textsc{Lex}_{\text{AX}}(\geq, >)\ \frac{}{\mathsf{SN}(\rightarrow^*_{S_1} \cdot \rightarrow_{S_2})}\ \begin{array}{l}\mathsf{WF}(\geq, >)\ \mathsf{SM}(\geq, >)\\ \forall l \rightarrow r \in S_2, l > r\\ \forall l \rightarrow r \in S_1, l \geq r\end{array}$$

$$\text{DP}\ \frac{\mathsf{SN}(\twoheadrightarrow_{DP(S),S})}{\mathsf{SN}(\rightarrow_S)} \qquad \text{DP}_{\text{AX}}(\geq, >)\ \frac{}{\mathsf{SN}(\twoheadrightarrow_{\mathscr{D},S})}\ \begin{array}{l}\mathsf{WF}(\geq, >)\ \mathsf{WM}(\geq, >)\\ \forall \langle l, r \rangle \in \mathscr{D}, l > r\\ \forall l \rightarrow r \in S, r \geq l\end{array}$$

2.2 Orderings by Interpretation

As explained in section above, a crucial task in termination proofs is to find *strictly* or *weakly monotonic* ordering pairs. In this section we describe the general framework of homomorphic interpretations which allows for both. All the following results are well known and can be found in [14, 8, 3]. In the sequel we suppose a non empty set D (domain), a quasi-ordering $\geq_D$ on D, and $>_D = \geq_D - \leq_D$. Therefore $(\geq_D, >_D)$ is an ordering pair. The following definitions and results are well known:

Definition 2.2.1. *A* valuation function *is a function $v : X \rightarrow D$ from variables to D.*

Definition 2.2.2. *A* homomorphic interpretation *φ is a function that takes a symbol f and returns a function $[f]_\varphi : D^n \rightarrow D$, where n is the arity of f. We define the homomorphic interpretation $\varphi(t)$ of a (possibly non-closed) term t as a function from valuation functions to D by induction on t as follows: $\varphi(x)(v) = v(x)$ and $\varphi(f(t_1, \ldots, t_n))(v) = [f]_\varphi(\varphi(t_1)(v), \ldots, \varphi(t_n)(v))$.*

Definition 2.2.3. *We define the ordering pair $(\succeq_\varphi, \succ_\varphi)$ on terms by: $s \succeq_\varphi t$ iff $\forall v \in (X \rightarrow D), \varphi(s)(v) \geq_D \varphi(t)(v)$ and $s \succ_\varphi t$ iff $\forall v \in (X \rightarrow D), \varphi(s)(v) >_D \varphi(t)(v)$.*

Theorem 2.2.1. *$(\succeq_\varphi, \succ_\varphi)$ is stable, and well-founded if $(\geq_D, >_D)$ is.*

Theorem 2.2.2. *If $[f]_\varphi$ is monotonic with respect to $>_D$ (respectively $\geq_D$), then $(\succeq_\varphi, \succ_\varphi)$ is* strictly monotonic *(respectively weakly monotonic).*

2.3 Matrix interpretation

The main idea of matrix interpretation of [11] is to define homomorphic interpretations suitable to apply rules MN, $\textsc{Lex}_{\text{AX}}$(strictly monotonic), and DP_{AX} (weakly monotonic) by interpreting terms as vectors ($D = \mathbb{N}^d$) using linear mappings represented by polynomials with matrix coefficients. The ordering pair on $\mathbb{N}^d$, that we note $(\geq_{\mathbb{N}^d}, >_{\mathbb{N}^d})$ is defined as follows: $(u_i) \geq_{\mathbb{N}^d} (v_i)$ iff $\forall i, u_i \geq_{\mathbb{N}} v_i$and $(u_i) >_{\mathbb{N}^d} (v_i)$ iff $\forall i, u_i \geq_{\mathbb{N}} v_i$ and $u_1 >_{\mathbb{N}} v_1$. As homomorphic interpretations defined by matrix polynomials may not be monotonic, Endrullis et alf [11] propose a restriction on the form of vectors and matrices to ensure strict monotonicity: the upper-left coefficient of vectors and matrices must be strictly positive.

In the following we define a family of interpretations parmetrized by the set of coefficients considered by the strict ordering. We adapt the restriction accordingly.

3 Generalized matrix interpretation

We use polynomials with *matrix* constants instead of vectors ($D = \mathbb{N}^{d \times d}$). This corresponds to the usual notion of polynomials where constants and coefficients have the same type. However all the following results and proofs are applicable to interpretations as defined in [11].

