Acta Cybernetica **19** (2009) 295–312.

# Weighted Automata Define a Hierarchy of Terminating String Rewriting Systems

Andreas Gebhardt* and Johannes Waldmann†

**Abstract**

The "matrix method" (Hofbauer and Waldmann 2006) proves termination of string rewriting via linear monotone interpretation into the domain of vectors over suitable semirings. Equivalently, such an interpretation is given by a weighted finite automaton. This is a general method that has as parameters the choice of the semiring and the dimension of the matrices (equivalently, the number of states of the automaton). We consider the semirings of nonnegative integers, rationals, algebraic numbers, and reals; with the standard operations and ordering. Monotone interpretations also allow to prove relative termination, which can be used for termination proofs that consist of several steps. The number of steps gives another hierarchy parameter. We formally define the hierarchy and we prove that it is infinite in both directions (dimension and steps).

**Keywords:** string rewriting, relative termination, weighted automaton, matrix interpretation, monotone algebra.

## 1 Introduction

Rewriting is pattern replacement in context. It serves as a model of computation that is Turing-complete. Thus all "interesting" semantic properties are undecidable, including the very natural question of *termination* [18]: for a given rewriting system, are all derivations finite? Since the problem is significant in practice, e.g. for the analysis of software, one is interested in semi-algorithms: computable methods of proving termination that are sound, but not complete.

One method to prove termination of rewriting is "matrix interpretation" [13]. These interpretations are in fact $\mathbb{N}$-weighted finite automata. Several automated termination provers now implement this method, and indeed the outcome of recent Termination Competitions is heavily influenced by "matrix proofs".

*Martin-Luther-Universität Halle-Wittenberg, Von-Seckendorff-Platz 1, D-06120 Halle , Germany. E-mail: `andreas.gebhardt@informatik.uni-halle.de`

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

Related to that, investigations of matrix method(s) mainly focused on proving correctness, and then efficiency of implementation in solving the corresponding constraint systems for the matrix entries.

With the present paper, we intend to start a systematic study of matrix method(s) as *proof systems*. We define a suitable hierarchy of termination problems and explore its properties.

One parameter of this hierarchy is the size of the matrices used in the proof, corresponding to the number of states of the automata.

Another parameter is the underlying (semi)ring. In the present paper, we consider weight rings that include $\mathbb{N}$. In [8] we reported on some experiments with non-negative rationals.

The matrix interpretation method in fact solves a more general problem: that of *relative termination*. A rewriting system $R$ terminates relative to a rewriting system $S$ if each mixed derivation (containing $R$ and $S$ steps in any order) contains only finitely many $R$ steps. While being an interesting concept in itself [10], relative termination helps to solve standard termination problems because it allows to compose termination proofs: if $R$ terminates relative to $S$ then termination of $R \cup S$ follows from termination of $S$, and the latter can be proved separately. That way, termination of a rewriting system can be shown incrementally, and the number of proof steps gives another interesting parameter for the hierarchy.

In the present paper, we focus on string rewriting. The matrix method has been generalized to term rewriting [6], but we leave the investigation of the corresponding hierarchy of terminating term rewriting systems for further study.

After giving preliminaries on string rewriting in Section 2 and on termination proofs via weighted word automata in Section 3, we define the corresponding hierarchy of (relatively) terminating rewriting systems in Section 4. Then we discuss the hierarchy with respect to matrix dimension in Section 5 (with a particular case in Section 6), choice of the weight semiring in Section 7, and number of proof steps in Section 8.

We obtain these results:

- the hierarchy is infinite with respect to matrix dimension (Theorem 2)
- rational weights are strictly more powerful than integral weights (Theorem 5)
- the hierarchy is infinite with respect to the number of proof steps (Theorem 6).

Some of the results in this paper have been announced in contributions to the Workshop on Termination [8] and to the Workshop on Weighted Automata [9].

## 2 Notation and Preliminaries

**Strings and Rewriting.** Given a finite alphabet $\Sigma$, denote by $\Sigma^*$ the set of finite words with letters from $\Sigma$. In fact $\Sigma^*$ is a monoid under the operation $\cdot$ of concatenation, with the empty word $\epsilon$ as unit.

A *string rewriting system* [3] is a set $R$ of rules, where a rule is a pair of words. We often write the rule $(l, r)$ as $(l \to r)$. A string rewriting system $R$ defines a (one-step) rewrite relation over $\Sigma^*$ by $u \to_R v$ if there exists $(l, r) \in R$ and $x, y \in \Sigma^*$ such that $u = x \cdot l \cdot y$ and $v = x \cdot r \cdot y$. For example, for $R = \{ab \to baa\}$ over $\Sigma = \{a, b\}$, we have $abb \to_R baab \to_R babaa \to_R bbaaaa$. We often write $R$ (the system) as a shorthand for $\to_R$ (the relation).

**Relations and Termination.** For a relation $\to$, we write $\mathrm{SN}(\to)$ if this $\to$ is well-founded, that is, if there is no infinite chain $x_0 \to x_1 \to \ldots$ We also say that $\to$ is *terminating*.

We denote the composition of relations $\to_1$ and $\to_2$ by $\to_1 \circ \to_2$, the transitive closure of a relation $\to$ by $\to^+$, and the transitive and reflexive closure by $\to^*$.

