Acta Cybernetica **19** (2009) 357–392.

Max/Plus Tree Automata for Termination of Term Rewriting

Adam Koprowski* and Johannes Waldmann†

Abstract

We use weighted tree automata as certificates for termination of term rewriting systems. The weights are taken from the arctic semiring: natural numbers extended with $-\infty$, with the operations "max" and "plus". In order to find and validate these certificates automatically, we restrict their transition functions to be representable by matrix operations in the semiring. The resulting class of weighted tree automata is called *path-separated*.

This extends the matrix method for term rewriting and the arctic matrix method for string rewriting. In combination with the dependency pair method, this allows for some conceptually simple termination proofs in cases where only much more involved proofs were known before. We further generalize to arctic numbers "below zero": integers extended with $-\infty$. This allows to treat some termination problems with symbols that require a predecessor semantics.

Correctness of this approach has been formally verified in the Coq proof assistant and the formalization has been contributed to the CoLoR library of certified termination techniques. This allows formal verification of termination proofs using the arctic matrix method in combination with the dependency pair transformation. This contribution brought a substantial performance gain in the certified category of the 2008 edition of the termination competition.

The method has been implemented by leading termination provers. We report on experiments with its implementation in one such tool, Matchbox, developed by the second author.

We also show that our method can simulate a previous method of quasi-periodic interpretations, if restricted to interpretations of slope one on unary signatures.

Keywords: term rewriting, termination, weighted tree automaton, max/plus algebra, arctic semiring, monotone algebra, matrix interpretation, formal verification

*Institute for Computing and Information Science, Radboud University Nijmegen, P.O. Box 9010, 6500 GL Nijmegen, The Netherlands *and* MLstate, rue Berlier 15, 75013 Paris, France, E-mail: `Adam.Koprowski@cs.ru.nl`

†Hochschule für Technik, Wirtschaft und Kultur Leipzig, Fb IMN, PF 30 11 66, D-04251 Leipzig, Germany, E-mail: `waldmann@imn.htwk-leipzig.de`

1 Introduction

One method of proving termination is interpretation into a well-founded algebra. Polynomial interpretations (over the naturals) are a well-known example of this approach. Another example is the recent development of the matrix method [22, 13] that uses linear interpretations over vectors of naturals, or equivalently, $\mathbb{N}$-weighted automata. In [38, 37] one of the authors extended this method (for string rewriting) to arctic automata, *i.e.*, on the max/plus semiring on $\{-\infty\}\cup\mathbb{N}$. Its implementation in the termination prover Matchbox [36] contributed to this prover winning the string rewriting division of the 2007 termination competition [31, 1].

The first contribution of the present work is a *generalization of arctic termination to term rewriting*. We use interpretations given by functions of the form $(\vec{x}_1, \ldots, \vec{x}_n) \mapsto M_1 \cdot \vec{x}_1 + \ldots + M_n \cdot \vec{x}_n + \vec{c}$. Here, $\vec{x}_i$ are (column) vector variables, $\vec{c}$ is a vector and $M_1, \ldots, M_n$ are square matrices, where all entries are arctic numbers, and operations are understood in the arctic semiring.

Functions of this shape compute the transition function of a weighted tree automaton [10, 9]. The vectors correspond to assignments from states to weights.

Since the max operation is not strictly monotone in single arguments, we obtain monotone interpretations only for the case when all function symbols are at most unary, *i.e.*, string rewriting. For symbols of higher arity, arctic interpretations are weakly monotone. These cannot prove termination, but only top termination, where rewriting steps are only applied at the root of terms. This is a restriction but it fits with the framework of the dependency pair method [4] that transforms a termination problem to a top termination problem.

The second contribution is a *generalization from arctic naturals to arctic integers*, *i.e.*, $\{-\infty\} \cup \mathbb{Z}$. Arctic integers allow for example to interpret function symbols by the predecessor function and this matches the "intrinsic" semantics of some termination problems. There is previous work on polynomial interpretations with negative coefficients [19, 20], where the interpretation for predecessor is also expressible using ad-hoc max operations. Using arctic integers, we obtain verified termination proofs for 10 of the 24 rewrite systems Beerendonk/* from the Termination Problem Database [2] (TPDB), simulating imperative computations. Previously, they could only be handled by the method of Bounded Increase [17] and polynomial interpretations with rational coefficients [30].

The third contribution is that we can express *quasi-periodic interpretations* [39] of slope one, another powerful method for proving termination of rewriting, as an instance of arctic interpretations for unary signatures.

The next contribution is the fact that the correctness of this method for proving top termination has been *formally verified with the proof assistant Coq* [34]. This extends previous work [27] and is now part of the CoLoR project [7] that gathers formalizations of termination techniques and employs them to certify proofs found by tools for automatic termination proving. This contribution was crucial in enabling CoLoR to win against the competing certification back-end, A3PAT [8], in the termination competition of 2008 [1].

A method to search for arctic interpretations is implemented for the termination prover Matchbox. It works by transformation to a boolean satisfiability problem and application of a state-of-the-art SAT solver (in this case, Minisat [11]). For a number of problems Matchbox produced certified termination proofs, where only un-certified proofs were available before. Recently the arctic interpretations method was also implemented in AProVE [16] and $\mathsf{T_TT_2}$ [28].

The paper is organized as follows. We present notation and basic facts on rewriting in Section 2 and give an introduction to proving termination of rewriting using the monotone algebra framework in Section 3. Then we give preliminaries on the arctic semiring in Section 4, and we relate the monotone algebra approach to the concept of weighted tree automata in Section 5. We present arctic interpretations for termination in Section 6, for top termination in Section 7 and the generalization to arctic integers in Section 8. In Section 9 we show that quasi-periodic interpretations of slope one for proving termination of string rewriting [39] are a special case of arctic matrix interpretations. We report on the formal verification in Section 10 and on performance of our implementation in Section 11. We present some discussion of the method, its limitations and related work in Section 12 and we conclude in Section 13.

Preliminary versions of the results from this paper have been presented at the Workshop on Termination [37], at the Workshop on Weighted Automata [26], and at the Conference on Rewriting Techniques and Applications [25]. We thank the anonymous referees for their comments.

2 Term Rewriting

In this section we shortly introduce the basic notions on term rewriting. For more details we refer to [5].

Let Σ be a signature, that is, a set of operation symbols each having a fixed arity in $\mathbb{N}$. For a set of variable symbols $\mathcal{V}$, disjoint from Σ, let $\mathcal{T}(\Sigma, \mathcal{V})$ be the set of terms over Σ and $\mathcal{V}$, that is, the smallest set satisfying

- $x \in \mathcal{T}(\Sigma, \mathcal{V})$ for all $x \in \mathcal{V}$, and
- if the arity of $f \in \Sigma$ is n and $t_i \in \mathcal{T}(\Sigma, \mathcal{V})$ for $i = 1, \ldots, n$, then $f(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \mathcal{V})$.

Terms are identified with finitely branching labeled trees. We denote a root of a term t by $\mathrm{root}(t)$ and $\mathrm{root}(f(t_1, \ldots, t_n)) = f$. By $\trianglelefteq$ we denote the sub-term relation on terms and we have $t \trianglelefteq u$ if t is a sub-tree of u.

A *term rewriting system* (TRS) $\mathcal{R}$ over $\Sigma, \mathcal{V}$ is a set of pairs $(\ell, r) \in \mathcal{T}(\Sigma, \mathcal{V}) \times \mathcal{T}(\Sigma, \mathcal{V})$, for which $\ell \notin \mathcal{V}$ and all variables in r occur in ℓ. Pairs (ℓ, r) are called *rewrite rules* and are usually written as $\ell \to r$.

A TRS with all functions having arity one is called a *string rewriting system* (SRS). For SRSs it is customary to write terms as strings, so $a_1(a_2(\ldots(a_n(x))\ldots))$ becomes $a_1 a_2 \cdots a_n$ and x becomes ϵ.

For a substitution $\sigma : \mathcal{V} \to \mathcal{T}(\Sigma, \mathcal{V})$ and a term t the application of σ to t, denoted by $t\sigma$, is a term defined inductively as:

- $x\sigma = \sigma(x)$ for all $x \in \mathcal{V}$, and
- $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$.

For a TRS $\mathcal{R}$ the *top rewrite relation* $\overset{\text{top}}{\to}_{\mathcal{R}}$ on $\mathcal{T}(\Sigma, \mathcal{V})$ is defined by $t \overset{\text{top}}{\to}_{\mathcal{R}} u$ if and only if there is a rewrite rule $\ell \to r \in \mathcal{R}$ and a substitution $\sigma : \mathcal{V} \to \mathcal{T}(\Sigma, \mathcal{V})$ such that $t = \ell\sigma$ and $u = r\sigma$. The *rewrite relation* $\to_{\mathcal{R}}$ is defined to be the smallest relation satisfying

- if $t \overset{\text{top}}{\to}_{\mathcal{R}} u$ then $t \to_{\mathcal{R}} u$, and
- if $t_i \to_{\mathcal{R}} u_i$ and $t_j = u_j$ for $j \neq i$, then $f(t_1, \ldots, t_n) \to_{\mathcal{R}} f(u_1, \ldots, u_n)$ for every $f \in \Sigma$ of arity n.

A relation $\to$ is terminating if it does not admit infinite descending chains $t_0 \to t_1 \to \ldots$, denoted as $\mathrm{SN}(\to)$. For relations $\to_1, \to_2$, we define $\to_1 / \to_2$ by $(\to_1) \cdot (\to_2)^*$. If $\mathrm{SN}(\to_1 / \to_2)$, we say that $\to_1$ is terminating relative to $\to_2$.

When given as arguments to SN we will often identify TRSs with the rewrite relations generated by them and hence abbreviate $\overset{\text{top}}{\to}_{\mathcal{R}}$ by $\mathcal{R}_{\text{top}}$ and $\to_{\mathcal{R}}$ by $\mathcal{R}$.

Now we will shortly introduce the dependency pair method [4] — a powerful approach for proving termination of rewriting, used by most of the termination provers.

Definition 2.1. [Dependency pairs] Let $\mathcal{R}$ be a TRS over a signature Σ. The set of *defined* symbols is defined as $\mathcal{D}_{\mathcal{R}} = \{\mathrm{root}(l) \mid l \to r \in \mathcal{R}\}$. We extend a signature Σ to the signature $\Sigma^\sharp$ by adding symbols $f^\sharp$ for every symbol $f \in \mathcal{D}_{\mathcal{R}}$. If $t \in \mathcal{T}(\Sigma, \mathcal{V})$ with $\mathrm{root}(t) \in \mathcal{D}_{\mathcal{R}}$ then $t^\sharp$ denotes the term that is obtained from t by replacing its root symbol with $\mathrm{root}(t)^\sharp$.

If $l \to r \in \mathcal{R}$ and $t \trianglelefteq r$ with $\mathrm{root}(t) \in \mathcal{D}_{\mathcal{R}}$ then the rule $l^\sharp \to t^\sharp$ is a *dependency pair* of $\mathcal{R}$. The set of all dependency pairs of $\mathcal{R}$ is denoted by $\mathrm{DP}(\mathcal{R})$. ◇

The main theorem underlying the dependency pair method is the following.

Theorem 2.2 ([4])**.** Let $\mathcal{R}$ be a TRS. $\mathrm{SN}(\mathcal{R})$ iff $\mathrm{SN}(\mathrm{DP}(\mathcal{R})_{\text{top}}/\mathcal{R})$. □

In this paper we will consider problems of termination of rewrite relations generated by some term rewriting systems. Three types of problems will be of interest:

- Full termination: *given a TRS $\mathcal{R}$, is it terminating, i.e., does* $\mathrm{SN}(\mathcal{R})$ *hold?*
- Relative termination: *given two TRSs $\mathcal{R}, \mathcal{S}$, does $\mathcal{R}$ terminate relative to $\mathcal{S}$, i.e., does* $\mathrm{SN}(\mathcal{R}/\mathcal{S})$ *hold?*
- Relative top termination: *given two TRSs $\mathcal{R}, \mathcal{S}$ does $\mathcal{R}$ terminate relative to $\mathcal{S}$ if we allow only top reductions in $\mathcal{R}$, i.e., does* $\mathrm{SN}(\mathcal{R}_{\text{top}}/\mathcal{S})$ *hold?*

Note that termination is a special case of relative termination as $\mathrm{SN}(\mathcal{R}) \iff \mathrm{SN}(\mathcal{R}/\emptyset)$, hence we will present results for relative termination only as they are immediately applicable for the full termination case. Relative top termination is of special interest due to its close relation with the dependency pair method, established in Theorem 2.2.

We will illustrate some term rewriting notions on an example.

Example 2.3. Consider the following three rules TRS $\mathcal{R}$, AG01/#3.41 from the TPDB [2], over the signature $\Sigma = \{0, \mathrm{p}, \mathrm{s}, \mathrm{fac}, \mathrm{times}\}$:

$$\begin{aligned} \mathrm{p}(\mathrm{s}(x)) &\to x \\ \mathrm{fac}(0) &\to \mathrm{s}(0) \\ \mathrm{fac}(\mathrm{s}(x)) &\to \mathrm{times}(\mathrm{s}(x), \mathrm{fac}(\mathrm{p}(\mathrm{s}(x)))) \end{aligned}$$

This TRS represents computation of the factorial function (without the rules for addition and multiplication) with natural numbers represented with zero (0), successor (s) and predecessor (p) and the factorial function (fac) expressed using multiplication (times).