We define in the following matrix interpretation as homomorphic interpretations as defined in Section 2.2. First we define the ordering (family) $(\geq_{\mathbb{N}^{d \times d}}, >^E_{\mathbb{N}^{d \times d}})$ on the domain, then we define the form of an interpretation, finally we prove in which cases interpretations are weakly and strictly monotonic.

3.1 The ordering

We define a family of orderings $(\geq_{\mathbb{N}^{d \times d}}, >^E_{\mathbb{N}^{d \times d}})$ parametrized by the set $E \subset \mathbb{N}$ of column and line numbers that can be considered for strict comparison between matrices (the large comparison being on all coefficients).

Definition 3.1.1. *We define the orderings* $\geq_{\mathbb{N}^{d \times d}}$ *and* $>^E_{\mathbb{N}^{d \times d}}$ *on* $\mathbb{N}^{d \times d}$ *as follows: Let* $m, m' \in \mathbb{N}^{d \times d}$ *and* $E \subseteq \{1, ..., d\}$*,* $m \geq_{\mathbb{N}^{d \times d}} m' \iff \forall i, k \in [1..d], m_{ik} \geq_{\mathbb{N}} m'_{ik}$ *and* $m >^E_{\mathbb{N}^{d \times d}} m' \iff \forall i, k \in [1..d], m_{ik} \geq_{\mathbb{N}} m'_{ik} \wedge \exists i, j \in E, m_{ij} >_{\mathbb{N}} m'_{ij}$

Remark 1. By definition, if $E \subset E'$ then $>^E_{\mathbb{N}^{d \times d}} \subset >^{E'}_{\mathbb{N}^{d \times d}}$.

Lemma 3.1.1. *For any* E*,* $(\geq_{\mathbb{N}^{d \times d}}, >^E_{\mathbb{N}^{d \times d}})$ *is a well-founded ordering pair.*

Proof. $>^E_{\mathbb{N}^{d \times d}}$ is well-founded because it is included in the ordering $>^\Sigma$ defined by $m >^\Sigma m' \iff \sum_{1 \leq i,j \leq d} m_{ik} >_{\mathbb{N}} \sum_{1 \leq ik \leq d} m_{ik}$ which is well-founded. Moreover $\geq_{\mathbb{N}^{d \times d}} . >^E_{\mathbb{N}^{d \times d}} \subseteq >^E_{\mathbb{N}^{d \times d}}$ follows from $\geq_{\mathbb{N}} . >_{\mathbb{N}} = >_{\mathbb{N}}$ on each coefficient. □

3.2 The interpretation

We now define the homomorphic interpretation of a symbol $f \in \Sigma$ by a matrix linear polynomial, as explained in definition 2.2.2.

Definition 3.2.1 (matrix interpretation). *Given a signature* Σ *and a dimension* $d \in \mathbb{N}$*, a* matrix interpretation φ *is a homomorphic interpretation that takes a symbol* f *of arity* n *and returns a function of the form:* $[f]_\varphi(m_1, \ldots, m_n) = F_1 m_1 + \cdots + F_n m_n + F_{n+1}$ *where* $F_i \in \mathbb{N}^{d \times d}$ *and* $m_1, \ldots, m_n$ *take their values in* $\mathbb{N}^{d \times d}$*.*

Definition 3.2.2 (E-interpretation). *An* E*-interpretation is a matrix interpretation where the ordering pair used on matrices is* $(\geq_{\mathbb{N}^{d \times d}}, >^E_{\mathbb{N}^{d \times d}})$*.*

Definition 3.2.3. *The ordering pair* $(\succeq_\varphi, \succ^E_\varphi)$ *is defined from* $(\geq_{\mathbb{N}^{d \times d}}, >^E_{\mathbb{N}^{d \times d}})$ *as explained in definitions 2.2.3 (with* $D = \mathbb{N}^{d \times d}$*).*

Lemma 3.2.1. *Given an interpretation* φ*, The ordering pair* $(\succeq_\varphi, \succ^E_\varphi)$ *is (1) stable by substitution and (2) well-founded.*

Proof. (1) is proved by theorem 2.2.1 and (2) by theorems 2.2.1 and 3.1.1. □

The following lemma shows that homomorphic interpretations are weakly monotonic with respect to $(\succeq_\varphi, \succ_\varphi^E)$.