For relations $\to_1, \to_2$, define $\to_1 / \to_2$ as $\to_1 \circ \to_2^*$. Then $\mathrm{SN}(\to_1 / \to_2)$ denotes that $\to_1$ is *terminating relative to* $\to_2$: there is no $(\to_1 \cup \to_2)$-chain containing infinitely many $\to_1$ steps. Note that $\to_1 / \emptyset = \to_1$.

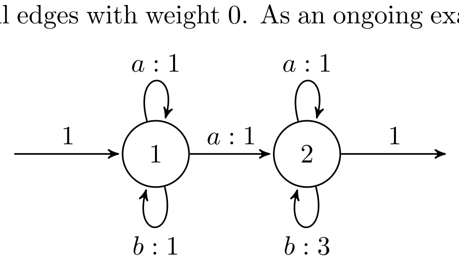
By the above remark, we write $\mathrm{SN}(R)$ ("the system $R$ is terminating") for $\mathrm{SN}(\to_R)$ ("the derivation relation of $R$ is terminating").

**Semirings.** A *semiring* [11] has a carrier $D$ with operations $+$ (addition) and $\cdot$ (multiplication) and designated elements 0 (zero) and 1 (unit), such that $(D, +, 0)$ is a commutative monoid, and $(D, \cdot, 1)$ is a monoid, addition distributes over multiplication from both sides, and $0 \cdot a = 0 = a \cdot 0$. A semiring is *partially ordered* [7] if there is a relation $\geq$ on $D$ that is compatible with the operations. In the present paper, we use semirings over the domains of natural numbers $\mathbb{N}$, non-negative rational numbers $\mathbb{Q}_{\geq 0}$, algebraic numbers $\mathrm{Alg}_{\geq 0}$, and real numbers $\mathbb{R}_{\geq 0}$; each with standard operations. For $\mathbb{N}$, we use the standard ordering; for the others, see below (after Theorem 1). The given domains are in fact positive cones of rings, but we rarely subtraction.

**Weighted automata.** A *weighted automaton* [2, 5, 15] $A = (D, \Sigma, Q, \lambda, \mu, \gamma)$ consists of a semiring $D$, an alphabet $\Sigma$, a set of states $Q$, and mappings

$$\lambda : Q \to D, \mu : (Q \times \Sigma \times Q) \to D, \gamma : Q \to D.$$

We picture such an automaton as a directed labelled graph (possibly with loops and parallel edges), with an edge $p \xrightarrow{x:w} q$ for each $\mu(p, x, q) = w$. An incoming edge (no source) $\xrightarrow{w} q$ denotes $\lambda(q) = w$, an outgoing edge (no target) $p \xrightarrow{w}$ denotes $\gamma(p) = w$. We omit all edges with weight 0. As an ongoing example for this section,



A *path* in the automaton is a sequence $q_0 \xrightarrow{x_1:w_1} q_1 \xrightarrow{x_2:w_2} \cdots \xrightarrow{x_n:w_n} q_n$. The *label* of this path is $x_1 x_2 \ldots x_n \in \Sigma^*$, and the *weight* of this path is $w_1 \cdot w_2 \cdot \cdots \cdot w_n \in D$. For instance, the path $1 \xrightarrow{a:1} 1 \xrightarrow{a:1} 2 \xrightarrow{b:3} 2$ has label $aab$ and weight 3. For each state $q$, there is an empty path from $q$ to $q$ with label $\epsilon$ and weight 1.

The function $\mu^* : (Q \times \Sigma^* \times Q) \to D$ computes the weight of a word $x = x_1 x_2 \ldots x_n$ from state $q_0$ to $q_n$ as the sum of the weights of all paths from $p$ to $q$ with label $x$:

$$\mu^*(q_0, x_1 \ldots x_n, q_n) = \sum_{q_1,\ldots,q_{n-1} \in Q} \prod_{1 \le k \le n} \mu(q_{k-1}, x_k, q_k)$$

For instance, $\mu^*(1, aab, 2)$ is computed from the paths $1 \xrightarrow{a:1} 1 \xrightarrow{a:1} 2 \xrightarrow{b:3} 2$ and $1 \xrightarrow{a:1} 2 \xrightarrow{a:1} 2 \xrightarrow{b:3} 2$, so the total weight is 6. We identify $\mu^*$ with $\mu$, and find it convenient to write $\mu(p, x, q) = d$ as $p \xrightarrow{x:d}_A q$.

The weight assigned by $A$ to a word $w$ is obtained by considering the functions $\lambda$ and $\gamma$ that give the weights for entering and leaving a state,

$$A(w) = \sum_{i,f \in Q} \lambda(i) \cdot \mu^*(i, w, f) \cdot \gamma(f).$$

In the example, $A(aab) = A(1, aab, 2) = 6$.

We say that state $q \in Q$ is *initial* if $\lambda(q) = 1$, and zero elsewhere; and $q$ is *final* if $\gamma(q) = 1$, and zero elsewhere. An automaton with unique initial state $i$ and unique final state $f$ is called $(i, f)$-*pointed*.

**Reduced automata.** We say that states $p$ *is connected to* state $q$ in $A$ if there is some $w \in \Sigma^*$ such that $\mu(p, w, q) \neq 0$. We write $p \to_A^* q$. An $(i, f)$-pointed automaton is called *reduced* if for each $q \in Q$, we have $i \to_A^* q \to_A^* f$. For each automaton $A$, there is a reduced automaton $A'$ that computes the same weight function as $A$. This $A'$ can be obtained from $A$ by simply deleting all unconnected states.