We have the following reduction sequence (with the redex at every step underlined):

$$\begin{gathered} \underline{\mathrm{fac}(\mathrm{s}(\mathrm{s}(0)))} \xrightarrow{\mathrm{top}}_{\mathcal{R}} \mathrm{times}(\mathrm{s}(\mathrm{s}(0)), \mathrm{fac}(\underline{\mathrm{p}(\mathrm{s}(\mathrm{s}(0)))})) \to_{\mathcal{R}} \\ \mathrm{times}(\mathrm{s}(\mathrm{s}(0)), \underline{\mathrm{fac}(\mathrm{s}(0))}) \to_{\mathcal{R}} \mathrm{times}(\mathrm{s}(\mathrm{s}(0)), \mathrm{times}(\mathrm{s}(0), \mathrm{fac}(\underline{\mathrm{p}(\mathrm{s}(0))}))) \to_{\mathcal{R}} \\ \mathrm{times}(\mathrm{s}(\mathrm{s}(0)), \mathrm{times}(\mathrm{s}(0), \underline{\mathrm{fac}(0)})) \to_{\mathcal{R}} \mathrm{times}(\mathrm{s}(\mathrm{s}(0)), \mathrm{times}(\mathrm{s}(0), \mathrm{s}(0))) \end{gathered}$$

calculating that factorial of two equals $2 \times (1 \times 1)$. We also have two defined symbols, $\mathcal{D}_{\mathcal{R}} = \{\mathrm{p}, \mathrm{fac}\}$, extended signature $\Sigma^\sharp = \Sigma \cup \{\mathrm{p}^\sharp, \mathrm{fac}^\sharp\}$ and two dependency pairs:

$$\begin{aligned} \mathrm{fac}^\sharp(\mathrm{s}(x)) &\to \mathrm{fac}^\sharp(\mathrm{p}(\mathrm{s}(x))) \\ \mathrm{fac}^\sharp(\mathrm{s}(x)) &\to \mathrm{p}^\sharp(\mathrm{s}(x)) \end{aligned}$$

We will prove termination of this example in the following section. $\lhd$

3 Monotone Algebras

We will now introduce the definitions and results of monotone algebras, following the presentation of [13].

Definition 3.1. [Monotonicity] Let A be a non-empty set. An operation $[f] : A \times \cdots \times A \to A$ is *monotone* with respect to a binary relation $\to$ on A if for all $a_1, \ldots, a_i, a_i', \ldots a_n \in A$ with $a_i \to a_i'$ we have

$$[f](a_1, \ldots, a_i, \ldots, a_n) \;\to\; [f](a_1, \ldots, a_i', \ldots, a_n) \qquad \diamond$$

Definition 3.2. [Σ-algebra] A Σ-*algebra*, $(A, \{f_A\}_{f \in \Sigma})$ consists of a non-empty set A together with a map $[f_A] : A^n \to A$ for every $f \in \Sigma$, where n is the arity of f. $\diamond$

Definition 3.3. [Weakly monotone Σ-algebra] Let $\mathcal{R}$ be a TRS over a signature Σ. A well-founded *weakly monotone* Σ*-algebra* is a quadruple $\mathcal{A} = (A, \{f_A\}_{f\in\Sigma}, >, \gtrsim)$ such that:

- $(A, \{f_A\}_{f\in\Sigma})$ is a Σ-algebra,
- all algebra operations are weakly monotone, *i.e.*, monotone with respect to $\gtrsim$,
- $>$ is a well-founded relation on A, and
- relations $\gtrsim$ and $>$ are compatible, that is: $> \cdot \gtrsim \subseteq >$ or $\gtrsim \cdot > \subseteq >$.

An *extended monotone* Σ*-algebra* $(A, \{f_A\}_{f\in\Sigma}, >, \gtrsim)$ is a weakly monotone Σ-algebra $(A, \{f_A\}_{f\in\Sigma}, >, \gtrsim)$ in which moreover for every $f \in \Sigma$ the operation $[f]$ is strictly monotone, *i.e.*, monotone with respect to $>$. ◇

Definition 3.4. For a weakly monotone Σ-algebra $\mathcal{A} = (A, \{f_A\}_{f\in\Sigma}, >, \gtrsim)$ we extend the order $\gtrsim$ on A to an order $\gtrsim_\alpha$ on terms, as

$$t \gtrsim_\alpha u \iff \forall_{\alpha:\mathcal{V}\to A} : \ [t]_\alpha \gtrsim [u]_\alpha$$

$>$ is extended to $>_\alpha$ in a similar way. ◇

Now we present a slight variant of the main theorem from [13], for proving relative (top)-termination with monotone algebras:

Theorem 3.5. Let $\mathcal{R}, \mathcal{R}', \mathcal{S}, \mathcal{S}'$ be TRSs over a signature Σ.

(a) Let $(A, [\cdot], >, \gtrsim)$ be an extended monotone algebra such that: $[\ell] \gtrsim_\alpha [r]$ for every rule $\ell \to r \in \mathcal{R} \cup \mathcal{S}$ and $[\ell] >_\alpha [r]$ for every rule $\ell \to r \in \mathcal{R}' \cup \mathcal{S}'$. Then $\mathrm{SN}(\mathcal{R}/\mathcal{S})$ implies $\mathrm{SN}(\mathcal{R} \cup \mathcal{R}'/\mathcal{S} \cup \mathcal{S}')$.

(b) Let $(A, [\cdot], >, \gtrsim)$ be a weakly monotone algebra such that: $[\ell] \gtrsim_\alpha [r]$ for every rule $\ell \to r \in \mathcal{R} \cup \mathcal{S}$ and $[\ell] >_\alpha [r]$ for every rule $\ell \to r \in \mathcal{R}'$. Then $\mathrm{SN}(\mathcal{R}_{\mathrm{top}}/\mathcal{S})$ implies $\mathrm{SN}(\mathcal{R}_{\mathrm{top}} \cup \mathcal{R}'_{\mathrm{top}}/\mathcal{S})$. □

We will illustrate the application of this theorem on a simple example using the matrix interpretation method [13].

Example 3.6. Consider the TRS from Example 2.3. We will show how Theorem 3.5a can be applied to this TRS in order to simplify the related termination problem.

For that we first need to choose a suitable monotone algebra. For the domain A we take vectors over $\mathbb{N}$ of length 2 with the following orders:

$$(u_1, u_2) \gtrsim (v_1, v_2) \iff u_1 \geq v_1 \land u_2 \geq v_2$$
$$(u_1, u_2) > (v_1, v_2) \iff u_1 > v_1 \land u_2 \geq v_2$$

Compatibility of those orders and well-foundedness of $>$ are immediate. For interpretations we take linear functions over this domain, so an n-ary symbol f is interpreted by:

$$[\mathrm{f}(\vec{x}_1, \ldots, \vec{x}_n)] = M_1\vec{x}_1 + \ldots + M_n\vec{x}_n + \vec{c} \tag{1}$$

where $\vec{x}_1, \ldots, \vec{x}_n, \vec{c} \in \mathbb{N}^2$ and $M_1, \ldots, M_n \in \mathbb{N}^{2\times2}$. So an interpretation of a symbol of arity n is given by n square matrices $M_1, \ldots, M_n$ of size 2×2 and one constant vector $\vec{c}$ of dimension 2. Such interpretations are always weakly monotone. We want to use Theorem 3.5a so we need an extended monotone algebra which requires strict monotonicity. For that we need some restrictions and it is easy to see that it can be guaranteed by requiring $M_i[1,1] > 0$ for $1 \le i \le n$.

Now our goal is to prove termination of the given TRS and we will do that by applying Theorem 3.5a instantiated with the extended monotone algebra that we just introduced. We recall that termination is a special case of relative termination so we will apply this theorem with $\mathcal{S} = \mathcal{S}' = \emptyset$. We need to find interpretations for all $f \in \Sigma$. Typically this is done automatically by dedicated tools — we will address this issue in Section 11. One of such tools, TPA [24], applied on this TRS generated the following interpretations:

$$[0] = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \qquad [\mathrm{fac}(\vec{x})] = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

$$[\mathrm{p}(\vec{x})] = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \vec{x} \qquad [\mathrm{times}(\vec{x}, \vec{y})] = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \vec{x} + \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} \vec{y}$$

$$[\mathrm{s}(\vec{x})] = \begin{pmatrix} 1 & 1 \\ 3 & 3 \end{pmatrix} \vec{x}$$

Note that the lack of the constant vector $\vec{c}$ in some of the above interpretations indicates that this constant is the zero vector $(0, 0)$.

Let us compute interpretations of the left and right hand side of the second rule $\mathrm{fac}(0) \to \mathrm{s}(0)$.

$$[\mathrm{fac}(0)] = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \end{pmatrix} \qquad [\mathrm{s}(0)] = \begin{pmatrix} 1 & 1 \\ 3 & 3 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$$

So using our order on vectors we obtain $[\mathrm{fac}(0)] > [\mathrm{s}(0)]$. In a similar way we compute interpretations for the remaining rules. Note that the fact that we restricted ourselves to linear functions means that their composition is linear too and hence all the interpretations that we obtain are of the same shape as in Equation (1).

$$\begin{aligned} [\mathrm{p}(\mathrm{s}(x))] &= \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 3 & 3 \end{pmatrix} \vec{x} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \vec{x} \\ [x] &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \vec{x} \end{aligned}$$

$$[\text{times}(\text{s}(x), \text{fac}(\text{p}(\text{s}(x))))] = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} [\text{s}(x)] + \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} [\text{fac}(\text{p}(\text{s}(x)))] = \begin{pmatrix} 7 & 7 \\ 0 & 0 \end{pmatrix} \vec{x}$$

$$[\text{fac}(\text{s}(x))] = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 3 & 3 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 7 & 7 \\ 6 & 6 \end{pmatrix} \vec{x} + \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

For both of the rules it is easy to see that regardless of the assignment to the vector $\vec{x}$ we always obtain that the interpretation of the left hand side is bigger or equal than that of the right hand side of a rule.

All in all we apply Theorem 3.5a with

$$\begin{aligned} \mathcal{R} &= \{\ \text{p}(\text{s}(x)) \to x, \quad \text{fac}(\text{s}(x)) \to \text{times}(\text{s}(x), \text{fac}(\text{p}(\text{s}(x))))\ \} \\ \mathcal{R}' &= \{\ \text{fac}(0) \to \text{s}(0)\ \} \\ \mathcal{S} = \mathcal{S}' &= \emptyset \end{aligned}$$

This allows us to remove the second rule and conclude termination of the whole system from termination of $\mathcal{R}$ only, which is easy to show, for instance, with the standard method of polynomial interpretations in combination with the dependency pair method. $\triangleleft$

4 The Arctic Semiring

A *commutative semiring* [18] consists of a carrier D, two designated elements $d_0, d_1 \in D$ and two binary operations $\oplus, \otimes$ on D, called semiring addition and semiring multiplication, respectively, such that both $(D, d_0, \oplus)$ and $(D, d_1, \otimes)$ are commutative monoids and multiplication distributes over addition: $\forall_{x,y,z \in D} :\ x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$.

One example of a semiring are the natural numbers with the standard operations $\oplus = +$ and $\otimes = *$. We will need the *arctic semiring* (also called the *max/plus algebra*) [15] with carrier $\mathbb{A}_{\mathbb{N}} \equiv \{-\infty\} \cup \mathbb{N}$, where semiring addition is the max operation with neutral element $-\infty$ and semiring multiplication is the standard plus operation with neutral element 0, so:

$$\begin{array}{llll} x \oplus y = y & \text{if } x = -\infty, & x \otimes y = -\infty & \text{if } x = -\infty \text{ or } y = -\infty, \\ x \oplus y = x & \text{if } y = -\infty, & x \otimes y = x + y & \text{otherwise}, \\ x \oplus y = \max(x, y) & \text{otherwise}. & & \end{array}$$

We also consider these operations for arctic numbers *below zero* (*i.e.*, *arctic integers*), that is, on the carrier $\mathbb{A}_{\mathbb{Z}} \equiv \{-\infty\} \cup \mathbb{Z}$.

For any semiring D, we can consider the space of linear functions (square matrices) on n-dimensional vectors over D. These functions (matrices) again form a semiring (though a non-commutative one), and indeed we write $\oplus$ and $\otimes$ for its operations as well.

A semiring is *ordered* [14] by $\geq$ if $\geq$ is a partial order compatible with the operations: $\forall_{x,y,z} :\ x \geq y \implies x \oplus z \geq y \oplus z$ and $\forall_{x,y,z} :\ x \geq y \implies x \otimes z \geq y \otimes z$.

The standard semiring of natural numbers is ordered by the standard $\geq$ relation. The semiring of arctic naturals and arctic integers is ordered by $\geq$, being the reflexive closure of $>$ defined as $\ldots > 1 > 0 > -1 > \ldots > -\infty$. Note that standard integers with standard operations form a semiring but it is not ordered in this sense, as we have for instance $1 \geq 0$ but $1 * (-1) = -1 \not\geq 0 = 0 * (-1)$.

We remark that $\geq$ is the "natural" ordering for the arctic semiring, in the following sense: $x \geq y \iff x = x \oplus y$. Since arctic addition is idempotent, some properties of $\geq$ follow easily, like the one presented below.

Lemma 4.1. For arctic integers a_1, a_2, b_1, b_2, if $a_1 \geq a_2 \wedge b_1 \geq b_2$, then $a_1 \oplus b_1 \geq a_2 \oplus b_2$ and $a_1 \otimes b_1 \geq a_2 \otimes b_2$. □

Arctic addition (*i.e.*, the max operation) is not strictly monotone in single arguments: we have, *e.g.*, $5 > 3$ but $5 \oplus 6 = 6 \not> 6 = 3 \oplus 6$. It is, however, "half strict" in the following sense: a strict increase in both arguments simultaneously gives a strict increase in the result, *i.e.*, $a_1 > b_1$ and $a_2 > b_2$ implies $a_1 \oplus a_2 > b_1 \oplus b_2$. There is one exception: arctic addition is obviously strict if one argument is arctic zero, *i.e.*, $-\infty$. This is the motivation for introducing the following relation:

$$a \gg b \iff (a > b) \vee (a = b = -\infty)$$

Below we present some of its properties needed later:

Lemma 4.2. For arctic integers a, a_1, a_2, b_1, b_2,

1. if $a_1 \gg a_2 \wedge b_1 \gg b_2$, then $a_1 \oplus b_1 \gg a_2 \oplus b_2$.