Lemma 3.2.2. *Let φ be a matrix interpretation. Then $(\succeq_\varphi, \succ_\varphi^E)$ is weakly monotonic.*

Proof. By lemma 2.2.2 it is sufficient to prove that for all symbol f, $[f]_\varphi$ is monotonic with respect to $\geq_{\mathbb{N}^{d\times d}}$. Let $f \in \Sigma$ of arity n and $1 \leq k \leq n$. Let $x, y, a_1 \dots a_n \in \mathbb{N}^{d\times d}$ s.t. $x \geq_{\mathbb{N}^{d\times d}} y$, let us show that $[f]_\varphi(a_1, ..., a_{k-1}, x, ..., a_n) \geq_{\mathbb{N}^{d\times d}} [f]_\varphi(a_1, ..., a_{k-1}, y, ..., a_n)$. By definition there exists $n+1$ matrices F_i such that:

$$\begin{aligned}
[f]_\varphi(a_1, ..., a_{k-1}, x, ..., a_n) &= F_1 a_1 + ... + F_k x + ... + F_n a_n + F_{n+1} \\
&= [f]_\varphi(..., 0, ...) + F_k x \\
[f]_\varphi(a_1, ..., a_{k-1}, y, ..., a_n) &= F_1 a_1 + ... + F_k y + ... + F_n a_n + F_{n+1} \\
&= [f]_\varphi(..., 0, ...) + F_k y
\end{aligned}$$

Since the (matrix $\times$ matrix) product is monotonic with respect to $\geq_{\mathbb{N}^{d\times d}}$, $F_k x \geq_{\mathbb{N}^{d\times d}} F_k y$ and thus $[f]_\varphi(a_1, ..., a_{k-1}, x, ..., a_n) \geq_{\mathbb{N}^{d\times d}} [f]_\varphi(a_1, ..., a_{k-1}, y, ..., a_n)$. □

Remark 2. The corollary of this lemma is that all matrix interpretations are suitable to define weakly monotonic orderings on terms, whatever E is. Therefore according to remark 1 we will always chose the maximal $E = \{1, \dots, d\}$ when searching weakly monotonic ordering pairs.

Remark 3. Since the (matrix $\times$ matrix) product is *not* monotonic with respect to $>^E_{\mathbb{N}^{d\times d}}$, there exists some E-interpretation such that $(\succeq_\varphi, \succ_\varphi^E)$ is not strictly monotonic.

Therefore we define the set of *E-compatible* matrices, parametrized by E, on which (matrix $\times$ matrix) product is monotonic with respect to $>^E_{\mathbb{N}^{d\times d}}$.

Definition 3.2.4. *Let $E \subseteq \{1, ..., d\}$, we call an E-position in a matrix $m \in \mathbb{N}^{d\times d}$ a position m_{ij} where $i \in E$ and $j \in E$. We also call E-columns and E-lines the sub-columns and sub-lines of E-positions.*

Definition 3.2.5 (**E-compatible matrices**)**.** *Let $E \subseteq \{1, ..., d\}$, we say that a matrix $m \in \mathbb{N}^{d\times d}$ is E-*compatible *if and only if each E-column is non null, that is at least one E-position on each E-column is non null.*

For example the matrix $\begin{pmatrix} \underline{\mathbf{0}} & 1 & \underline{\mathbf{0}} & 0 \\ 0 & 0 & 0 & 0 \\ \underline{\mathbf{2}} & 3 & \underline{\mathbf{1}} & 0 \\ 0 & 2 & 1 & 0 \end{pmatrix}$ is $\{1,3\}$-compatible whereas $\begin{pmatrix} \underline{\mathbf{1}} & 1 & \underline{\mathbf{0}} & 0 \\ 0 & 0 & 0 & 0 \\ \underline{\mathbf{2}} & 3 & \underline{\mathbf{0}} & 0 \\ 0 & 2 & 1 & 0 \end{pmatrix}$ is not.

Definition 3.2.6 (**E-compatible interpretation**)**.** *Let φ be a matrix interpretation. We say that φ is E-compatible if for all symbol f s. t. $[f]_\varphi(m_1, \dots, m_n) = F_1 m_1 + \dots + F_n m_n + F_{n+1}$, the matrices $F_1 \dots F_n$ are E-compatible. Notice that F_{n+1} does not need to be E-compatible.*

The following lemma shows that E-compatible homomorphic interpretations are strictly monotonic with respect to $(\succeq_\varphi, \succ_\varphi^E)$.