**Matrices.** The function $\mu$ of a weighted automaton can also be visualized as a mapping that assigns to each letter $x \in \Sigma$ a square matrix, also called $\mu(x)$, that is indexed by $Q \times Q$. For the example automaton, we have these matrices

$$\mu(a) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \mu(b) = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$$

This mapping can be extended from letters to words, by matrix multiplication: $\mu(x_1 \ldots x_n) = \mu(x_1) \cdot \ldots \cdot \mu(x_n)$, and this corresponds with the function $\mu^*$ defined above, that is, the entry at position $(p, q)$ in the matrix product $\mu(x_1 \ldots x_n)$ is the weight of the word $x_1 \ldots x_n$ from $p$ to $q$, as defined above. In the example, we compute

$$\mu(aab) = \mu(a) \cdot \mu(a) \cdot \mu(b) = \begin{pmatrix} 1 & 6 \\ 0 & 3 \end{pmatrix}.$$

If we view $\lambda$ as a row vector and $\gamma$ as a column vector, then $A(w) = \lambda \cdot \mu(w) \cdot \gamma$. For example,

$$A(aab) = \lambda \cdot \mu(aab) \cdot \gamma = \begin{pmatrix} 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 6 \\ 0 & 3 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 6$$

For $(i, f)$-pointed automata, $\lambda$ and $\gamma$ are unit vectors, so $A(w)$ is just the entry at position $(i, f)$ in the square matrix $\mu(w)$. Usually, $i$ is the first index and $f$ is last, so $(i, f)$ marks the top right position.

# 3 Termination Proofs from Weighted Automata

An $(i, f)$-pointed automaton $A$ is is called *weakly compatible with a rewriting system* $R$, if $\forall a \in \Sigma : \mu(i, a, i) \geq 1 \wedge \mu(f, a, f) \geq 1$ and for each rule $(l \to r) \in R$, and states $p, q \in Q$, we have $\mu(p, l, q) \geq \mu(p, r, q)$. The automaton is called *strictly compatible* with $R$ if additionally for each rule $(l \to r) \in R$, $\mu(i, l, f) > \mu(i, r, f)$. (In this paper we only use sub-semirings of $\mathbb{R}_{\geq 0}$, so all weights are non-negative.)

The main result of [13], written here in the language of weighted automata, is

**Theorem 1.** *If there is an* $\mathbb{N}$*-weighted automaton* $A$ *that is strictly compatible with a rewriting system* $R$ *and weakly compatible with a rewriting system* $S$*, then* $\mathrm{SN}(R/S)$*.* □

The intution is that for a rewrite step $xly \to xry$ using a rule $(l \to r) \in R$, each path $i \xrightarrow{x:*} i \xrightarrow{l:*} f \xrightarrow{y:*} f$ has strictly larger weight than the corresponding path $i \xrightarrow{x:*} i \xrightarrow{r:*} f \xrightarrow{y:*} f$. The total weight of $xly$ ($xry$, resp.) may include contributions from other paths, but for these we require a weak decrease. By strictness of "$<$" w.r.t. addition, we get a total decrease.

We give an example where Theorem 1 is applied with $S = \emptyset$.

**Example 1.** For the rewriting system $R = \{ab \to baa\}$, consider the $(1, 2)$-pointed automaton with transition matrices

$$\mu(a) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \mu(b) = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix}$$

By matrix multiplication, we compute

$$\mu(ab) = \begin{pmatrix} 1 & 3 \\ 0 & 3 \end{pmatrix}, \quad \mu(baa) = \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}$$

and we note $\mu(i, ab, f) = 3 > 2 = \mu(i, baa, f)$, and weak inequalities (in fact, equalities) elsewhere. This shows that the automaton is strictly compatible with $R$. From the theorem, we conclude $\mathrm{SN}(R/\emptyset)$, thus $\mathrm{SN}(R)$. □

In [8] it was observed that the theorem also holds if we replace $\mathbb{N}$ by $\mathbb{Q}_{\geq 0}$, and this easily extends to $\mathrm{Alg}_{\geq 0}$ and $\mathbb{R}_{\geq 0}$. Now $>$ is not well-founded on $\mathbb{Q}_{\geq 0}$ and indeed we use a different ordering: $x >_\epsilon y \iff x \geq \epsilon + y$ where

$$\epsilon = \inf \{\mu(i, l, f) - \mu(i, r, f) \mid (l \to r) \in R\}.$$

If the automaton is strictly compatible with a finite system $R$, then this is a positive number, and therefore $>_\epsilon$ is well-founded. Under the conditions of the theorem, we have $u \to_R v$ implies $\mu(i, u, f) >_\epsilon \mu(i, v, f)$.

The following is an easy observation:

**Lemma 1.** *If $A$ fulfills the conditions of Theorem 1, then there is a reduced automaton $A'$ with the same properties.*

*Proof.* We take $A'$ as the reduced automaton of $A$, obtained by deleting states that are unreachable from $i$ or do not reach $f$. Denote by $\mu'$ the transition function of $A'$. For states $p, q$ of $A'$, and letter $x \in \Sigma$, we have $\mu'(p, x, q) = \mu(p, x, q)$. Therefore, also for $w \in \Sigma^*$ we have $\mu'(p, w, q) = \mu(p, w, q)$. Since initial and final state of $A$ and $A'$ coincide (respectively), we are done. □

The following example shows an application of the theorem with non-empty $S$.