2. if $a_1 \gg a_2 \wedge b_1 \geq b_2$, then $a_1 \otimes b_1 \gg a_2 \otimes b_2$.

3. if $b_1 \gg b_2$, then $a \otimes b_1 \gg a \otimes b_2$.

Proof. By simple case analysis (whether an element is $-\infty$ or not) and some properties of addition and max operations over integers. □

Note that properties 2 and 3 in the above lemma would not hold if we were to replace $\gg$ with $>$.

An arctic natural number $a \in \mathbb{A}_{\mathbb{N}}$ is called *finite* if $a \neq -\infty$. An arctic integer $a \in \mathbb{A}_{\mathbb{Z}}$ is called *positive* if $a \geq 0$ (that excludes $-\infty$ and negative numbers).

Lemma 4.3. Let $m, n \in \mathbb{A}_{\mathbb{N}}$ and $a, b \in \mathbb{A}_{\mathbb{Z}}$, then:

1. if m is finite and n arbitrary, then $m \oplus n$ is finite.

2. if a is positive and b arbitrary, then $a \oplus b$ is positive.

3. if m and n are finite, then $m \otimes n$ is finite.

Proof. Direct computation. □

By analogy to linear algebra over $(\mathbb{N}, +, \cdot)$, we consider sequences (vectors) and rectangular arrays (matrices) of arctic numbers. Sequences $\mathbb{A}^d$ form a semimodule over $\mathbb{A}$, the elements of which we call *arctic vectors*. Operations in the semimodule are $\oplus : \mathbb{A}^d \times \mathbb{A}^d \to \mathbb{A}$ defined by component-wise addition and component-wise multiplication by a scalar value $\otimes : \mathbb{A} \times \mathbb{A}^d \to \mathbb{A}^d$. Then, arctic matrices represent linear functions from vectors to vectors: An arctic matrix M maps a (column) vector $\vec{x}$ to a (column) vector $M \otimes \vec{x}$ and this mapping is linear: $M \otimes (\vec{x} \oplus \vec{y}) = M \otimes \vec{x} \oplus M \otimes \vec{y}$.

We can combine those linear functions (matrices) in the usual way, and we re-use symbols $\oplus$ and $\otimes$ for matrix addition and matrix multiplication. Square arctic matrices form a non-commutative semiring with these operations. *E.g.* the 3×3 identity matrix is

$$\begin{pmatrix} 0 & -\infty & -\infty \\ -\infty & 0 & -\infty \\ -\infty & -\infty & 0 \end{pmatrix}$$

We will be interested in linear functions over arctic vectors of the following shape:

Definition 4.4. Let $\mathbb{A}$ be an arctic domain (so either arctic naturals $\mathbb{A}_{\mathbb{N}}$ or arctic integers $\mathbb{A}_{\mathbb{Z}}$). An *(n-ary) arctic linear function (over $\mathbb{A}$)* (with linear factors $M_1, \ldots, M_n$ and an absolute part $\vec{c}$) is a function of the following shape:

$$f(\vec{x}_1, \ldots, \vec{x}_n) = M_1 \otimes \vec{x}_1 \oplus \ldots \oplus M_n \otimes \vec{x}_n \oplus \vec{c}$$

So an arctic linear function over column vectors $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{A}^d$ is described by a column vector $\vec{c} \in \mathbb{A}^d$ and square matrices $M_1, \ldots, M_n \in \mathbb{A}^{d \times d}$. $\diamond$

Note that for brevity from now on we will omit the semiring multiplication sign $\otimes$ and use the following notation for arctic linear functions:

$$f(\vec{x}_1, \ldots, \vec{x}_n) = M_1 \vec{x}_1 \oplus \ldots \oplus M_n \vec{x}_n \oplus \vec{c}$$

Example 4.5. Consider a linear function:

$$f(\vec{x}, \vec{y}) = \begin{pmatrix} 1 & -\infty \\ 0 & -\infty \end{pmatrix} \vec{x} \oplus \begin{pmatrix} -\infty & -\infty \\ 0 & 1 \end{pmatrix} \vec{y} \oplus \begin{pmatrix} -\infty \\ 0 \end{pmatrix}$$

Evaluation of this function on some exemplary arguments yields:

$$f(\begin{pmatrix} -\infty \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ -\infty \end{pmatrix}) = \begin{pmatrix} 1 & -\infty \\ 0 & -\infty \end{pmatrix} \begin{pmatrix} -\infty \\ 0 \end{pmatrix} \oplus \begin{pmatrix} -\infty & -\infty \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ -\infty \end{pmatrix} \oplus \begin{pmatrix} -\infty \\ 0 \end{pmatrix} = \begin{pmatrix} -\infty \\ 1 \end{pmatrix}$$

$\lhd$

5 Weighted Tree Automata

In this section we instantiate the monotone algebra framework with the initial algebraic semantics of weighted tree automata of a certain shape. This allows to put the matrix method (Example 3.6) into perspective, and it also is the basis for the generalization to arctic matrices (following sections).

A weighted tree automaton [10, 9] is a finite-state device that computes a mapping from trees over some signature into some semiring. This computational model is obtained from classical (Boolean) automata by assigning weights to transitions.

Formally, a D-weighted tree automaton is a tuple $A = (D, Q, \Sigma, \delta, F)$ where D is a semiring, Q is a finite set of states, Σ is a ranked signature, δ is a transition function that assigns to any k-ary symbol $f \in \Sigma_k$ a function $\delta_f : Q^k \times Q \to D$ and F is a mapping $Q \to D$. The idea is that $\delta_f(q_1, \ldots, q_k, q)$ gives the weight of the transition from $(q_1, \ldots, q_k)$ to q, and $F(q)$ gives the weight of the final state q.

We use the following tree automaton as an ongoing example for this section. This example is related to the matrix interpretation shown in Example 3.6, in a way that will be made precise later.

Example 5.1. For the signature $\Sigma = \{0/0, \mathrm{p}/1, \mathrm{s}/1, \mathrm{fac}/1, \mathrm{times}/2\}$ (from Example 2.3), a $\mathbb{N}$-weighted tree automaton with states $Q = \{a, b, c\}$ is given by:

$$\begin{aligned}
&(0) && \delta_0(b) = 2, \delta_0(c) = 1,\\
&(\mathrm{p}) && \delta_{\mathrm{p}}(a, a) = \delta_{\mathrm{p}}(a, b) = \delta_{\mathrm{p}}(c, c) = 1,\\
&(\mathrm{s}) && \delta_{\mathrm{s}}(a, a) = \delta_{\mathrm{s}}(b, a) = 1,\ \delta_{\mathrm{s}}(a, b) = \delta_{\mathrm{s}}(b, b) = 3,\ \delta_{\mathrm{s}}(c, c) = 1,\\
&(\mathrm{fac}) && \delta_{\mathrm{fac}}(a, a) = 1,\ \delta_{\mathrm{fac}}(b, a) = \delta_{\mathrm{fac}}(b, b) = \delta_{\mathrm{fac}}(c, b) = 2,\ \delta_{\mathrm{fac}}(c, c) = 1,\\
&(\mathrm{times}) && \delta_{\mathrm{times}}(a, c, a) = 1,\ \delta_{\mathrm{times}}(c, b, a) = 2,\ \delta_{\mathrm{times}}(c, c, c) = 1,
\end{aligned}$$

$F(a) = 1$, and all other transitions have weight 0. ◁

For any tree $t = f(t_1, \ldots, t_k)$ over Σ, and $q \in Q$, denote by $A_q(t)$ the weight that A assigns to t in state q:

$$A_q(t) = \sum\{\delta_f(q_1, \ldots, q_k, q) \cdot A_{q_1}(t_1) \cdot \ldots \cdot A_{q_k}(t_k) \mid q_1, \ldots, q_k, q \in Q\}$$

and the total weight $A(t)$ is $\sum\{F(q) \cdot A_q(t) \mid q \in Q\}$.

Example 5.2. (continued) We find $A_a(0) = 0, A_b(0) = 2, A_c(0) = 1$, since the symbol 0 is nullary and thus $A_q(0) = \delta_0(q)$. Then, for example, $A_b(\mathrm{s}(0)) = \delta_{\mathrm{s}}(a, b) \cdot A_a(0) + \delta_{\mathrm{s}}(b, b) \cdot A_b(0) + \delta_{\mathrm{s}}(c, b) \cdot A_c(0) = 3 \cdot 0 + 3 \cdot 2 + 0 \cdot 1 = 6$. ◁

This is called *initial algebra semantics* of a tree automaton. Indeed, the automaton is a Σ-algebra where the carrier set consists of weight vectors, indexed by states. Let $V = (Q \to D)$ be the set of such vectors. Then for each k-ary symbol f, the transition δ_f computes a function $[\delta_f] : V^k \to V$ by $[\delta_f](v_1, \ldots, v_k) = w$ where

$$w_q = \sum\{\delta_f(q_1, \ldots, q_k, q) \cdot v_{1,q_1} \cdot \ldots \cdot v_{k,q_k} \mid q_1, \ldots, q_k \in Q\}.$$

Example 5.3. (continued) For the unary fac symbol, we have the unary function

$$[\mathrm{fac}] : V^1 \to V : (v_{1,a},\, v_{1,b},\, v_{1,c}) \mapsto (v_{1,a} + 2v_{1,b},\, 2v_{1,b} + 2v_{1,c},\, v_{1,c}).$$

Since 0 is a nullary symbol, its interpretation [0] is of type $V^0 \to V$, that is, it takes an empty argument list and produces a vector $[0] = (0, 2, 1)$. ◁

By distributivity in the semiring, each function $[\delta_f]$ is multilinear (linear in each argument):

$$\begin{array}{rl} & [\delta_f](\ldots, v_{i-1}, v_i + v'_i, v_{i+1}, \ldots) \\ = & [\delta_f](\ldots, v_{i-1}, v_i, v_{i+1}, \ldots) + [\delta_f](\ldots, v_{i-1}, v'_i, v_{i+1}, \ldots). \end{array}$$

For a given tree automaton A over Σ, the collection $\{[\delta_f] \mid f \in \Sigma\}$ constitutes an algebra with carrier V. Therefore, interpretations of function symbols $[\delta_f]$ can be lifted to interpretations of terms.

Example 5.4. (continued) In the algebra of the automaton:

$$[\mathrm{fac}(0)] = (0 + 2 \cdot 2,\ 2 \cdot 2 + 2 \cdot 1,\ 1) = (4, 6, 1). \qquad \triangleleft$$

It is convenient to think of elements of V as column vectors, and F as a row vector. Then $A(t)$ is the dot product $F \cdot (A_1(t), \ldots, A_{|Q|}(t))^T$.

Example 5.5. (continued) $A(\mathrm{fac}(0)) = (1, 0, 0) \cdot (4, 6, 1)^T = 4.$ $\triangleleft$

With these preparations, we can apply the monotone algebra approach for proving termination of term rewriting, where the algebra is given by a finite weighted tree automaton.

In order to obtain a method that can be automated easily, we restrict the shape of the automata transitions, so that the interpretation of each function symbol is a sum of linear functions in single arguments, and an absolute part, cf. Equation 1.

Definition 5.6. A weighted tree automaton $A = (D, Q, \Sigma, \delta, F)$ is called *path-separated* with initial state $i \in Q$ if for each k-ary transition with non-zero weight we have that

- at most one of the initial k arguments is $\neq i$:

$$\delta_f(q_1, \ldots, q_k, q) \neq 0 \Rightarrow \exists_{\leq 1} 1 \leq j \leq k : q_j \neq i.$$

- if the target is i, then all sources are i, and the weight is unit:

$$\delta_f(q_1, \ldots, q_k, i) = (\text{if } q_1 = \ldots = q_k = i \text{ then } 1 \text{ else } 0). \qquad \diamond$$

Example 5.7. (continued) The given automaton is path-separated, with $i = c$ as the initial state. $\triangleleft$

Proposition 5.8. The following conditions are equivalent for a weighted tree automaton $A = (D, Q, \Sigma, \delta, F)$:

- A is path-separated with initial state i,
- each action of $[\delta_f]$ has the following form:

$$[\delta_f](\vec{v}_1, \ldots, \vec{v}_k) \mapsto M_1 \cdot \vec{v}_1 + \ldots + M_k \cdot \vec{v}_k + \vec{a},$$

where M_j are square matrices of dimension $|Q| \times |Q|$, with all entries in row i and in column i are zero; and $\vec{a}$ is a vector, with entry one at position i.

Proof. Let A be path-separated with an initial state i. For any $f \in \Sigma_k$, we have $\vec{a}[q] = \delta_f(i, \ldots, i, q)$, if none of the first k arguments is $\neq i$, and $M_j[q, p] = \delta_f(i, \ldots, i, p, i, \ldots, i, q)$ where p is the single non-i state among the first k arguments. By the path-separation restriction, these cases cover all possible transitions. $\square$

Example 5.9. (continued) $[\mathrm{fac}](\vec{v}) = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \vec{v} + \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}$. $\lhd$

Under these conditions, for each t we have $A_i(t) = 1$. So we drop the entry at i in $\vec{a}$, and also each row i and each column i in M_j. Then by Proposition 5.8, a path-separated tree automaton corresponds to a matrix interpretation of shape (1) and vice-versa.