Lemma 3.2.3. *Let φ be an E-compatible interpretation. Then $(\succeq_\varphi, \succ_\varphi^E)$ is strictly monotonic.*

Proof. We proceed as above: By lemma 2.2.2 it is sufficient to prove that the following property holds for all symbol f (of arity n):

$$\forall 1 \leq k \leq n, \forall a_1 \dots a_{i-1}, a_{i+1} \dots a_n \in \mathbb{N}^{d\times d}, \forall x, y \in \mathbb{N}^{d\times d}, x >_{\mathbb{N}^{d\times d}}^E y \rightarrow$$
$$[f](a_1, \dots, a_{k-1}, x, a_{k+1}, \dots, a_n) >_{\mathbb{N}^{d\times d}}^E [f](a_1, \dots, a_{k-1}, y, a_{k+1}, \dots, a_n)$$

By definition there exists n E-compatible matrices $F_1 \dots F_n$ and a matrix F_{n+1} s.t.:

$$[f](a_1..., x, ...a_n) = F_1 m_1 + ... + F_k x + ... + F_n m_n + F_{n+1} = [f](..., 0, ...) + F_k x$$
$$[f](a_1..., y, ...a_n) = F_1 m_1 + ... + F_k y + ... + F_n m_n + F_{n+1} = [f](..., 0, ...) + F_k y$$

Therefore it is sufficient to prove that $\forall 1 \leq k \leq n, \forall x, y \in \mathbb{N}^{d\times d}, x >_{\mathbb{N}^{d\times d}}^E y \implies F_k x >_{\mathbb{N}^{d\times d}}^E F_k y$. Since the product ($E$ compatible matrix) $\times$ (matrix) is monotonic with respect to $>_{\mathbb{N}^{d\times d}}^E$, the statement of the lemma follows. □

The corollary of this lemma is that when an E-interpretation is E-compatible, it can be used to build a strictly monotonic ordering pair on terms.

3.3 Proving termination

To prove termination of a given TRS R using rules MN, Lex_{AX} or DP_{AX}, we need to compare matrix interpretations of the left hand side and the right hand side of rules with $\succ_\varphi$. These interpretations can be computed by developing polynomials, as stated by the two following lemmas:

Lemma 3.3.1. *Let φ be a matrix interpretation and t, a term with n free variables $x_1 \dots x_n$. There exists $n+1$ matrices $M_1 \dots M_{n+1}$ such that $\varphi(t)(v) = M_1 v(x_1) + \dots + M_n v(x_n) + M_{n+1}$.*

Proof. By induction on t. If t is a variable x, then by definition 2.2.2 the property holds: $\varphi(x)(v) = v(x)$. If $t = f(t_1, \dots, t_m)$ then by definition 2.2.2: $\varphi(t)(v) = [f]_\varphi(\varphi(t_1)(v), \dots, \varphi(t_m)(v)) = F_1(\varphi(t_1)(v)) + \dots + F_m(\varphi(t_m)(v)) + F_{m+1}$ where by induction hypothesis each $\varphi(t_i)(v)$ is itself a linear polynomial of the form $\sum_j M_{i_j} v(x_j) + M_{i_{n+1}}$. Thus $\varphi(t)(v)$ is equal to $(\sum_k F_k M_{k_1}) v(x_1) + \dots + (\sum_k F_k M_{k_n}) v(x_n) + \left(\sum_k F_k M_{k_{n+1}}\right) + F_{m+1}$. □

Lemma 3.3.2. *Let φ be an E-compatible homomorphic interpretation and t a term containing n variables $x_1 \dots x_n$. There exists a set of n E-compatible matrices $M_1 \dots M_n$ and a matrix M_{n+1} such that $\varphi(t)(v) = M_1 v(x_1) + \dots + M_n v(x_n) + M_{n+1}$.*

Proof. We proceed by the same induction as above and in equation above $F_1 \dots F_n$ are E-compatible matrices by hypothesis, and $M_{k_1} \dots M_{k_n}$ are E-compatible matrices by induction hypothesis. Since matrix addition and product are stable on E-compatible matrices we can conclude that the $\sum_k F_k M_{k_i}$ are E-compatible matrices in equation above. □

Therefore in order to check that rules or dependency pairs are decreasing, we must compare matrix linear polynomials, which is decidable:

Lemma 3.3.3. *Let t and u be terms such that $\varphi(t)(v) = L_1 v(x_1) + \cdots + L_k v(x_k) + L_{k+1}$ and $\varphi(u)(v) = R_1 v(x_1) + \cdots + R_k v(x_k) + R_{k+1}$. If $\forall 1 \leq i \leq k+1, L_i \geq_{\mathbb{N}^{d\times d}} R_i$ then $\varphi(t)(v) \geq_{\mathbb{N}^{d\times d}} \varphi(u)(v)$ for any valuation $v : \mathbb{N}^k \to \mathbb{N}$. If moreover $L_{k+1} >^E_{\mathbb{N}^{d\times d}} R_{k+1}$, then $\varphi(t)(v) >^E_{\mathbb{N}^{d\times d}} \varphi(u)(v)$ for any valuation $v : \mathbb{N}^k \to \mathbb{N}$.*

Proof. Let v be a valuation. Since $\forall 1 \leq i \leq k+1, L_i \geq_{\mathbb{N}^{d\times d}} R_i$, the matrix $m = \varphi(t)(v) - \varphi(u)(v)$ is such that $m = \sum_{i=1}^{k} ((L_i - R_i)v(x_i)) + L_{k+1} - R_{k+1} \geq_{\mathbb{N}^{d\times d}} \mathbf{0}$ which proves the first property. If $L_{k+1} >^E_{\mathbb{N}^{d\times d}} R_{k+1}$ then moreover we have $m \geq_{\mathbb{N}^{d\times d}} L_{k+1} - R_{k+1} >^E_{\mathbb{N}^{d\times d}} \mathbf{0}$. □

4 Proof search

In this section we describe the adaptation of the method of [11] for generating termination proofs. The main differences are the choice of an E, the treatment of E-compatibility and the ordering constraints using E.

Due to the symmetrical shape of our orderings with respect to matrices, it is clear that for E and E' having the same cardinality, if there exists an E-interpretation satisfying conditions of lemma 3.3.3, then there exists an E'-interpretation satisfying the same conditions, obtained by applying to all matrices the same column and line permutation. Therefore it is enough to try each E of the form $\{1, \ldots, n\}$ where $2 \leq n \leq d$.

4.1 Manna and Ness Criterion

In order to prove the termination of a given TRS S using Rule MN, we need to find an E and an E-compatible matrix interpretation φ such that $\forall l \to r \in S, \varphi(l) \succ^E_\varphi \varphi(r)$. This amounts to solving constraints on matrix coefficients. More precisely, for each rule $l \to r \in S$, where $\varphi(l) = \sum_1^n L_i x_i + L_{n+1}$ and $\varphi(r) = \sum_1^n R_i x_i + R_{n+1}$ (If r has less variables than l, the corresponding R_i are null matrices), we have the following constraint: $\forall 1 \leq i \leq n, L_i \geq_{\mathbb{N}^{d\times d}} R_i$ and $L_{n+1} >^E_{\mathbb{N}^{d\times d}} R_{n+1}$. The E-compatibility of interpretation are also expressed as constraints on E-positions.

We try to solve these constraints using a SAT solver, which is common practice [12, 1],[11]. In order to call the SAT solver once, we encode the constraints corresponding to all desired sizes of E in one disjunctive formula.

4.2 Lexicographic composition criterion

In order to use the lexicographic criteria we need to split the TRS S into two systems S_1 and S_2, such that we can apply rule LEX_{AX} to prove $\mathsf{SN}(\to_1^* \cdot \to_2)$. Then we are left with the property $\mathsf{SN}(S_1)$ that can be proved by any other criterion recursively.

$$\text{LEX}\ \dfrac{\dfrac{\vdots}{\mathsf{SN}(\to_{S_1})} \qquad \text{LEX}_{\text{AX}}(\succeq_\varphi, \succ^E_\varphi)\ \dfrac{}{\mathsf{SN}(\to^*_{S_1} \cdot \to_{S_2})}\ \begin{array}{l}\mathsf{WF}(\succeq_\varphi, \succ^E_\varphi)\ \mathsf{SM}(\succeq_\varphi, \succ^E_\varphi) \\ \forall l \to r \in S_2, l \succ^E_\varphi r \\ \forall l \to r \in S_1, l \succeq_\varphi r\end{array}}{\mathsf{SN}(\to_{S_1 \cup S_2})}$$

In order to find φ we first fix E then we solve the following constraint: $\forall l \to r \in S, l \succeq_\varphi r \wedge \exists l \to r \in S, l \succ_\varphi^E r$. The existential part of this property may be expressed by a disjunction on rules of S. If a solution is found, then S_2 is the set of strictly decreasing rules and S_1 the remaining ones. As previously we can try several E.