**Example 2.** Take $S = \{ab \to baa\}$ and $R = \{cb \to bcc\}$, and the $(1, 2)$-pointed automaton with matrices

$$\mu(a) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mu(b) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \mu(c) = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$$

We compute

$$\mu(cb) = \begin{pmatrix} 3 & 3 \\ 0 & 1 \end{pmatrix}, \quad \mu(bbc) = \begin{pmatrix} 3 & 2 \\ 0 & 1 \end{pmatrix}, \quad \mu(ab) = \mu(baa) = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

This shows that the automaton is strictly compatible with $R$ and weakly compatible with $S$, thus $\mathrm{SN}(R/S)$. □

Now we introduce an additional notation:

**Definition 1.** *For rewriting systems $R, S$ we write*

$$R \vdash S \iff R \supseteq S \wedge \mathrm{SN}((R \setminus S)/S).$$

**Example 3.** Example 1 shows that $\{ab \to baa\} \vdash \emptyset$. Example 2 shows that $\{ab \to baa, cb \to bbc\} \vdash \{ab \to baa\}$. □

**Proposition 1.** *$R \vdash S$ if and only if each infinite $R$-derivation ends with an infinite $S$-derivation.*

*Proof.* We have $\mathrm{SN}((R \setminus S)/S)$ if and only if each $R$-derivation contains only finitely many steps from $R \setminus S$. □

The notation "$\vdash$" supports the idea of composing termination proofs. Indeed,

**Proposition 2.** *The relation $\vdash$ is transitive.*

*Proof.* Given $R \vdash S$ and $S \vdash T$, we have to show that each infinite $R$-derivation ends with an infinite $T$-derivation. Assume there is an $R$-derivation with infinitely many steps from $R \setminus T$. If this derivation contains infinitely many steps from $R \setminus S$, then this contradicts $R \vdash S$. So it contains only finitely many steps from $R \setminus S$. After the last of these, we have an $S$-derivation. By $S \vdash T$, it contains only finitely many steps from $S \setminus T$, and then continues as an infinite $T$-derivation. $\square$

We obtain the following

**Corollary 1.** *If* $R \vdash^* \emptyset$*, then* $\mathrm{SN}(R)$*.* $\square$

Here is a typical application:

**Example 4.** By Example 3, we have

$$\{ab \to baa, cb \to bbc\} \vdash \{ab \to baa\} \vdash \emptyset,$$

thus the rewriting system on the left is terminating. $\square$

# 4 A Hierarchy of Relative Termination

We relate the general idea of relative termination, as denoted by "$\vdash$", with the idea of matrix interpretations.

**Definition 2.** *We denote by* $\mathfrak{M}(W, n)$ *the set of pairs of rewriting systems* $(R, S)$ *for which an automaton exists with weight domain* $W$ *and* $n$ *states that is strictly compatible with* $R \setminus S$ *and weakly compatible with* $S$*. We also write* $R \overset{\mathfrak{M}(W,n)}{\vdash} S$*.*

Indeed $\mathfrak{M}(W, n)$ is a relation on rewriting systems, and by Theorem 1, we have that $R \overset{\mathfrak{M}(W,n)}{\vdash} S$ implies $R \vdash S$.

For relations $\mathfrak{M}(W, n)$ we will make use of standard operations on relations like composition, iteration (exponentiation) and (reflexive and) transitive closure.

**Example 5.** By Example 1 we get $\{ab \to baa\} \overset{\mathfrak{M}(\mathbb{N},2)}{\vdash} \emptyset$. $\square$

By abuse of notation we sometimes write $R \in \mathfrak{M}(W, d)$ for $(R, \emptyset) \in \mathfrak{M}(W, d)$.

**Example 6.** By Examples 1,2, and the above abuse of notation,

$$\{ab \to baa, cb \to bbc\} \in \mathfrak{M}(\mathbb{N}, 2)^2.$$

The exponent 2 indicates that the termination proof is composed of two steps. $\square$

**Definition 3.** *The* matrix termination hierarchy *consists of the classes* $\mathfrak{M}(W, d)^s$ *of pairs of rewriting systems, where*

- $W \in \{\mathbb{N}, \mathbb{Q}_{\geq 0}, \mathrm{Alg}_{\geq 0}, \mathbb{R}_{\geq 0}\}$ *is a weight semiring,*

- *$d$ is a natural number $\geq 0$ giving the matrix dimension (automaton size),*
- *and $s$ is a natural number $\geq 1$ counting the proof steps $s$.*

We abbreviate $\cup_{n\geq 0}\mathfrak{M}(W,n)$ by $\mathfrak{M}(W)$. Then in our notation $\mathfrak{M}(\mathbb{N})$ is the set of all rewriting systems that have a one-step termination proof using some natural-weighted automaton. Using transitivity, $\mathfrak{M}(\mathbb{N})^+$ is the set of all systems with a multi-step termination proof using such automata.