Example 5.10. (continued) $[\mathrm{fac}](\vec{v}) = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} \vec{v} + \begin{pmatrix} 0 \\ 2 \end{pmatrix}$. $\lhd$

We call these tree automata path-separated because their semantics can be computed as the sum of matrix products along all paths of the input, and the values along different paths do not influence each other.

Here, a *path* is a sequence of function symbols with directions. Formally, for any term $t = f(t_1, \ldots, t_k)$, we define

$$\mathrm{paths}(t) = \{f_0\} \cup \{f_i \circ p \mid 1 \leq i \leq k, p \in \mathrm{paths}(t_i)\}.$$

This is a mapping from $\mathcal{T}(\Sigma)$ to nonempty sequences of pairs of symbols and numbers, with a pair (f, i) denoted by f_i; actually to a subset of $P_\Sigma = (\Sigma \times \mathbb{N}_{>0})^*(\Sigma \times \{0\})$.

Example 5.11.

$$\mathrm{paths}(\mathrm{times}(0, \mathrm{fac}(0))) = \{\mathrm{times}_0,\ \mathrm{times}_1 \circ 0_0,\ \mathrm{times}_2 \circ \mathrm{fac}_0,\ \mathrm{times}_2 \circ \mathrm{fac}_1 \circ 0_0\}. \lhd$$

For a path-separated tree automaton $A = (D, Q, \Sigma, \delta, F)$ and each k-ary symbol f, where δ_f is as in (1), define a mapping $[\cdot]$ from paths in P_Σ to vectors by $[f_0] = \vec{c}$ and $[f_i \circ p] = M_i \cdot [p]$. Then it follows from distributivity of addition (of vectors) over multiplication (with matrices) that for each term t,

$$A(t) = \sum \{F \cdot [p] \mid p \in \mathrm{paths}(t)\}.$$

This illustrates why we call these automata path-separated.

We briefly comment on the effect of the path-separation restriction. Consider a signature with a binary symbol g. A matrix interpretation of dimension one interprets g with a function $(x_1, x_2) \mapsto m_1 x_1 + m_2 x_2 + a$. This corresponds to a path-separated $\mathbb{N}$-weighted automaton with just two states, one of which being the initial state.

The general form of a transition function of a tree automaton with two states, one of them initial, is $(x_1, x_2) \mapsto m_{12} x_1 x_2 + m_1 x_1 + m_2 x_2 + a$. The "$m_{12} x_1 x_2$"

component cannot be part of a path-separated tree automaton's transition function. We really lose expressiveness here, *e.g.*, the tree automaton's transition $(x_1, x_2) \mapsto x_1 x_2$ cannot be expressed by matrix interpretations, even with additional states, since it grows faster (doubly exponential) than any matrix-representable function (exponential).

On the other hand, if the signature contains no symbols of arity > 1, then each tree automaton has an equivalent path-separated automaton (of size $|Q| + 1$, since in general we need to add the initial state).

6 Full Arctic Termination

In this section, we instantiate the monotone algebra approach for proving termination of rewriting by using algebras defined by path-separated arctic tree automata.

The algebra domain consists of vectors of arctic naturals, $\mathbb{A}_{\mathbb{N}}^d$. Every $f \in \Sigma$ will be interpreted by an arctic linear function (Definition 4.4) and we will refer to such interpretations as *arctic Σ-interpretations.*

We define orders on arctic vectors and matrices by taking a point-wise extension of the orders $\gg$ and $\geq$ introduced in Section 4. We will use the same notation, *i.e.*, $\gg$ and $\geq$, for those lifted orders. Now we take the vector extension of $\gg$ and $\geq$ as, respectively, the strict and non-strict order of the algebra. Note that they are compatible, *i.e.*, $\gg \cdot \geq \subseteq \gg$. However with this choice we do not get well-foundedness of the strict order as $-\infty \gg -\infty$. Therefore we will restrict first components of vectors to finite elements (*i.e.*, elements different from $-\infty$, as introduced before Lemma 4.3). Effectively our algebra becomes $(\mathbb{N} \times \mathbb{A}_{\mathbb{N}}^{d-1}, \{f_A\}_{f \in \Sigma}, \gg, \geq)$.

We will consider arctic linear functions over the domain of our algebra, so we must make sure that evaluation of those functions stays within the domain, *i.e.*, that the first vector component is finite. The following definition and lemma address this issue.

Definition 6.1. An n-ary arctic linear function

$$f(\vec{x}_1, \ldots, \vec{x}_n) = M_1 \vec{x}_1 \oplus \ldots \oplus M_n \vec{x}_n \oplus \vec{c}$$

over $\mathbb{A}_{\mathbb{N}}$ is called *somewhere finite* if:

- $\vec{c}[1]$ is finite, or
- $M_i[1, 1]$ is finite for some $1 \leq i \leq n$. ◇

Lemma 6.2. Let f be an n-ary arctic linear function over $\mathbb{A}_{\mathbb{N}}$, $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{N} \times \mathbb{A}_{\mathbb{N}}^{d-1}$ and $\vec{v} = f(\vec{x}_1, \ldots, \vec{x}_n)$. If f is somewhere finite then $\vec{v}[1]$ is finite.

Proof.

$$f(\vec{x}_1, \ldots, \vec{x}_n)[1] = (M_1 \vec{x}_1)[1] \oplus \ldots \oplus (M_n \vec{x}_n)[1] \oplus \vec{c}[1] \tag{2}$$

Since f is somewhere finite we have:

- $\vec{c}[1]$ is finite, or

- for some $1 \leq i \leq n$, $M_i[1,1]$ is finite but then $(M_i\vec{x}_i)[1] = M_i[1,1]\vec{x}_i[1] \oplus \ldots \oplus M_i[1,d]\vec{x}_i[d]$, which is finite by Lemma 4.3, as $M_i[1,1]$ is finite.

In either case one of the summands in Equation 2 is finite, making the whole expression finite by Lemma 4.3. □

To apply the monotone algebra theorem, Theorem 3.5, we will need to compare arctic linear functions, *i.e.*, we will need some properties ensuring that, for arbitrary arguments, one arctic function always gives a vector that is greater (or greater equal) than the result of application of some other arctic functions to the same arguments. This is addressed in the following lemma, which is the arctic counterpart of the absolute positiveness criterion used for polynomial interpretations [23].

Definition 6.3. Let f, g be arctic linear functions over $\mathbb{A}$:

$$f(\vec{x}_1, \ldots, \vec{x}_n) = M_1\vec{x}_1 \oplus \ldots \oplus M_n\vec{x}_n \oplus \vec{c}$$
$$g(\vec{x}_1, \ldots, \vec{x}_n) = N_1\vec{x}_1 \oplus \ldots \oplus N_n\vec{x}_n \oplus \vec{d}$$

We will say that f is greater (resp. greater equal) than g, notation $f \gg_\lambda g$ (resp. $f \geq_\lambda g$) iff:

- $c \gg d$ (resp. $c \geq d$) and
- $\forall_{1 \leq i \leq n} : \ M_i \gg N_i$ (resp. $M_i \geq N_i$). ⋄

We will justify the above definition in Lemma 6.5, but first we need an auxiliary result:

Lemma 6.4. Let $M, N \in \mathbb{A}^{d \times d}$ and $\vec{x}, \vec{y} \in \mathbb{A}^d$.

1. If $M \gg N$ and $\vec{x} \geq \vec{y}$ then $M\vec{x} \gg N\vec{y}$.
2. If $M \geq N$ and $\vec{x} \geq \vec{y}$ then $M\vec{x} \geq N\vec{y}$.

Proof. Immediate using Lemma 4.1 and the first two properties of Lemma 4.2. □

Lemma 6.5. Let f, g be arctic linear functions over $\mathbb{A}$ and let $\vec{x}_1, \ldots, \vec{x}_n$ be arbitrary vectors.

1. If $f \gg_\lambda g$ then $f(\vec{x}_1, \ldots, \vec{x}_n) \gg g(\vec{x}_1, \ldots, \vec{x}_n)$.
2. If $f \geq_\lambda g$ then $f(\vec{x}_1, \ldots, \vec{x}_n) \geq g(\vec{x}_1, \ldots, \vec{x}_n)$.

Proof. We will prove only the first case — the other one is analogous.

$$f(\vec{x}_1, \ldots, \vec{x}_n) = M_1\vec{x}_1 \oplus \ldots \oplus M_n\vec{x}_n \oplus \vec{c}$$
$$g(\vec{x}_1, \ldots, \vec{x}_n) = N_1\vec{x}_1 \oplus \ldots \oplus N_n\vec{x}_n \oplus \vec{d}$$

We have $\vec{c} \gg \vec{d}$ and $\forall_{1 \leq i \leq n} : \ M_i \gg N_i$ as $f \gg_\lambda g$ and hence $M_i\vec{x}_i \gg N_i\vec{x}_i$ by Lemma 6.4. So every vector summand of the evaluation of f is related by $\gg$ with a corresponding summand of g and we conclude by Lemma 4.1. □

Clearly arctic linear functions are weakly monotone (because so is the max operation, *i.e.*, arctic addition) and we establish this property in the following lemma.

Lemma 6.6. Every arctic linear function f over $\mathbb{A}$ is monotone with respect to $\geq$.

Proof. Let $x_i \geq x'_i$. We have:

$$f(\vec{x}_1, \ldots, \vec{x}_i, \ldots, \vec{x}_n) = M_1\vec{x}_1 \oplus \ldots \oplus M_i\vec{x}_i \oplus \ldots \oplus M_n\vec{x}_n \oplus \vec{c}$$
$$f(\vec{x}_1, \ldots, \vec{x}'_i, \ldots, \vec{x}_n) = M_1\vec{x}_1 \oplus \ldots \oplus M_i\vec{x}'_i \oplus \ldots \oplus M_n\vec{x}_n \oplus \vec{c}$$

All the summands are equal except for the one corresponding to the i'th argument, where we have $M_i\vec{x}_i \geq M_i\vec{x}'_i$ by Lemma 6.4 and we conclude

$$f(\vec{x}_1, \ldots, \vec{x}_i, \ldots, \vec{x}_n) \geq f(\vec{x}_1, \ldots, \vec{x}'_i, \ldots, \vec{x}_n)$$

by Lemma 4.1. □

However, to obtain an extended weakly monotone algebra, and prove full termination using it, we need strict monotonicity. As remarked in Section 4, arctic addition is not strictly monotone. Hence functions introduced in Definition 4.4 are strictly monotone only if the $\oplus$ operation is essentially redundant; for instance it is immediately lost for functions of more than one argument. This essentially restricts our method to unary rewriting [35]; a proper extension of string rewriting. As such, it had been described in [37] and had been applied by Matchbox in the 2007 termination competition. The following theorem provides a termination criterion for such systems. In the next section we will look at top termination problems, which will allow us to lift this restriction and consider arbitrary TRSs.

Theorem 6.7. Let $\mathcal{R}, \mathcal{R}', \mathcal{S}, \mathcal{S}'$ be TRSs over a signature Σ and $[\cdot]$ be an arctic Σ-interpretation over $\mathbb{A}_{\mathbb{N}}$. If:

- every function symbol has arity at most 1,
- every constant $a \in \Sigma$ is interpreted by $[a] = \vec{c}$ with $\vec{c}[1]$ finite,
- every unary symbol $s \in \Sigma$ is interpreted by $[s(\vec{x})] = M \otimes \vec{x}$ with $M[1,1]$ finite,
- $[\ell] \geq_\lambda [r]$ for every rule $\ell \to r \in \mathcal{R} \cup \mathcal{S}$,
- $[\ell] \gg_\lambda [r]$ for every rule $\ell \to r \in \mathcal{R}' \cup \mathcal{S}'$ and
- $\mathrm{SN}(\mathcal{R}/\mathcal{S})$.

Then $\mathrm{SN}(\mathcal{R} \cup \mathcal{R}'/\mathcal{S} \cup \mathcal{S}')$.

Proof. By Theorem 3.5a. Note that, by Lemma 6.5, $[\ell] \geq_\lambda [r]$ (resp. $[\ell] \gg_\lambda [r]$) implies $[\ell] \geq_\alpha [r]$ (resp. $[\ell] \gg_\alpha [r]$). So we only need to show that $(\mathbb{N} \times \mathbb{A}_{\mathbb{N}}^{d-1}, [\cdot], \gg, \geq)$ is an extended monotone algebra. The order $\gg$ is well-founded on this domain as with every decrease we get a decrease in the first component of the vector, which belongs to $\mathbb{N}$. Arctic functions are always weakly monotone by Lemma 6.6 and it is an easy observation that, due to the first three premises of this theorem, the interpretations that we allow here are strictly monotone. Finally we stay within the domain by Lemma 6.2 as the interpretation functions $[f]$ that we restrict to are somewhere finite (again by the first three assumptions). $\square$

We now present an example illustrating this theorem.

Example 6.8. The relative termination problem `SRS/Waldmann/r2` is

$$\{\mathrm{c\,a\,c} \to \epsilon,\ \mathrm{a\,c\,a} \to \mathrm{a}^4\ \ /\ \epsilon \to \mathrm{c}^4\}$$

In the 2007 termination competition, it had been solved by Jambox [12] via “self labeling” and by Matchbox via essentially the following arctic proof.