4.3 Dependency pairs criterion

In order to use the dependency pair criterion, we first need to apply DP then find an matrix interpretation φ satisfying the condition of Rule $\mathrm{DP}_{\mathrm{AX}}$. This is done by similar techniques than above taking the maximal E as explained in remark 2.

4.4 Comparison with previous notions of matrix interpretation

The interpretation defined in [11] almost corresponds to one member of our family of interpretations, namely $\{1\}$-interpretations. To be precise it corresponds to $\{1\}$-interpretations where constant coefficients of polynomials are vectors instead of matrices. In the following we analyze the differences between $\{1\}$-interpretations and E-interpretations where $|E| > 1$ in the case of each criterion. For the symmetry reasons given in Section 4, we focus on $\{1, \dots, k\}$-interpretations.

MN and $\mathbf{L\scriptsize{EX}}_{\mathbf{AX}}$— In the strict monotonic setting, when $E \neq \{1\}$ matrix interpretations do not solve the same sets of problems. This is due to several facts. On one hand a greater E makes more matrices comparable. For instance $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ are comparable with $>^{\{1,2\}}_{\mathbb{N}^{2\times2}}$ but not with $>^{\{1\}}_{\mathbb{N}^{2\times2}}$. Therefore the comparison of *constant* coefficient of polynomials is more powerful when E is greater.

On the other hand strict monotonicity constraints (for non constant coefficients) are such that the sets of allowed matrices are different when E changes. More precisely there is no inclusion relation between them. For example if f is a unary symbol, then $[f](m) = \begin{pmatrix} \mathbf{0} & 0 \\ 1 & 1 \end{pmatrix} m + \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ is $\{1,2\}$-compatible and not $\{1\}$-compatible, whereas $[f](m) = \begin{pmatrix} 1 & \mathbf{0} \\ 1 & \mathbf{0} \end{pmatrix} m + \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$ is $\{1\}$-compatible and not $\{1,2\}$-compatible. Therefore the set of ordering problems solved by different sizes of E are usually different. For this reason an implementation should try all possible size for E. In practice in our prototype C*i*ME3 this is configurable.

$\mathbf{DP}_{\mathbf{AX}}$— In the weak monotonic setting, there is no monotonicity constraint on matrices, therefore the set of allowed matrices is the same whatever E is. Therefore the maximal $E = \{1, \dots, n\}$ is always more powerful because, as said above, it allows for more matrices to be compared strictly.

However this statement is not true anymore when trying to remove *only one* pair $\langle l, r\rangle$ of a set of dependency pairs $\mathscr{D}$. This is done (for example in the graph refinement) by finding a weakly monotonic well-founded ordering pair $(\geq, >)$ such that: $l > r$ and $\forall \langle t, u\rangle \in \mathscr{D}, t \geq u$ and $\forall t \to u \in \mathscr{D}, t \geq u$. In that case, the fact that only one pair needs to be ordered strictly implies that if a solution exists with any non empty E, then by the adequate permutation of columns and lines, we can obtain an interpretation which also works for $E = \{1\}$. Therefore for example *the choice of E is not critical anymore* when using the graph refinement, as shown in the results of section 6. However, a greater E may lead to shorter proofs, which is interesting in the framework of termination *certificate* (see Section 7).

As a conclusion, we see that the best strategy is to try all possible sizes of E for MN and LEX, and only the maximal E for DP.

5 Examples

In this section we show examples of rewrite systems where $\{1, 2\}$-interpretations are used to prove termination, whereas $\{1\}$-interpretations cannot. In all these examples, matrix coefficients are forced to be 0 or 1. It is worth noticing that some of these examples can be solved by $\{1\}$-interpretations if the bound on matrix coefficients is higher, but at a price of a greater search space.