We have these immediate observations:

**Proposition 3.** *1. If $n \leq n'$, then for all $W$, $\mathfrak{M}(W,n) \subseteq \mathfrak{M}(W,n')$.*

*2. If $W$ is a sub-semiring of $W'$, then for all $n$, $\mathfrak{M}(W,n) \subseteq \mathfrak{M}(W',n)$.*

*3. If $1 \leq s \leq s'$, then for all $W, n$*

$$\mathfrak{M}(W,n) \subseteq \mathfrak{M}(W,n)^{\leq s} \subseteq \mathfrak{M}(W,n)^{\leq s'} \subseteq \mathfrak{M}(W,n)^*.$$

*Proof.* (1) We can introduce useless states in the automaton. (2) Each $W$-interpretation is also a $W'$-interpretation. (3) Each sequence with $\leq s$ steps is also a sequence with $\leq s'$ steps. $\square$

While these statements are obvious, the following problems are not:

- Which of the obvious inclusions are strict?
- Are there non-obvious inclusions?
- Are the hierarchies (w.r.t. number of states, number of steps) infinite?
- What levels $\mathfrak{M}(W,n)^s$ are inhabited?

We will answer some of them in the rest of the paper.

# 5 Number of States

In this section we present a terminating rewriting system that needs large matrices for a termination proof. The construction works for any size, so we infer that the "matrix size hierarchy" is infinite.

We consider, for $d \geq 2$, the alphabet $\Sigma_d = \{s, 1, \ldots, d, f\}$. These are $d$ numbers and two extra letters $s, f$ (start and final). We take any enumeration $e_1, \ldots$ of even permutations of $\{1, \ldots, d\}$ and enumeration $o_1, \ldots$ of odd permuations of $\{1, \ldots, d\}$. Then consider the string rewriting system

$$R_d = \{se_kf \to so_kf \mid 1 \leq k \leq d!/2\}.$$

**Example 7.** For $d = 4$, we get the rule set

$$\begin{array}{lll} s1234f \to s2134f, & s2314f \to s2341f, & s3124f \to s1324f, \\ s3241f \to s3214f, & s1342f \to s3142f, & s3412f \to s3421f, \\ s2143f \to s1243f, & s2431f \to s2413f, & s1423f \to s4123f, \\ s4213f \to s4231f, & s4132f \to s1432f, & s4321f \to s4312f. \end{array}$$

**Lemma 2.** *There is no strict subset $S$ of $R_{2d}$ such that $(R_{2d}, S) \in \mathfrak{M}(\mathbb{N}, d)$.*

*Proof.* We use the Amitsur-Levitzki Theorem [14, 4]. It says that the elementary symmetric polynomial in $2d$ variables

$$s(x_1, \ldots, x_{2d}) = \sum_{\pi \text{ is a permutation of } \{1,\ldots,2d\}} (-1)^{\mathrm{sgn}(\pi)} x_{\pi(1)} \cdot \ldots \cdot x_{\pi(2d)}$$

is identically zero for $d \times d$-matrices.

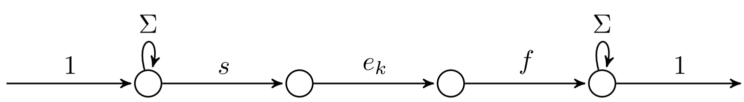
Any $d$-dimensional matrix interpretation $[\cdot]$ has

$$\sum_{(l \to r) \in R} ([l] - [r]) = [s] \left( \sum ([e_k] - [o_k]) \right) [f] = 0.$$

If $[\cdot]$ is weakly compatible with $R_{2d}$, then $\sum_{(l \to r) \in R_{2d}} ([l] - [r]) \geq 0$, and this implies $\forall (l \to r) \in R_{2d} : [l] = [r]$. So, $[\cdot]$ cannot be strictly compatible with any rule of $R_{2d}$. □

**Lemma 3.** *For $d' = 2d + 3$, $R_{2d} \in \mathfrak{M}(\mathbb{N}, d')^{(2d)!/2}$*

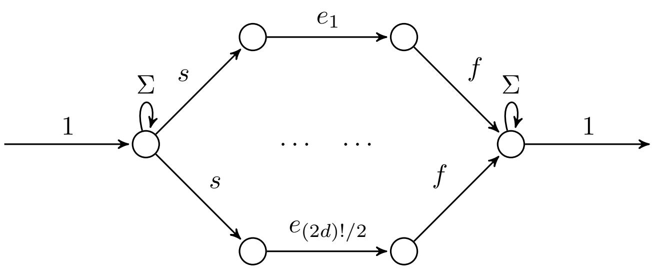
*Proof.* For each $k$, we give a matrix interpretation $[\cdot]$ of dimension $d'$ that is weakly compatible with all rules of $R_{2d}$ and strictly compatible with rule $se_k f \to so_k f$. The interpretation represents an automaton that just counts the number of factors $se_k f$. This is a word with $2d + 2$ letters, so counting can be done with $2d + 3$ states. The counting automaton consists of loops at initial and final state, and a path labelled $se_k f$ (and all unit weights) from initial to final state.




This works since $se_k f$ is not self-overlapping (no non-trivial prefix is equal to a suffix). The count reduces by one at each rewrite step, since there are no overlaps between $se_k f$ and $so_{k'} f$ either. Applying these interpretations for all $k$, in any order, gives the result: termination of $R_{2d}$ can be shown by a sequence of $(2d)!/2$ matrix interpretations of size $d'$. □

**Lemma 4.** *For $d' = 2 + (2d + 1)(2d)!/2$, we have $R_{2d} \in \mathfrak{M}(\mathbb{N}, d')$.*

*Proof.* We build an automaton that contains all the automata constructed in the proof of Lemma 3 in parallel.