We use the following arctic interpretation

$$[\mathrm{a}](\vec{x}) = \begin{pmatrix} 0 & 0 & -\infty \\ 0 & 0 & -\infty \\ 1 & 1 & 0 \end{pmatrix} \vec{x} \qquad [\mathrm{c}](\vec{x}) = \begin{pmatrix} 0 & -\infty & -\infty \\ -\infty & -\infty & 0 \\ -\infty & 0 & -\infty \end{pmatrix} \vec{x}$$

It is immediate that [c] is a permutation (it swaps the second and third component of its argument vector), so $[\mathrm{c}]^2 = [\mathrm{c}]^4$ is the identity and we have $[\epsilon] = [\mathrm{c}]^4$. A short calculation shows that [a] is idempotent, so $[\mathrm{a}] = [\mathrm{a}^4]$. We compute

$$[\mathrm{c\,a\,c}](\vec{x}) = \begin{pmatrix} 0 & -\infty & 0 \\ 1 & 0 & 1 \\ 0 & -\infty & 0 \end{pmatrix} \vec{x} \quad [\mathrm{a\,c\,a}](\vec{x}) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 2 & 2 & 1 \end{pmatrix} \vec{x} \quad [\mathrm{a}^4](\vec{x}) = \begin{pmatrix} 0 & 0 & -\infty \\ 0 & 0 & -\infty \\ 1 & 1 & 0 \end{pmatrix} \vec{x}$$

therefore $[\mathrm{c\,a\,c}](\vec{x}) \geq_\lambda [\epsilon](\vec{x})$ and $[\mathrm{a\,c\,a}](\vec{x}) \gg_\lambda [\mathrm{a}^4](\vec{x})$. Note also that all the top left entries of matrices are finite. This allows us to remove the strict rule $\mathrm{a\,c\,a} \to \mathrm{a}^4$ using Theorem 6.7. The remaining strict rule can be removed by counting letters a. $\lhd$

7 Arctic Top Termination

As explained earlier, there are no strictly monotone, linear arctic functions of more than one argument. Therefore in this section we change our attention from full termination to top termination problems, where only weak monotonicity is required. This is not a very severe restriction as it fits with the widely used dependency pair method that replaces a full termination problem with an equivalent top termination problem, as remarked in Section 2.

The monotone algebra that we are going to use is the same as in Section 6, *i.e.*, $(\mathbb{N} \times \mathbb{A}_{\mathbb{N}}^{d-1}, \{f_A\}_{f\in\Sigma}, \gg, \geq)$. However now for proving top termination we will employ the second part of Theorem 3.5, so we only need a monotone algebra, instead of an extended monotone algebra. This allows us to consider arbitrary TRSs, as without the requirement of strict monotonicity we can allow arctic linear functions of more than one argument. The following theorem allows us to prove top termination in this setting:

Theorem 7.1. Let $\mathcal{R}, \mathcal{R}', \mathcal{S}$ be TRSs over a signature Σ and $[\cdot]$ be an arctic Σ-interpretation over $\mathbb{A}_{\mathbb{N}}$. If:

- for each $f \in \Sigma$, $[f]$ is somewhere finite,
- $[\ell] \geq_\lambda [r]$ for every rule $\ell \to r \in \mathcal{R} \cup \mathcal{S}$,
- $[\ell] \gg_\lambda [r]$ for every rule $\ell \to r \in \mathcal{R}'$ and
- $\mathrm{SN}(\mathcal{R}_{\mathrm{top}}/\mathcal{S})$.

Then $\mathrm{SN}(\mathcal{R}_{\mathrm{top}} \cup \mathcal{R}'_{\mathrm{top}}/\mathcal{S})$.

Proof. By Theorem 3.5b. By the same argument as in Theorem 6.7, $(\mathbb{N} \times \mathbb{A}_{\mathbb{N}}^{d-1}, [\cdot], \gg, \geq)$ is a weakly monotone algebra. So we only need to show that the evaluation stays within the algebra domain which follows from Lemma 6.2 and the first assumption. $\square$

We will illustrate this theorem on an example now.

Example 7.2. Consider the rewriting system `secret05/tpa2`:

$$\begin{array}{ll} (1)\ \mathrm{f}(\mathrm{s}(x), y) \to \mathrm{f}(\mathrm{p}(\mathrm{s}(x) - y), \mathrm{p}(y - \mathrm{s}(x))) & (3) \quad \mathrm{p}(\mathrm{s}(x)) \to x \\ (2)\ \mathrm{f}(x, \mathrm{s}(y)) \to \mathrm{f}(\mathrm{p}(x - \mathrm{s}(y)), \mathrm{p}(\mathrm{s}(y) - x)) & (4) \quad x - 0 \to x \\ & (5)\ \mathrm{s}(x) - \mathrm{s}(y) \to x - y \end{array}$$

It was solved in the 2007 competition by AProVE [16] using narrowing followed by polynomial interpretations and by $\mathsf{T_TT_2}$ [28] using polynomial interpretations with negative constants. In 2008 both provers used arctic interpretations to solve this problem.

After the dependency pair transformation, 9 dependency pairs can be removed using polynomial interpretations leaving the essential two dependency pairs:

$$\begin{array}{l} (1^\sharp)\ \mathrm{f}^\sharp(\mathrm{s}(x), y) \to \mathrm{f}^\sharp(\mathrm{p}(\mathrm{s}(x) - y), \mathrm{p}(y - \mathrm{s}(x))) \\ (2^\sharp)\ \mathrm{f}^\sharp(x, \mathrm{s}(y)) \to \mathrm{f}^\sharp(\mathrm{p}(x - \mathrm{s}(y)), \mathrm{p}(\mathrm{s}(y) - x)) \end{array}$$

So now, according to the dependency pair Theorem 2.2, we need to consider the relative top termination problem $\mathrm{SN}(\mathcal{R}_{\mathrm{top}}/\mathcal{S})$, where $\mathcal{R} = \{(1^\sharp), (2^\sharp)\}$ and $\mathcal{S} = \{(1), (2), (3), (4), (5)\}$. For that consider the following arctic interpretation

$$\begin{array}{ll} [\mathrm{f}^\sharp(\vec{x}, \vec{y})] = \begin{pmatrix} -\infty & -\infty \\ -\infty & -\infty \end{pmatrix} \vec{x} \oplus \begin{pmatrix} 0 & 0 \\ -\infty & -\infty \end{pmatrix} \vec{y} \oplus \begin{pmatrix} 0 \\ -\infty \end{pmatrix} & [0] = \begin{pmatrix} 3 \\ 3 \end{pmatrix} \\ [\vec{x} - \vec{y}] = \begin{pmatrix} 0 & -\infty \\ 0 & 0 \end{pmatrix} \vec{x} \oplus \begin{pmatrix} -\infty & -\infty \\ 0 & 0 \end{pmatrix} \vec{y} \oplus \begin{pmatrix} 0 \\ 0 \end{pmatrix} & [\mathrm{p}(\vec{x})] = \begin{pmatrix} 0 & -\infty \\ 0 & -\infty \end{pmatrix} \vec{x} \oplus \begin{pmatrix} -\infty \\ -\infty \end{pmatrix} \\ [\mathrm{f}(\vec{x}, \vec{y})] = \begin{pmatrix} 0 & 0 \\ 0 & -\infty \end{pmatrix} \vec{x} \oplus \begin{pmatrix} 2 & 0 \\ 0 & -\infty \end{pmatrix} \vec{y} \oplus \begin{pmatrix} 0 \\ -\infty \end{pmatrix} & [\mathrm{s}(\vec{x})] = \begin{pmatrix} 0 & 0 \\ 2 & 1 \end{pmatrix} \vec{x} \oplus \begin{pmatrix} 0 \\ 2 \end{pmatrix} \end{array}$$

which is somewhere finite and removes the second dependency pair:

$$[\mathrm{f}^\sharp(x, \mathrm{s}(y))] = \begin{pmatrix} -\infty & -\infty \\ -\infty & -\infty \end{pmatrix} \vec{x} \oplus \begin{pmatrix} 2 & 1 \\ -\infty & -\infty \end{pmatrix} \vec{y} \oplus \begin{pmatrix} 2 \\ -\infty \end{pmatrix}$$

$$[\mathrm{f}^\sharp(\mathrm{p}(x - \mathrm{s}(y)), \mathrm{p}(\mathrm{s}(y) - x))] = \begin{pmatrix} -\infty & -\infty \\ -\infty & -\infty \end{pmatrix} \vec{x} \oplus \begin{pmatrix} 0 & 0 \\ -\infty & -\infty \end{pmatrix} \vec{y} \oplus \begin{pmatrix} 0 \\ -\infty \end{pmatrix}$$

It is also weakly compatible with all the rules. The remaining dependency pair can be removed by a standard matrix interpretation of dimension two. $\triangleleft$

8 . . . Below Zero

In this section we will boldly go below zero: we extend the domain of matrix and vector coefficients from $\mathbb{A}_\mathbb{N}$ (arctic naturals) to $\mathbb{A}_\mathbb{Z}$ (arctic integers). This allows to interpret some function symbols by the "predecessor" function $x \mapsto x - 1$, and so represents their "intrinsic" semantics. This is the same motivation as the one for allowing polynomial interpretations with negative coefficients [19, 20].

We need to be careful though, as the relation $\gg$ on vectors of arctic integers is not well-founded. We will solve it in a similar way as in Sections 6 and 7, that is by restricting the first component of the vectors in our domain to natural numbers, which restores well-foundedness. So we are working in the $(\mathbb{N} \times \mathbb{A}_\mathbb{Z}^{d-1}, \{f_A\}_{f \in \Sigma}, \gg, \geq)$ algebra.

Again we need to make sure that we do not go outside of the domain, *i.e.*, the first vector component needs to be positive. This is ensured by the following property:

Definition 8.1. An n-ary arctic linear function

$$f(\vec{x}_1, \ldots, \vec{x}_n) = M_1 \otimes \vec{x}_1 \;\oplus\; \ldots \oplus M_n \otimes \vec{x}_n \;\oplus\; \vec{c}$$

over $\mathbb{A}_\mathbb{Z}$ is called *absolutely positive* if $\vec{c}\,[1]$ is positive. $\diamond$

Lemma 8.2. Let f be an n-ary arctic linear function over $\mathbb{A}_\mathbb{Z}$, $\vec{x}_1, \ldots, \vec{x}_n \in \mathbb{N} \times \mathbb{A}_\mathbb{Z}^{d-1}$ and $\vec{v} = f(\vec{x}_1, \ldots, \vec{x}_n)$. If f is absolutely positive then $\vec{v}[1] \in \mathbb{N}$.

Proof. Immediate, as $\vec{c}\,[1]$ positive by the definition of absolutely positive function.

$$\vec{v}[1] = f(\vec{x}_1, \ldots, \vec{x}_n)[1] = \max(\vec{c}\,[1], \ldots) \geq 0 \qquad \square$$

We can now present the main theorem of this section.

Theorem 8.3. Let $\mathcal{R}, \mathcal{R}', \mathcal{S}$ be TRSs over a signature Σ and $[\cdot]$ be an arctic Σ-interpretation over $\mathbb{A}_\mathbb{Z}$. If:

- for each $f \in \Sigma$, $[f]$ is absolutely positive,
- $[\ell] \geq_\lambda [r]$ for every rule $\ell \to r \in \mathcal{R} \cup \mathcal{S}$,

- $[\ell] \gg_\lambda [r]$ for every rule $\ell \to r \in \mathcal{R}'$ and
- $\mathrm{SN}(\mathcal{R}_{\mathrm{top}}/\mathcal{S})$.

Then $\mathrm{SN}(\mathcal{R}_{\mathrm{top}} \cup \mathcal{R}'_{\mathrm{top}}/\mathcal{S})$.

Proof. By Theorem 3.5b. We proved that $(\mathbb{N} \times \mathbb{A}_{\mathbb{N}}^{d-1}, \{f_A\}_{f\in\Sigma}, \gg, \geq)$ is a weakly monotone algebra in Theorem 7.1 — now the domain is extended from arctic naturals to arctic integers but all the properties carry over easily. The fact that we respect the algebra domain is ensured by the first property and Lemma 8.2. □

We now illustrate this theorem on an example.

Example 8.4. Let us consider the Beerendonk/2.trs TRS from the TPDB [2], consisting of the following six rules:

$$\begin{aligned} \mathrm{cond}(\mathrm{true}, x, y) &\to \mathrm{cond}(\mathrm{gr}(x, y), \mathrm{p}(x), \mathrm{s}(y)) & \mathrm{gr}(\mathrm{s}(x), \mathrm{s}(y)) &\to \mathrm{gr}(x, y) \\ \mathrm{gr}(0, x) &\to \mathrm{false} & \mathrm{gr}(\mathrm{s}(x), 0) &\to \mathrm{true} \\ \mathrm{p}(0) &\to 0 & \mathrm{p}(\mathrm{s}(x)) &\to x \end{aligned}$$

This is a straightforward encoding of the following imperative program

```
while x > y do (x, y) := (x-1, y+1);
```

with $x, y \in \mathbb{N}$ and the predecessor of x, *i.e.*, $x - 1$, defined on this domain, so $0 - 1 = 0$. This program is obviously terminating, however its encoding as the above TRS posed a serious challenge for the tools in the termination competition. We will now show a termination proof for this system using an arctic below zero interpretation.