LEX and LEX$_{\text{AX}}$— Consider the following rewrite system: $\{(1)\ \mathit{plus}(\mathit{plus}(x, y), z) \to \mathit{plus}(x, \mathit{plus}(y, z));\ (2)\ \mathit{times}(x, s(y)) \to \mathit{plus}(x, \mathit{times}(y, x))\}$. Rule (2) can be removed as explained in section 4.2 by the following interpretation:

$$[\mathit{plus}]_\varphi(x, y) = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} x + \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} y + \begin{pmatrix}1 & 0\\0 & 0\end{pmatrix} \qquad [s]_\varphi(x) = \begin{pmatrix}1 & 1\\1 & 1\end{pmatrix} x + \begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}$$

$$[\mathit{times}]_\varphi(x, y) = \begin{pmatrix}1 & 1\\1 & 1\end{pmatrix} x + \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} y$$

and rule (1) by:

$$[\mathit{plus}]_\varphi(x, y) = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix} y + \begin{pmatrix}1 & 1\\0 & 1\end{pmatrix} x + \begin{pmatrix}0 & 1\\1 & 1\end{pmatrix}$$

DP— Consider the system: $f(0, x) \to f(s(x), x);\ f(x, s(z)) \to s(f(0, z))$ which leads to the following dependency pairs: $\langle f(x, s(z)), f(0, z)\rangle$ and $\langle f(0, x), f(s(x), x)\rangle$. There is no matrix $\{1\}$-interpretation (with coefficients bound ≤ 1) such that all pairs are strictly decreasing and all rule weakly decreasing. However there is a $\{1, 2\}$-interpretation (DP$_{\text{AX}}$):

$$[f]_\varphi(x, y) = \begin{pmatrix}0 & 0\\0 & 1\end{pmatrix} x + \begin{pmatrix}1 & 0\\1 & 0\end{pmatrix} y \quad [s]_\varphi(x) = \begin{pmatrix}1 & 0\\0 & 0\end{pmatrix} x + \begin{pmatrix}0 & 1\\0 & 0\end{pmatrix} \quad [0]_\varphi() = \begin{pmatrix}1 & 1\\0 & 1\end{pmatrix}$$

6 Results

The benchmarks were made with a prototype of C*i*ME on the 1436 problems of the termination problems database (TPDB) (category *standard TRS termination*, 2008-11-04 termination competition). Ordering constraints are solved by giving an upper bound b to matrix coefficients and then by translation to the SAT solver `minisat2` [10]. Each call to the SAT solver is limited to 100s and the overall timeout is 300s for each problem. The first table compares the number of problems solved using *matrix 2×2 interpretations only* with different E and b. The tested criteria are: MN, LEX, DP, DPG (graph refinement of dependency pairs), LGST (LEX *then* graph and subterm refinements). The latter being close to the best heuristic of C*i*ME. The second table shows the results using the strategy LGST and the usual combination of orderings of C*i*ME (linear polynomial, RPO, simple polynomial) followed by matrix interpretations (2×2 and 3×3). This shows how our matrix interpretations increase the power of the full system[2].

Ordering = Matrix interpretation only, matrix size = 2															
Criterion	MN			LEX			DP			DPG			LGST		
Bounds	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
$E=\{1\}$	88	167	186	230	278	285	266	345	358	433	448	452	466	479	482
$E=\{1,2\}$	+41,-0	+14,-0	+6,-6	+26,-0	+39,-2	+36,-3	+63,-0	+11,-1	+3,-2	+0,-0	+0,-6	+0,-12	+14,-0	+30,-11	+30,-7

Ordering = usual+Matrix interpretation						
Criterion	LGST (mat. 2×2)			LGST (mat. 3×3)		
Bounds	1	2	3	1	2	3
$E=\{1\}$	576	583	586	592	588	587
$E=\{1,2\}$	+5,-0	+12,-0	+16,-1	+3,-0	+7,-10	+10,-10
$E=\{1,2,3\}$	N/A	N/A	N/A	+7,-3	+13,-19	+10,-17

A cell containing $+n, -m$ sums up the comparison with $E=\{1\}$: n new problems solved, m problems not solved anymore because of timeouts. Timeouts are caused by larger E leading to more complex constraints, despite the search space is the same. Our benchmarks showed an average overhead time of 20 to 30%. This explains why the current state of our implementation does not always reflect the expected improvement of our interpretations, in particular with 3×3 matrices. Except those timeouts, larger E is, as expected, always more powerful excepted in the DPG column (see section 4.4).

7 Conclusion and Future work

Our approach generalizes the original matrix interpretations. It should naturally extend to other refinement of matrix interpretations such as arctic interpretations (where the usual plus/times operations are generalized to an arbitrary semi-ring [17]). Our approach using true polynomials over matrices, instead of mixing matrices and vectors, may allow for *matrix non linear polynomials*. Another point is that when discovering a solution our implementation (an early prototype of C*i*ME-3) *produces a proof trace* which we translate into a *proof certificate*[5] for verification. We are currently working on adapting our proofs to our matrix interpretations.