It has one initial and one final state, and $(2d)!/2$ paths each using $(2d+1)$ individual states. □

As a corollary, we obtain

**Theorem 2.** *For each $W \in \{\mathbb{N}, \mathbb{Q}_{\geq 0}, \mathrm{Alg}_{\geq 0}, \mathbb{R}_{\geq 0}\}$: The hierarchy $\mathfrak{M}(W,d)_{d=0,1,\dots}$ is infinite.*

*Proof.* Assume, to the contrary, that there is $d$ such that $\mathfrak{M}(W,d) = \mathfrak{M}(W,d+1) = \dots$ By Lemma 2, the system $R_{2d}$ is not in $\mathfrak{M}(W,d)$, and by Lemma 4, $R_{2d} \in \mathfrak{M}(W,d')$ for some $d' > d$. □

# 6 Small Automata

We have more information on the lower levels of the hierarchy:

**Proposition 4.** *These inclusions are strict:*

$$\mathfrak{M}(\mathbb{N},0) \subset \mathfrak{M}(\mathbb{N},1) \subset \mathfrak{M}(\mathbb{N},2) \subset \mathfrak{M}(\mathbb{N},3).$$

*Proof.* We prove $R_1 = \{a \to b\} \in \mathfrak{M}(\mathbb{N},1) \setminus \mathfrak{M}(\mathbb{R}_{\geq 0},0)$. A strictly compatible 1-dimensional interpretation of the required shape is given by $[a] = 2, [b] = 1$. Any interpretation in $\mathfrak{M}(\mathbb{N},0)$ is necessarily constant, so it is strictly compatible only with the empty set of rules, and not with $R_1$.

We prove $R_2 = \{ab \to ba\} \in \mathfrak{M}(\mathbb{N},2) \setminus \mathfrak{M}(\mathbb{R}_{\geq 0},1)$. A strictly compatible 2-dimensional interpretation is given by

$$[a] = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, [b] = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Any one-dimensional matrix interpretation $[\cdot]$ is commutative, so $[ab] = [ba]$ and it cannot be strictly compatible with $R_2$.

We prove $R_3 = \{aa \to aba\} \in \mathfrak{M}(\mathbb{N},3) \setminus \mathfrak{M}(\mathbb{R}_{\geq 0},2)$. A strictly compatible 3-dimensional interpretation is

$$[a] = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, [b] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Any two-dimensional interpretation $[\cdot]$ of the required shape has main diagonal entries $\geq 1$ and thus $[aba] \geq [aa]$, contradicting strict compatibility with $R_3$. □

If the automata under consideration have only one state, the weight domain is not really important, and the "step hierarchy" collapses.

**Lemma 5.** $\mathfrak{M}(\mathbb{R}_{\geq 0}, 1) \subseteq \mathfrak{M}(\mathbb{N}, 1)$.

*Proof.* An interpretation $[\cdot]$ by a one-state $\mathbb{R}_{\geq 0}$-weighted automaton corresponds to a multiplicative weight assignment (the weight of a word is the product of the weight of its letters). Note that all weights are positive, by definition. Taking logarithms, we get an additive assignment (the weight of a word is the sum of its letter weights). The conditions of weak and strict compatibility give rise to a system of linear equalities and inequalities between letter weights. The coefficients are natural numbers (namely, numbers of occurences of letters in sides of rules). If such a system has any solution at all, then it also has a rational solution. Since the system is moreover homogenous (the linear functions contain no absolute parts), any rational solution can be scaled to give an integer solution. In fact the components are naturals, since weights must be non-negative. From natural additive weights we can get back to multiplicative weights by exponentiation. If we take any natural base, then the weights are natural (they are powers of the base). $\square$

**Example 8.** For $R = \{aaa \to bca\}, S = \{b \to cac\}$ we obtain the system of inequalities

$$\begin{aligned} \log[a] \geq 0 \wedge \log[b] \geq 0 \wedge \log[c] \geq 0 \\ \wedge 2\log[a] - \log[b] - \log[c] > 0 \wedge -\log[a] + \log[b] - 2\log[c] \geq 0. \end{aligned}$$

One solution is $\log[a] = 4, \log[b] = 6, \log[c] = 1$. We can take base 2 and obtain multiplicative weights $[a] = 16, [b] = 64, [c] = 2$. This proves $R \cup S \,\big|\!\overset{\mathfrak{M}(\mathbb{N},1)}{\overline{\phantom{xxxx}}}\, S$. $\square$

As a corollary to Lemma 5, we obtain

**Theorem 3.** $\mathfrak{M}(\mathbb{N}, 1) = \mathfrak{M}(\mathbb{Q}_{\geq 0}, 1) = \mathfrak{M}(\mathrm{Alg}_{\geq 0}, 1) = \mathfrak{M}(\mathbb{R}_{\geq 0}, 1)$. $\square$

Now we consider the number of proof steps when using one-state automata. We show that two-step proofs are not stronger than one-step proofs.