We begin by applying the dependency pair method and obtaining four dependency pairs, three of which can be easily removed (for instance using standard matrix or polynomial interpretations) leaving the following single dependency pair:

$$\mathrm{cond}^\sharp(\mathrm{true}, x, y) \to \mathrm{cond}^\sharp(\mathrm{gr}(x, y), \mathrm{p}(x), \mathrm{s}(y))$$

Now, consider the following arctic matrix interpretation of dimension 1, so a degenerated case where arctic vectors and matrices simply become arctic numbers:

$$\begin{aligned} [\mathrm{cond}^\sharp(\vec{x}, \vec{y}, \vec{z})] &= (0)\vec{x} \oplus (0)\vec{y} \oplus (-\infty)\vec{z} \oplus (0) & [0] &= (0) \\ [\mathrm{cond}(\vec{x}, \vec{y}, \vec{z})] &= (0)\vec{x} \oplus (2)\vec{y} \oplus (-\infty)\vec{z} \oplus (0) & [\mathrm{false}] &= (0) \\ [\mathrm{gr}(\vec{x}, \vec{y})] &= (-1)\vec{x} \oplus (-\infty)\vec{y} \oplus (0) & [\mathrm{true}] &= (2) \\ [\mathrm{p}(\vec{x})] &= (-1)\vec{x} \oplus (0) & [\mathrm{s}(\vec{x})] &= (2)\vec{x} \oplus (3) \end{aligned}$$

This interpretation is absolutely positive, gives us a decrease for the dependency pair

$$\begin{aligned} [\mathrm{cond}^\sharp(\mathrm{true}, x, y)] &= (\;\;\,0)\vec{x} \oplus (-\infty)\vec{y} \oplus (2) \\ [\mathrm{cond}^\sharp(\mathrm{gr}(x, y), \mathrm{p}(x), \mathrm{s}(y))] &= (-1)\vec{x} \oplus (-\infty)\vec{y} \oplus (0) \end{aligned}$$

and all the original rules are oriented weakly. ◁

Remark 8.5. We discuss a variant which looks more liberal, but turns out to be equivalent to the one given here. We cannot allow $\mathbb{Z} \times \mathbb{A}_{\mathbb{Z}}^{d-1}$ for the domain, because it is not well-founded for $\gg$. So we can restrict the admissible range of negative values by some bound $c > -\infty$, and use the domain $\mathbb{A}_{\mathbb{Z}\geq c} \times \mathbb{A}_{\mathbb{Z}}^{d-1}$ where $\mathbb{A}_{\mathbb{Z}\geq c} := \{b \in \mathbb{A}_{\mathbb{Z}} \mid b \geq c\}$. Now to ensure that we stay within this domain we would demand that the first position of the constant vector of every interpretation is greater or equal than c.

Note however that this c can be fixed to 0 without any loss of generality as every interpretation using lower values in those positions can be "shifted" upwards. For any interpretation $[\cdot]$ and arctic number d construct an interpretation $[\cdot]'$ by $[t]' := [t] \otimes d$. This is obtained by going from $[f] = M_1\vec{x}_1 \oplus \ldots M_k\vec{x}_k \oplus \vec{c}$ to $[f]' = M_1\vec{x}_1 \oplus \ldots M_k\vec{x}_k \oplus \vec{c} \otimes d$. (A linear function with absolute part can be scaled by scaling the absolute part.) □

9 Quasi-Periodic Interpretations

Example 9.1. We consider the string rewriting system $S = \{bab \to a^3,\ a^3 \to b^3\}$, `Waldmann/jw1.srs` from TPDB, as a (running) example. Termination could not be established automatically by any of the programs taking part in the competition 2006. Then, Aleksey Nogin and Carl Witty produced a handwritten proof, that had been streamlined by Hans Zantema, and it had later been generalized into the method of *quasi-periodic interpretations* [39]. ◁

We recall the basic notion:

Definition 9.2. A function $f : \mathbb{N} \to \mathbb{N}$ is called *quasi-periodic* of *slope* s and *period* p if for all x, we have $f(x + p) = f(x) + sp$. ◇

In [39] it had been shown that quasi-periodic interpretations can prove termination of some rewrite systems for which no other proof was known (at the time). We now relate this approach to arctic matrix interpretations, by showing that they can simulate quasi-periodic interpretations of slope one for unary signatures.

Example 9.3. The dependency pairs transformation reduces the termination problem for S from Example 9.1 to the top termination problem $\mathrm{SN}(R_{\mathrm{top}}/S)$, with

$$R = \{Bab \to Aaa,\ Aaa \to Bbb\}$$

where all length-decreasing dependency pairs have already been removed. The proof given in [39] uses these quasi-periodic functions of period 3:

x	0	1	2	3	4	5	...
$[a](x) = [A](x)$	1	2	3	4	5	6	...
$[b](x) = [B](x)$	0	3	3	3	6	6	...

which induce these interpretations of the words in the rules:

x	0	1	2	3	4	5	...
$[Bab](x) = [bab](x)$	3	6	6	6	9	9	...
$[Aaa](x) = [aaa](x)$	3	4	5	6	7	8	...
$[Bbb](x) = [bbb](x)$	0	3	3	3	6	6	...

We infer that for all x, $[bab](x) \geq [aaa](x)$ and $[aaa](x) > [bbb](x)$, so there can not be infinitely many top applications of $Aaa \to Bbb$. This is the essential step in the termination proof. $\triangleleft$

We give an encoding from weakly monotonic quasi-periodic functions of slope one to arctic matrices and show that it is a morphism (it maps composition to multiplication) and that it respects weak and strong compatibility with a string rewriting system.

9.1 Basic translation

Throughout, we fix the natural number $p > 0$ to be the period.

Then each $x \in \mathbb{N}$ has a unique representation $x = qp + r$ with $0 \leq r < p$.

We define a mapping

$$\begin{aligned} \mathrm{av} &: \mathbb{N} \to \mathbb{A}^p \\ \mathrm{av} &: x \mapsto (-\infty, \dots, -\infty, \underbrace{q}_{\text{at position } r} -\infty, \dots, -\infty) \end{aligned}$$

In this section, vector indices start from 0 (not 1).

Example 9.4. For period $p = 3$, we have $\mathrm{av}(0) = (0, -\infty, -\infty)$ and $\mathrm{av}(4) = (-\infty, 1, -\infty)$. $\triangleleft$

For a quasi-periodic function f we define its *associated arctic matrix* $[f]$ (of size $p \times p$) by giving its column vectors:

$$[f] \quad = \quad \left(\mathrm{av}(f(0))^T \quad \dots \quad \mathrm{av}(f(p-1))^T \right)$$

Example 9.5. For period $p = 3$, consider the quasi-periodic functions

x	0	1	2	3	4	5	...
$f(x)$	1	2	4	4	5	7	...
$g(x)$	3	3	5	6	6	8	...

with associated matrices

$$[f] = \begin{pmatrix} -\infty & -\infty & -\infty \\ 0 & -\infty & 1 \\ -\infty & 0 & -\infty \end{pmatrix} \qquad [g] = \begin{pmatrix} 1 & 1 & -\infty \\ -\infty & -\infty & -\infty \\ -\infty & -\infty & 1 \end{pmatrix} \qquad \triangleleft$$

Lemma 9.6. If f is quasi-periodic of period p and slope one, then $[f] \otimes \mathrm{av}(x)^T = \mathrm{av}(f(x))^T$.

Proof. Let $x = pq + r$ with $0 \le r < p$. Since the slope of f is one, we have $f(x) = pq + f(r)$ and we put $f(r) = pq' + r'$ with $0 \le r' < p$.

We compute the entry at position i in $[f] \otimes \mathrm{av}(x)^T$. Since $\mathrm{av}(x)^T$ has exactly one finite entry, namely q at position r, we get q times the i-th position of the r-th column of $[f]$, which is $q \otimes \mathrm{av}(f(r))[i]$. This is finite exactly for $i = r'$, and then the value is $q \otimes q'$. So, the result vector is $\mathrm{av}(p(q + q') + r')$, and by the above, indeed $f(x) = pq + pq' + r'$. □

The mapping $[\cdot]$ is in fact a homomorphism:

Lemma 9.7. If f and g are both quasi-periodic functions of common period p and slope one, then $[f \circ g] = [g] \otimes [f]$.

Here, function composition is $(f \circ g) : x \mapsto g(f(x))$ and $\otimes$ is the (arctic) matrix product.

Proof. We compute row i of $[g] \otimes [f]$, which is $[g]$ times row i of $[f]$, being $[g] \otimes \mathrm{av}(f(i))^T$. By Lemma 9.6, this is $\mathrm{av}(g(f(i)))^T$. □

We remark that a matrix interpretation of this shape corresponds to a *complete and deterministic* weighted (word) automaton. This means that for each state and letter, there is exactly one transition with nonzero weight.

9.2 Weak Compatibility

Now we treat compatibility. Referring to Example 9.5, the function g is greater than the function f, but their associated matrices are not comparable w.r.t. $\gg$ or $\ge$. This will be repaired as follows. We start with weak compatibility.

For ease of presentation, we use arctic values below zero. We will see later that this can be removed.

We define an arctic triangular matrix of size $p \times p$ by

$$D \quad = \quad (\text{if } i \le j \text{ then } 0 \text{ else } -1)_{i,j}$$

Example 9.8. For $p = 3$ we get:

$$D = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -1 & -1 & 0 \end{pmatrix} \qquad \lhd$$

Lemma 9.9. For $x = pq + r$ with $0 \le r < p$, we have

$$D \otimes \mathrm{av}(x)^T = (\underbrace{q, \ldots, q}_{r+1 \text{ entries}}, \underbrace{q-1, \ldots, q-1}_{p-r-1 \text{ entries}})^T$$

Proof. Entry number i (counting starts at 0) of the result vector is: if $i \leq r$ then q else $q - 1$. $\square$

Lemma 9.10. If $x \leq y$, then $D \otimes \mathrm{av}(x)^T \leq D \otimes \mathrm{av}(y)^T$ component-wise.

Proof. Follows directly from Lemma 9.9. $\square$

The matrices that arise here have a special shape:

Definition 9.11. An arctic matrix is called *flat* if each row $(x_1, \ldots, x_n)$ fulfills $x_1 \leq \ldots \leq x_n \leq x_1 + 1$ and each column $(y_1, \ldots, y_m)^T$ fulfills $y_m + 1 \geq y_1 \geq \ldots \geq y_m$. $\diamond$

Note that we define this for any rectangular shape.

Lemma 9.12. If f is a weakly monotone quasi-periodic function of slope one, then $D \otimes [f]$ is flat.

Proof. By Lemma 9.9, each column of $D \otimes [f]$ has the required shape. For the shape of the rows, we argue as follows. The j-th and the $(j+1)$-th column of $D \otimes [f]$ are $D \otimes \mathrm{av}(f(j))^T$ and $D \otimes \mathrm{av}(f(j+1))^T$, respectively. By weak monotonicity of f, we have $f(j) \leq f(j+1)$, so by Lemma 9.10, each row of $D \otimes [f]$ is weakly increasing. Since f is monotonic, and also quasi-periodic of period p and slope one, we have $f(p-1) \leq f(p) = p + f(0)$. By Lemma 9.10,

$$D \otimes \mathrm{av}(f(p-1))^T \leq D \otimes \mathrm{av}(p + f(0))^T$$

Note that $\mathrm{av}(p + x) = \mathrm{av}(x) \otimes 1$, since arctic multiplication by 1 means just to increase each (finite) entry by one, therefore

$$D \otimes \mathrm{av}(f(p-1))^T \leq D \otimes \mathrm{av}(f(0))^T \otimes 1$$

so for each row index i, we have $(D \otimes [f])[i, p-1] \leq (D \otimes [f])[i, 0] + 1$. $\square$

Lemma 9.13. If M is flat, then $M \otimes D = M$.

Proof. The entry at position (i, j) in $M \otimes D$ is the dot product of row i in M and column j in D. Let row i of M be $\vec{x} = (x_1, \ldots, x_p)$, Column j in D has shape

$$(\underbrace{0, \ldots, 0}_{j \text{ times}}, \underbrace{-1, \ldots, -1}_{p-j \text{ times}})^T$$

The dot product of these vectors is

$$\max\{x_1, \ldots, x_j, x_{j+1} - 1, \ldots, x_p - 1\}$$

By flatness of M, we have $x_1 \leq \ldots x_p \leq x_1 + 1$, so the maximum is realized by x_j. This is exactly the value of the entry at position (i, j) in M. $\square$

Lemma 9.14. For any weakly monotonic quasi-periodic function f of slope one, $D \otimes [f] \otimes D = D \otimes [f]$.

Proof. By Lemma 9.12, $D \otimes [f]$ is flat. By Lemma 9.13, the claim follows. $\square$

Definition 9.15. $M_D := D \otimes M$ $\diamond$

Example 9.16. For f, g from Example 9.5,

$$[f]_D = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix} \qquad [g]_D = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad \lhd$$

Now we present two important properties of the translation $f \mapsto [f]_D$. It is a homomorphism (function composition corresponds to matrix multiplication), Lemma 9.17, and it respects the weak ordering, Lemma 9.18.

Lemma 9.17. $[f \circ g]_D = [g]_D \otimes [f]_D$

Proof. By Lemma 9.7, $[f \circ g]_D = ([g] \otimes [f])_D$. Denote $[f]$ by F and $[g]$ by G. Then $G_D F_D = DGDF = DGF = (GF)_D$ by Lemma 9.14. $\square$

Lemma 9.18. If quasi-periodic functions f, g of period p and slope one fulfill $\forall x : f(x) \geq g(x)$, then $[f]_D \geq [g]_D$.

Proof. We consider column r. It has value $D \otimes \mathrm{av}(f(r))^T$ resp. $D \otimes \mathrm{av}(g(r))^T$, with $f(r) \geq g(r)$ by assumption. The result then follows from Lemma 9.10. $\square$

The given translation $f \mapsto D \otimes [f]$ may create arctic matrices with negative entries. It can be verified that -1 is the *only* negative value that may ever appear, and that it is safe to replace it by $-\infty$, in order to obtain an interpretation with arctic naturals.

9.3 Strict Compatibility

Again referring to Example 9.5, the function g is strictly greater than the function f, but as Example 9.16 shows, $[g]_D \not\gg [f]_D$. By modifying the interpretation of some symbols, we obtain strict compatibility. For easier presentation, we use rational weights. Weights can be made integral by scaling: this is multiplication by a constant, resp. arctic exponentiation.

Define E as the $p \times p$ square matrix where all rows are equal to vector $F = (0, 1/p, \ldots, (p-1)/p)$. The interesting property of F is:

Lemma 9.19. $F \otimes \mathrm{av}(x)^T = x/p$.