References

1. Elena Annov, Michael Codish, Jürgen Giesl, Peter Schneider-Kamp, and René Thiemann. A sat-based implementation for rpo termination. In *International Conference on Logic for Programming, Artificial Intelligence and Reasoning (Short Paper)*, November 2006.
2. Thomas Arts and Jürgen Giesl. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236:133–178, 2000.
3. Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[2] C*i*ME is devoted to certification of termination proofs, it only implements criteria that it can certify and it is not multi-threaded. Therefore it performs lower than state of the art provers such as APROVE.

4. Ahlem Ben Cherifa and Pierre Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Science of Computer Programming*, 9(2):137–159, 1987.
5. Évelyne Contejean, Pierre Courtieu, Julien Forest, Olivier Pons, and Xavier Urbain. Certification of automated termination proofs. In Boris Konev and Frank Wolter, editors, *6th International Symposium on Frontiers of Combining Systems (FroCos 07)*, volume 4720 of *Lecture Notes in Artificial Intelligence*, pages 148–162, Liverpool,UK, September 2007. Springer-Verlag.
6. Évelyne Contejean, Claude Marché, Ana Paula Tomás, and Xavier Urbain. Mechanically proving termination using polynomial interpretations. *Journal of Automated Reasoning*, 34(4):325–363, 2005.
7. Pierre Courtieu, Julien Forest, and Xavier Urbain. Certifying a Termination Criterion Based on Graphs, Without Graphs. In César Muñoz and Otmane Ait Mohamed, editors, *21st International Conference on Theorem Proving in Higher Order Logics (TPHOLs'08)*, volume 5170 of *Lecture Notes in Computer Science*, pages 183–198, Montréal, Canada, August 2008. Springer-Verlag.
8. Nachum Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science*, 17(3):279–301, March 1982.
9. Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–320. North-Holland, 1990.
10. Niklas Eén and Armin Biere. Effective preprocessing in sat through variable and clause elimination. In *In proc. SAT'05, volume 3569 of LNCS*, pages 61–75. Springer, 2005.
11. Jörg Endrullis, Johannes Waldmann, and Hans Zantema. Matrix interpretations for proving termination of term rewriting. *jar*, 40(2-3):195–220, 2008.
12. Carsten Fuhs, Aart Middeldorp, Peter Schneider-kamp, and Harald Zankl. Sat solving for termination analysis with polynomial interpretations. In *In SAT'07, LNCS 4501*, pages 340–354. Springer-Verlag, 2007.
13. Alfons Geser. *Relative termination*. Dissertation, Fakultät für Mathematik und Informatik, Universität Passau, Germany, 1990. 105 pages. Also available as: Report 91-03, Ulmer Informatik-Berichte, Universität Ulm, 1991.
14. Jürgen Giesl, editor. *Term Rewriting and Applications*, volume 3467 of *Lecture Notes in Computer Science*, Nara, Japan, April 2005. Springer-Verlag.
15. Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, and Stephan Falke. Mechanizing and Improving Dependency Pairs. *Journal of Automated Reasoning*, 37(3):155–203, 2006.
16. Dieter Hofbauer and Johannes Waldmann. Termination of string rewriting with matrix interpretations. In Frank Pfenning, editor, *RTA*, volume 4098 of *Lecture Notes in Computer Science*, pages 328–342. Springer, 2006.
17. Adam Koprowski and Johannes Waldmann. Arctic termination ...below zero. In Andrei Voronkov, editor, *RTA*, volume 5117 of *Lecture Notes in Computer Science*, pages 202–216, Hagenberg, Austria, July 2008. Springer-Verlag.
18. Keiichirou Kusakari, Masaki Nakamura, and Yoshihito Toyama. Argument filtering transformation. In Gopalan Nadathur, editor, *Principles and Practice of Declarative Programming, International Conference PPDP'99*, volume 1702 of *Lecture Notes in Computer Science*, pages 47–61, Paris, 1999. Springer-Verlag.
19. Dallas S. Lankford. On proving term rewriting systems are Noetherian. Technical Report MTP-3, Mathematics Department, Louisiana Tech. Univ., 1979. Available at `http://perso.ens-lyon.fr/pierre.lescanne/not_accessible.html`.