**Lemma 6.** $\mathfrak{M}(\mathbb{N}, 1)^2 = \mathfrak{M}(\mathbb{N}, 1)$

*Proof.* We are given a two-step one-dimensional termination proof, and we need to construct an equivalent one-step proof. Assume weight function $f$ is strictly compatible with $R$ and weakly compatible with $S \cup T$, and weight function $g$ is strictly compatible with $S$ and weakly compatible with $T$. We construct a weight function $h$ that is strictly compatible with $R \cup S$ and weakly compatible with $T$, as follows. (In light of the previous, we write the weight functions additively.)

We will define

$$h(x) = f(x) \cdot c + g(x),$$

for a suitable natural number $c > 0$. Such an interpretation $h$ is weakly compatible with $T$, since both $f$ and $g$ have this property. Interpretation $h$ is strictly compatible with $S$, since $f$ is weakly compatible with $S$ and $g$ is strictly compatible with $S$.

We put
$$c = 1 + \sup\{\max(0, g(r) - g(l)) \mid (l \to r) \in R\}$$
This is one plus the maximal increase of $g$ weights, for $R$ rules.

It remains to check that $h$ is strictly compatible with $R$. If $u \to_R v$, then $f(u) - f(v) \geq 1$ by strict compatibility (and using that weights are natural), and $g(u) - g(v) \geq -c + 1$ by definition of $c$. By definition of $h$ we get $h(u) - h(v) \geq c + (-c + 1) = 1$, and this proves the claim. $\square$

**Example 9.** We have $\{a^2 \to b^3, b^5 \to a^3\} \mid\!\xrightarrow{\mathfrak{M}(\mathbb{N},1)} \{b^5 \to a^3\}$ by the interpretation $f : a \mapsto 5, b \mapsto 3$; and $\{b^5 \to a^3\} \mid\!\xrightarrow{\mathfrak{M}(\mathbb{N},1)} \emptyset$ by $g : a \mapsto 0, b \mapsto 1$. Since $g(a^2) = 0, g(b^3) = 3$, we put $c = 4$ and get $\{a^2 \to b^3, b^5 \to a^3\} \mid\!\xrightarrow{\mathfrak{M}(\mathbb{N},1)} \emptyset$ by $h : a \mapsto 20, b \mapsto 13$. $\square$

As a corollary to Lemma 6, we obtain

**Theorem 4.** $\mathfrak{M}(\mathbb{N}, 1)^* = \mathfrak{M}(\mathbb{N}, 1)$ $\square$

# 7 Choice of Weight Domain

We compare the power of matrix interpretations w.r.t. the weight domain.

We give an example $(R \cup S, S) \in \mathfrak{M}(\mathbb{Q}_{\geq 0}, 3)^2 \setminus \mathfrak{M}(\mathbb{N})^*$, that is, with a two-step termination proof of rational-weighted automata of size 3, but no natural-weighted termination proof of any size and number of steps.

The rewriting systems are
$$R = \{baa \to abc, ca \to ac, cb \to ba\}, S = \{\epsilon \to b\}.$$

**Lemma 7.** $(R \cup S, S) \in \mathfrak{M}(\mathbb{Q}_{\geq 0}, 3) \circ \mathfrak{M}(\mathbb{N}, 1)$.

*Proof.* We use the following interpretation
$$[a] = \begin{pmatrix} 1 & 1 & 0 \\ 0 & \frac{5}{2} & 6 \\ 0 & 0 & 1 \end{pmatrix}, \ [b] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ [c] = \begin{pmatrix} 1 & 2 & 0 \\ 0 & \frac{5}{2} & \frac{7}{2} \\ 0 & 0 & 1 \end{pmatrix}.$$
giving these interpretations for the rules:
$$[baa] = \begin{pmatrix} 1 & \frac{7}{2} & 6 \\ 0 & \frac{25}{8} & \frac{21}{2} \\ 0 & 0 & 1 \end{pmatrix} \qquad [abc] = \begin{pmatrix} 1 & \frac{13}{4} & \frac{7}{4} \\ 0 & \frac{25}{8} & \frac{83}{8} \\ 0 & 0 & 1 \end{pmatrix}$$
$$[ca] = \begin{pmatrix} 1 & 6 & 12 \\ 0 & \frac{25}{4} & \frac{37}{2} \\ 0 & 0 & 1 \end{pmatrix} \qquad [ac] = \begin{pmatrix} 1 & \frac{9}{2} & \frac{7}{2} \\ 0 & \frac{25}{4} & \frac{59}{4} \\ 0 & 0 & 1 \end{pmatrix}$$
$$[cb] = \begin{pmatrix} 1 & 1 & 0 \\ 0 & \frac{5}{4} & \frac{7}{2} \\ 0 & 0 & 1 \end{pmatrix} \qquad [ba] = \begin{pmatrix} 1 & 1 & 0 \\ 0 & \frac{5}{4} & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

This proves $R \cup S \stackrel{\mathfrak{M}(\mathbb{Q}_{\geq 0},3)}{\longmapsto} \{cb \to ba, \epsilon \to b\}$.

By another interpretation $[a] = [b] = 1, [c] = 2$ we get

$$\{cb \to ba, \epsilon \to b\} \stackrel{\mathfrak{M}(\mathbb{N},1)}{\longmapsto} \{\epsilon \to b\}.$$

□

**Lemma 8.** *There is no $T \subset R \cup S$ such that $(R \cup S, T) \in \mathfrak{M}(\mathbb{N})$.*

*Proof.* Assume there is a matrix interpretation of any size (an $\mathbb{N}$-weighted automaton with any number of states) that is weakly compatible with $R \cup S$ and strictly compatible with one of the rules from $R$. (It cannot be strictly compatible with $S$ since $S$ is non-terminating.)