Proof. Let $x = pq + r$. The vector $\mathrm{av}(x)$ has only one non-zero entry, namely q at position r. Then $F \otimes \mathrm{av}(x)^T = r/p + q = x/p$. $\square$

This implies

Lemma 9.20. If $x, y \in \mathbb{N}$ and $x > y$, then $F \otimes \mathrm{av}(x)^T > F \otimes \mathrm{av}(y)^T$ and $E \otimes \mathrm{av}(x)^T \gg E \otimes \mathrm{av}(y)^T$.

Proof. The first statement follows from the previous lemma. Then the second statement follows, as all rows of E are equal to F. □

Now the application is that we can multiply an interpretation (that was translated according to $f \mapsto [f]_D$) from the left by E, to get the desired relation:

Lemma 9.21. If quasi-periodic functions f, g of period p and slope one fulfill $f < g$ point-wise, then $E \otimes [f]_D \ll E \otimes [g]_D$.

Proof. We have $E \otimes [f]_D = ED[f] = E[f]$, since E is flat and Lemma 9.13 applies. Column r of $E[f]$ is the product of E and column r of $[f]$, thus $E \otimes \mathrm{av}(f(r))^T$. This is to be compared with $E \otimes \mathrm{av}(g(r))^T$, so we apply Lemma 9.20. □

9.4 Putting it all together

While we achieve weak compatibility (w.r.t. $\geq$) by the translation $[\cdot]_D$, we get strict compatibility (w.r.t. $\gg$) only for the shape of top rewrite relations that arise from the dependency pair transformation.

Theorem 9.22. Given a weakly monotonic quasi-periodic interpretation of period p and slope one that is weakly compatible with S and R and strictly compatible with R', where the top symbols of $R \cup R'$ do not occur in S, there is an arctic matrix interpretation of dimension p that fulfills the conditions of Theorem 7.1: it is weakly compatible with S and R, and strictly compatible with R'.

Proof. This interpretation is obtained by taking as the translation of a non-top symbol interpretation f the matrix $[f]_D$, and for a top symbol, the matrix $E \otimes [f]_D$. This interpretation is somewhere finite since the top left entry of each matrix is finite. This follows from $f(0) \geq 0$. This translation computes the correct values by Lemma 9.17, and we get weak compatibility by Lemma 9.18 as well as strict compatibility by Lemma 9.21. □

Example 9.23. For the quasi-periodic interpretation from Example 9.3,

$$[a] = \begin{pmatrix} -\infty & -\infty & 1 \\ 0 & -\infty & -\infty \\ -\infty & 0 & -\infty \end{pmatrix} \qquad [a]_D = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

$$[b] = \begin{pmatrix} 0 & 1 & 1 \\ -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty \end{pmatrix} \qquad [b]_D = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

$$[a^3]_D = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \qquad [b^3]_D = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

$$E \otimes [a^3]_D = \begin{pmatrix} 1 & 4/3 & 5/3 \\ 1 & 4/3 & 5/3 \\ 1 & 4/3 & 5/3 \end{pmatrix} \qquad E \otimes [b^3]_D = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$$E \otimes [a]_D = \begin{pmatrix} 1/3 & 2/3 & 1 \\ 1/3 & 2/3 & 1 \\ 1/3 & 2/3 & 1 \end{pmatrix} \qquad E \otimes [b]_D = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$\lhd$

Remark 9.24. We comment on the effect of restricting the slope to one. Quasi-periodic interpretations of higher slope cannot be represented by arctic matrix interpretations: for instance, the function $f : x \mapsto 2x$ is a quasi-periodic function of slope 2 and period 1 (trivially), and iterated application of f gives exponentially increasing values, while arctic matrices (of any dimension) can only give linear growth. On the other hand, as has been remarked already in [39], if quasi-periodic functions are applied together with other termination methods, restricting the slope to one does not seem to reduce the power of the method too much. There are several hard termination problems where slope one is sufficient. $\square$

10 Certification

The theory developed in this paper for proving termination with arctic (below-zero) interpretations (*i.e.*, Theorem 7.1 and Theorem 8.3) is accompanied by formal proofs in the Coq proof assistant [34, 6]. Coq is a proof assistant/checker based on the Calculus of Inductive Constructions (CIC) [33] — a very expressive logic supporting simple, inductive, dependent and polymorphic types.

The certification has been carried out within the CoLoR project [7]. The main part of CoLoR is a library of termination techniques formalized in Coq. The aim of the project is to gather such formalizations within this library and then use them to certify concrete proofs produced by some existing termination provers.

This is accomplished by introducing an intermediate format for termination proofs. Automated termination provers only need to support this format (which should be easy to accomplish) and then Rainbow, a simple tool developed within the CoLoR project, transforms proofs in this format to actual Coq scripts certifying termination. Then Coq is run as a proof checker to verify correctness of the proof reported by the termination tool, with the use of the termination criteria formalized in the CoLoR library.

We formalized the arctic techniques for proving termination of term rewriting presented in this paper, so the criteria corresponding to Theorems 7.1 and 8.3 but not that of Theorem 6.7.

Let us illustrate the CoLoR approach to certification on the termination criteria developed in this paper. An arctic termination proof requires providing an arctic interpretation, Definition 4.4, which consists of:

- a natural number d, indicating the dimension of vectors and matrices that are used,

- a theorem to be used: either Theorem 7.1 or Theorem 8.3 and
- for every symbol f the arctic function corresponding to it (over $\mathbb{A}_{\mathbb{N}}$ in case of Theorem 7.1 or over $\mathbb{A}_{\mathbb{Z}}$ for Theorem 8.3), *i.e.*, the constant vector $\vec{c}$ and its linear factors $M_1, \ldots, M_n$, where n is the arity of f.

Such arctic interpretation can be specified by means of a formal grammar, which is part of the aforementioned proof format. Then proofs like those presented in Example 7.2 or Example 8.4 can be easily expressed in this format and translated by Rainbow to formal Coq proofs, allowing to check their correctness with Coq. Similar transformation can be applied to termination proofs generated by termination provers, allowing a fully automatic certification of their results. We implemented this in the termination prover Matchbox [36] and we report on the results in the following section.

The basis of this formalization work was the certification of the matrix interpretations method [27], the method introduced shortly in Example 3.6, which consists of formalizations of:

- a semiring structure,
- vectors and matrices over arbitrary semirings of coefficients,
- the monotone algebras framework and
- the matrix interpretation method.

The framework of monotone algebras was used without any changes at all. Vectors and matrices were formalized in [27] for arbitrary semirings, however all the results involving orders were developed for the usual orders on natural numbers, as used in the matrix interpretation method. So the first step in the formalization process was to generalize the semiring structure to a semiring equipped with two orders $(>, \geq)$ and to adequately generalize results on vectors and matrices. Then the arctic semiring was developed in this setting.

As for the technique itself it has a lot in common with the technique of matrix interpretations. Therefore the common parts were extracted to a module `MatrixBasedInt` which was then specialized to the matrix interpretation method (`MatrixInt`) and to a basis for arctic based methods (`ArcticBasedInt`), which was narrowed down to the methods of arctic interpretations (`ArcticInt`) and arctic below-zero interpretations (`ArcticBZInt`). This hierarchy is depicted in Figure 1.

We did not yet formalize the developments of Section 9 on quasi-periodic interpretations. Note that this is not strictly necessary: one can still certify quasi-periodic termination proofs with CoLoR indirectly: just carry out the transformation according to Theorem 9.22, and submit the corresponding arctic interpretation for certification. This has been realized by the termination prover Matchbox in the Certified Termination Competition for string rewriting in 2008.

Considering the extension of the formal proof format and the Rainbow tool it was minimal. The format for the matrix interpretation proofs was already developed in [27] and it essentially requires to provide matrix interpretations for all the

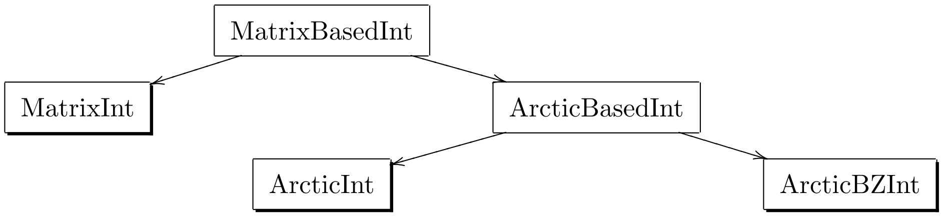

Figure 1: Hierarchy of different matrix-based methods in CoLoR.

function symbols in the signature. The format for arctic interpretations is the same except that:

- it indicates which matrix-based method is to be used, indicated by different XML tags (as the common proof format of CoLoR is in the end specified using XML syntax),
- the entries of vectors and matrices are from a different domain.

Below we illustrate the performance of the winning tools in the certified competitions of 2007 and 2008. In case of the 2008 competition we narrow down the results only to the set of problems used in 2007, to make the figures and tools' performances comparable.

category	year	winner	termination proofs
TRS	2007	CoLoR + TPA	$354/975 \approx 36.3\%$
	2008	CoLoR + AProVE	$420/975 \approx 43.1\%$
SRS	2007	Matchbox (uncertified)	$337/517 \approx 65.2\%$
	2008	CoLoR + Matchbox	$354/517 \approx 68.5\%$

Figure 2: Performance of the winners of the certified competition in 2007 and 2008.

In the SRS category the only improvement in Matchbox [36] from 2007 to 2008 was the addition of arctic interpretations, and the transformation from quasi-periodic to arctic interpretations.

Even more importantly we are comparing here the un-certified Matchbox 2007 (there was no certified SRS category in 2007) with its certified 2008 counterpart. That means that some features not supported by CoLoR had to be switched off in Matchbox for the 2008 competition and still it performed better than in the previous year. Roughly half of the proofs produced by Matchbox involved arctic interpretations, showing the importance of this technique for string rewriting.

The interpretation of the results in the TRS category is much more difficult. TPA [24] and AProVE [16] are very different tools using different proof search strategies. Also apart from the addition of arctic interpretations, AProVE could take advantage of the dependency graph decomposition technique that was not available in CoLoR in the previous year.

In the following section we will present the experimental data comparing performance of Matchbox with different sets of techniques in use, which will allow for a better evaluation of the impact of arctic interpretations on termination proving power.

11 Implementation

The implementation in Matchbox follows the scheme described in [13]. The constraint problem for the arctic interpretation is translated to a constraint problem for matrices, for arctic numbers and, finally, for Boolean variables. This is then solved by Minisat [11].

An arctic number is represented by a pair $a = (b; v_0, v_1, \ldots, v_n)$ where b is a Boolean value and $v_0, \ldots, v_n$ is a sequence of Booleans (all numbers have fixed bit-width). If b is 1, then a represents $-\infty$, if b is 0, then a represents the binary value of $v_0, \ldots, v_n$.

To represent integers, we use two's complement representation, *i.e.*, the most significant bit is the "sign bit".

Note that implementation of max/plus operation is less expensive than standard plus/times: with a binary representation both max and plus can be computed (encoded) with a linear size formula (whereas a naive implementation of the standard multiplication requires quadratic size and asymptotically better schemes do not pay off for small bit widths).

It is useful to require the following, for each arctic number $a = (b, v)$: if the infinity bit b is set, then $v = 0$. Then $(b, v) \oplus (b', v') = (b \wedge b', \max(v, v'))$. For $(b, v) \otimes (b', v')$ we compute $c = b \vee b'$, $u = (u_0, \ldots, u_n) = v + v'$ and the result is $(c; \neg c \wedge u_0, \ldots, \neg c \wedge u_n)$.

To represent arctic integers, we use a similar convention: if the infinity flag b is set, we require that the number v represents the lowest value of its range.

The following table lists the numbers of certified proofs that we obtain with the dependency pair method (without analyzing the dependency graph) and the following matrix methods: (s)tandard, (a)rctic, below (z)ero. The problem set used for this experiment is the TPDB of the 2007 competition, so the problems correspond to those presented in Figure 2.

Runs were executed on a single core of an Intel X5365 processor running at 3GHz. All proofs are available for inspection at the Matchbox web page [36]. In all cases we used standard matrices of dimension 1 and 2 to remove rules before the dependency pair transformation, and then matrix dimensions d from 1 up; with numbers of bit width 3, and a timeout of $1+d^2$ seconds for each individual attempt.

Here, we count only verified proofs, so we are missing about 3 to 5 proofs where Coq does not finish in reasonable time.

For a SRS $\mathcal{R}$ we consider $\mathrm{reverse}(\mathcal{R}) = \{\mathrm{reverse}(l) \to \mathrm{reverse}(r) \mid (l \to r) \in \mathcal{R}\}$. Clearly this transformation preserves termination both ways (the implication $\mathrm{SN}(\mathrm{reverse}(\mathcal{R})) \implies \mathrm{SN}(\mathcal{R})$ has been formalized in CoLoR). Half of the allotted

problem set	time	number of proofs found by the method			
		s	sa	sz	saz
975 TRS	1 min	361	376	388	389
	10 min	365	381	393	394
517 SRS	1 min	178	312	298	320
	10 min	185	349	323	354

Figure 3: Performance of Matchbox using different versions of matrix (arctic) interpretations.

time is spent for each of $\mathcal{R}$ and reverse($\mathcal{R}$). This increases the score considerably (by about one third).

In the previous section we already mentioned that the method of arctic interpretations was implemented in AProVE. Recently it was also incorporated into $\mathsf{T_TT_2}$ and both those provers used it in the termination competition of 2008, where they took the first two places in the main categories (full termination of term/string rewriting systems).

12 Discussion

Arctic naturals form a sub-semiring of arctic integers. So the question comes up whether Theorem 8.3 subsumes Theorem 7.1. Note that the prerequisites for both theorems are incomparable. Still there might be a method to construct from a somewhere-finite interpretation (above zero) an equivalent absolutely positive interpretation (below zero). We are not aware of any. Experience with implementation shows that it is useful to have both methods, especially for string rewriting. Naturals are easier to handle than integers because they do not require signed arithmetics. So typically we can increase the bit width or the matrix dimension for naturals. Our implementation finds several proofs according to Theorem 7.1 where it fails to find a proof according to Theorem 8.3 and vice-versa.

It is interesting to ask whether the preconditions of Theorems 6.7,7.1,8.3 can be weakened. We discussed one variant in Remark 8.5. In general, a linear interpretation $[\cdot]$ with coefficients in $\mathbb{A}_\mathbb{N}$ ($\mathbb{A}_\mathbb{Z}$ respectively) is admissible for a termination proof if for each ground term t, the value $[t]$ is finite (positive, respectively). This is in fact a reachability problem for weighted (tree) automata. It is decidable for interpretations on arctic naturals, but it is undecidable for arctic integers (this follows from a result of Krob [29] on tropical word automata). In our setting, we do not guess an interpretation and then decide whether it is admissible. Rather, we have to formulate the decision algorithm as part of the constraint system for the interpretation. Therefore we chose sharper conditions on interpretations that imply finiteness (positiveness, respectively) and have an easy constraint encoding.

Another question is the relation of the standard matrix method to the arctic matrix method(s). Performance of our implementation suggests that neither method subsumes the other, but this may well be a problem of computing resources, as we hardly reach matrix dimension 5 and bit width 3.

As for the relation to other termination methods (*e.g.*, path orderings), the only information we have is that arctic (and other) matrix methods can do non-simple termination, while path orders and polynomial interpretations cannot; and on the other hand, the arctic matrix method implies a linear bound on derivational complexity (see below), which is easily surpassed by path orders and other interpretations.

The full arctic termination method bounds lengths of derivations:

Lemma 12.1. For a rewriting system $\mathcal{R}$ that fulfills the requirements of Theorem 6.7 for $\mathcal{S} = \emptyset$, the derivational complexity of $\mathcal{R}$ is linear.

Proof. For a finite arctic vector $\vec{x} = (x_1, \ldots, x_k)$, define $|\vec{x}| = \max(x_1, \ldots, x_k)$.

Then $|\vec{x} \oplus \vec{y}| \leq \max(|\vec{x}|, |\vec{y}|)$ and $|\vec{x} \otimes \vec{y}^T| \leq |\vec{x}| + |\vec{y}|$.

For a finite arctic matrix A of dimension $k \times k$, define $|A| = \max\{A[i,j] \mid 1 \leq i,j \leq k\}$. Then $|A \otimes \vec{x}| \leq |A| + |\vec{x}|$ and $|A \otimes B| \leq |A| + |B|$.

For an interpretation $[\cdot]$ of some signature Σ, and any word $w \in \Sigma^*$, this implies that $|[w]| \leq c \cdot |w|$ where $c = \max\{|[f]| : f \in \Sigma\}$.

Now we remark that $u \to_{\mathcal{R}} v$ implies $[u] \gg [v]$, and $\vec{x} \gg \vec{y}$ implies $|\vec{x}| > |\vec{y}|$. Thus the derivational complexity of $\mathcal{R}$ is linear: any derivation starting from u has at most $c \cdot |u|$ steps. $\square$

This means that rewriting systems with higher derivational complexity (*e.g.*, quadratic: $\{\mathrm{a\,b} \to \mathrm{b\,a}\}$, or exponential $\{\mathrm{a\,b} \to \mathrm{b}^2\,\mathrm{a}\}$) do not admit an arctic termination proof. Note that both these systems admit a standard matrix proof.

It seems very difficult to combine this argument with the dependency pair method, as it can drastically alter (*i.e.*, reduce) derivational complexity [32].

Example 12.2. The following rewriting system [21] has a derivational complexity that is not primitive recursive:

$$\{\mathrm{s}(x) + (y + z) \to x + (\mathrm{s}(\mathrm{s}(y)) + z), \mathrm{s}(x) + (y + (z + w)) \to x + (z + (y + w))\}$$

and still it has, after dependency pairs transformation, an easy termination proof by "counting symbols" [13]. Note however that arctic interpretations cannot count globally: to compute the interpretation $[f(t_1, t_2)]$, it is impossible to add values from subtrees $[t_1], [t_2]$, as we can only take the maximum of $[t_1], [t_2]$. Yet we find an arctic proof, as follows. The given system is in fact an encoding of a length-preserving string rewriting system on the infinite alphabet $\mathbb{N}$. Both rules keep the right spine of terms (corresponding to the length of the simulated string) intact, so we can remove dependency pairs that shrink it, using the interpretation $[+](x, y) = y \otimes 1$. We are left with two dependency pairs (that directly correspond to the original rules). They can be handled by $[+](x, y) = x$ and $[\mathrm{s}](x) = x \otimes 1$. So instead of numbers of symbols, we were just using path lengths. $\lhd$

Max/plus polynomials have been used by Amadio [3] as quasi-interpretations (*i.e.*, functions are weakly monotone), to bound the space complexity of derivations. Proving termination directly was not intended.

13 Conclusions

We presented the arctic interpretations method for proving termination of term rewriting. It is based on the matrix interpretation method [13] where the usual plus/times operations on $\mathbb{N}$ are generalized to an arbitrary semiring, in this case instantiated by the arctic semiring (max/plus algebra) on $\{-\infty\} \cup \mathbb{N}$.

Matrix interpretations are an efficient realization of a certain class of weighted tree automata. It remains a subject of further study to characterize the family of weighted tree languages that can be represented in that way.

We also generalized this to arctic integers. This generalization allowed us to solve 10 of Beerendonk/* examples that are difficult to prove terminating and thus far could only be solved by AProVE with the Bounded Increase [17] technique, dedicated to such class of problems coming from transformations from imperative programs and with polynomial interpretations with rational coefficients [30].

Our presentation of the theory is accompanied by a formalization in the Coq proof assistant. By becoming part of the CoLoR project this formalization allows us to formally verify termination proofs involving the arctic matrix method. It was evaluated in the certified category of the termination competition in 2008 and turned out to be a crucial contribution allowing CoLoR to win with the competing certification back-end, A3PAT [8].

We want to remark here that all performance data and all examples presented in this paper were collected from problems of TPDB and we did not "cook up" any special examples to show off the arctic method. The emphasis of these examples (in fact, of the whole paper) is not to provide termination proofs where none were known before, but rather to provide certified (and often conceptually simpler) termination proofs where only uncertified proofs were available up to now.

References

[1] The termination competition. `http://www.termination-portal.org/wiki/Termination_Competition`.

[2] Termination problems data base. `http://dev.aspsimon.org/projects/termcomp/downloads/`.

[3] Amadio, R. M. Synthesis of max-plus quasi-interpretations. *Fundamenta Informaticae*, 65(1-2):29–60, 2005.

[4] Arts, T. and Giesl, J. Termination of term rewriting using dependency pairs. *Theoretical Computer Science*, 236(1-2):133–178, 2000.

[5] Baader, F. and Nipkow, T. *Term Rewriting and All That.* Cambridge University Press, 1998.

[6] Bertot, Y. and Castéran, P. *Coq'Art: The Calculus of Inductive Constructions.* EATCS Texts in Theoretical Computer Science. Springer, 2004.

[7] Blanqui, F., Delobel, W., Coupet-Grimal, S., Hinderer, S., and Koprowski, A. CoLoR, a Coq library on rewriting and termination. In *Workshop on Termination (WST)*, 2006. `http://color.loria.fr`.

[8] Contejean, É., Courtieu, P., Forest, J., Pons, O., and Urbain, X. Certification of automated termination proofs. In *International Symposium on Frontiers of Combining Systems (FroCoS)*, volume 4720 of *Lecture Notes in Computer Science*, pages 148–162, 2007.

[9] Droste, M., Pech, C., and Vogler, H. A Kleene theorem for weighted tree automata. *Theory of Computing Systems*, 38(1):1–38, 2005.

[10] Droste, M. and Vogler, H. Weighted tree automata and weighted logics. *Theoretical Computer Science*, 366(3):228–247, 2006.

[11] Eén, N. and Sörensson, N. An extensible SAT-solver. In *Theory and Applications of Satisfiability Testing (SAT)*, volume 2919 of *Lecture Notes in Computer Science*, pages 502–518, 2003.

[12] Endrullis, J. Jambox. `http://joerg.endrullis.de`.

[13] Endrullis, J., Waldmann, J., and Zantema, H. Matrix interpretations for proving termination of term rewriting. *Journal of Automated Reasoning*, 40(2–3):195–220, 2008.

[14] Fuchs, L. *Partially Ordered Algebraic Systems.* Addison-Wesley, 1962.

[15] Gaubert, S. and Plus, M. Methods and applications of (max, +) linear algebra. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1200 of *Lecture Notes in Computer Science*, pages 261–282, 1997.

[16] Giesl, J., Thiemann, R., Schneider-Kamp, P., and Falke, S. Automated termination proofs with AProVE. In *International Conference on Rewriting Techniques and Applications (RTA)*, volume 3091 of *Lecture Notes in Computer Science*, pages 210–220, 2004.

[17] Giesl, J., Thiemann, R., Swiderski, S., and Schneider-Kamp, P. Proving termination by bounded increase. In *International Conference on Automated Deduction (CADE)*, volume 4603 of *Lecture Notes in Computer Science*, pages 443–459, 2007.

[18] Golan, J. S. *Semirings and their Applications.* Kluwer, 1999.

[19] Hirokawa, N. and Middeldorp, A. Polynomial interpretations with negative coefficients. In *International Conference on Artificial Intelligence and Symbolic Computation (AISC)*, volume 3249 of *Lecture Notes in Computer Science*, pages 185–198, 2004.

[20] Hirokawa, N. and Middeldorp, A. Tyrolean termination tool: Techniques and features. *Information and Computation*, 205(4):474–511, 2007.

[21] Hofbauer, D. and Lautemann, C. Termination proofs and the length of derivations. In *International Conference on Rewriting Techniques and Applications (RTA)*, volume 355 of *Lecture Notes in Computer Science*, pages 167–177, 1989.

[22] Hofbauer, D. and Waldmann, J. Termination of string rewriting with matrix interpretations. In *International Conference on Rewriting Techniques and Applications (RTA)*, volume 4098 of *Lecture Notes in Computer Science*, pages 328–342, 2006.

[23] Hong, H. and Jakuš, D. Testing positiveness of polynomials. *Journal of Automated Reasoning*, 21(1):23–38, 1998.

[24] Koprowski, A. TPA: Termination proved automatically. In *International Conference on Rewriting Techniques and Applications (RTA)*, volume 4098 of *Lecture Notes in Computer Science*, pages 257–266, 2006. `http://www.win.tue.nl/tpa`.

[25] Koprowski, A. and Waldmann, J. Arctic termination ... below zero. In *International Conference on Rewriting Techniques and Applications (RTA)*, volume 5117 of *Lecture Notes in Computer Science*, pages 202–216, 2008.

[26] Koprowski, A. and Waldmann, J. Max/plus tree automata for termination of term rewriting. In *Workshop on Weighted Automata Theory and Applications (WATA)*, 2008.

[27] Koprowski, A. and Zantema, H. Certification of proving termination of term rewriting by matrix interpretations. In *Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 4910 of *Lecture Notes in Computer Science*, pages 328–339, 2008.

[28] Korp, M., Sternagel, C., Zankl, H., and Middeldorp, A. Tyrolean Termination Tool 2. In *International Conference on Rewriting Techniques and Applications (RTA)*, volume 5595 of *Lecture Notes in Computer Science*, pages 295–304, 2009. See `http://colo6-c703.uibk.ac.at/ttt2`.

[29] Krob, D. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. In *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 623 of *Lecture Notes in Computer Science*, pages 101–112, 1992.

[30] Lucas, S. Polynomials over the reals in proofs of termination: from theory to practice. *RAIRO: Theoretical Informatics and Applications*, 39(3):547–586, 2005.

[31] Marché, C., Waldmann, J., and Zantema, H. The termination competition 2007. `http://www.imn.htwk-leipzig.de/~waldmann/talk/07/wst/competition`.

[32] Moser, G. and Schnabl, A. The derivational complexity induced by the dependency pair method. In *International Conference on Rewriting Techniques and Applications (RTA)*, volume 5595 of *Lecture Notes in Computer Science*, pages 255–269, 2009.

[33] Paulin-Mohring, C. Inductive definitions in the system Coq - rules and properties. In *International Conference on Typed Lambda Calculi and Applications (TLCA)*, volume 664 of *Lecture Notes in Computer Science*, pages 328–345, 1993.

[34] The Coq Development Team. The Coq proof assistant: Reference manual, version 8.2. `http://coq.inria.fr`, 2009.

[35] Thiemann, R., Zantema, H., Giesl, J., and Schneider-Kamp, P. Adding constants to string rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 19(1):27–38, 2008.

[36] Waldmann, J. Matchbox: A tool for match-bounded string rewriting. In *International Conference on Rewriting Techniques and Applications (RTA)*, volume 3091 of *Lecture Notes in Computer Science*, pages 85–94, 2004. `http://dfa.imn.htwk-leipzig.de/matchbox`.

[37] Waldmann, J. Arctic termination. In *Workshop on Termination (WST)*, 2007.

[38] Waldmann, J. Weighted automata for proving termination of string rewriting. *Journal of Automata, Languages and Combinatorics*, 12(4):545–570, 2007.

[39] Zantema, H. and Waldmann, J. Termination by quasi-periodic interpretations. In *International Conference on Rewriting Techniques and Applications (RTA)*, volume 4533 of *Lecture Notes in Computer Science*, pages 404–418, 2007.

Received 1st August 2008