We assume the automaton is reduced. All edges labelled by $b$ are unit loops: they go from some state $q$ to $q$ and have weight one. The reason is that weak compatibility with $S$ requires $[\epsilon] \geq [b]$, but $[\epsilon]$ is the unit matrix, for any interpretation $[\cdot]$.

The plan of the proof is now: we show that the interpretation of $b$ is indeed the unit matrix (each state has a $b$ loop), and then we derive a contradiction from that.

Consider the subset $A$ of states that are reachable from the initial state $i$ of the automaton by $a$ edges. (Here and in the following, when we speak of an edge, then we mean that it has a non-zero weight.)

We claim that state $q$ in $A$ is also reachable from $i$ by $c$ edges, and has a $b$ loop. The proof is by induction on the distance to $i$. Assume the transition $p \xrightarrow{a} q$ has weight $> 0$, and the claim holds true for $p$. Then we have a path $p \xrightarrow{b} p \xrightarrow{a} q$. By weak compatibility with rule $cb \to ba$, there must be a path $p \xrightarrow{cb} q$. Since $b$ transitions are loops, this can only take the form of $p \xrightarrow{c} q \xrightarrow{b} q$.

Every state $r$ reachable from $i$ by any mixture of $a$ and $c$ steps is also in $A$ (that is, reachable by $a$ steps alone): assume by induction that there is a transition $q \xrightarrow{c} r$ and the claim holds true for $q$. Then $q$ is in $A$, so there is a transition $p \xrightarrow{a} q$, thus a path $p \xrightarrow{a} q \xrightarrow{b} q \xrightarrow{c} r$. By weak compatibility with $baa \to abc$, there must be a path $p \xrightarrow{baa} r$. Since a $b$ edge is a loop, there is some $q'$ such that the path is $p \xrightarrow{b} p \xrightarrow{a} q' \xrightarrow{a} r$. This shows that $r$ is in $A$, since it is reachable from $p \in A$ by $a$ steps.

The final state $f$ does also belong to $A$: since the interpretation was assumed to be strictly compatible with some rule $(l \to r) \in R$, there must be a path $i \xrightarrow{l} f$. Since $b$ steps (loops) are irrelevant for reachability, the claim follows.

Since the automaton was reduced, we have shown that *each* state belongs to $A$, thus each state has a $b$ loop. This implies that the interpretation of letter $b$ is the unit matrix. Now we replace $b$ by $\epsilon$ in $R$, obtaining $R' = \{aa \to ac, ca \to ac, c \to a\}$. We claim that the automaton is weakly compatible with $R'$ and strictly compatible with at least one rule from $R'$. This holds true since $[baa] = [aa]$ etc., and the automaton was assumed to be weakly compatible with $R$ and strictly compatible with at least one rule from $R$.

On the other hand, there is a looping $R'$-derivation

$$\underline{aa}a \to a\underline{ca} \to aa\underline{c} \to aaa \to \dots$$

that uses each rule of $R'$ infinitely often. This contradicts the fact that the automaton is strictly compatible with at least one rule of $R'$, since this rule must be relatively terminating w.r.t. the others.

In all, this proves that the interpretation (automaton) does not exist. □

As a corollary to Lemma 7 and Lemma 8, we get

**Theorem 5.** *$\mathfrak{M}(\mathbb{Q}_{\geq 0}, 3)^2 \setminus \mathfrak{M}(\mathbb{N})^*$ is non-empty.*

## 8 Number of Proof Steps

We first recall an example that shows that two-step proofs (even of dimension two) are more powerful than one-step proofs (of any dimension). Then we generalize, and show that the "step hierarchy" is infinite. The underlying reason is derivational complexity. The following is a basic fact of linear algebra:

**Lemma 9.** *Let $A$ be any finite set of square matrices of identical shape. The coefficients in a product of any $k$ matrices from $A$ are bounded by an exponential function of $k$.* □

This will be used in the following form:

**Corollary 2.** *For disjoint rewriting systems $R$ and $S$: if there is a family of $R \cup S$-derivations*

$$d_1 : w_{1,1} \to \dots \to w_{1,n_1}, d_2 : w_{2,1} \to \dots \to w_{2,n_2}, \dots$$

*such that the number of $R$ steps in $d_k$ is not bounded by an exponential function of $|w_{k,1}|$, then $(R \cup S, S) \notin \mathfrak{M}(\mathbb{R}_{\geq 0})$.*

*Proof.* Assume to the contrary that there is some $(i, f)$-pointed automaton $A$ with the given properties: strictly compatible with $R$ and weakly compatible with $S$. Then $u \to_R v$ implies $\mu(i, u, f) > \mu(i, v, f)$, and $u \to_S v$ implies $\mu(i, u, f) \geq \mu(i, v, f)$, So the number of $R$ steps in the derivation starting in $w_{k,1}$ is bounded by $\mu(i, w_{k,1}, f)$, which is an exponential function by Lemma 9, contradicting the assumption. □

**Lemma 10.** *There is $R \in \mathfrak{M}(\mathbb{N}, 2)^2 \setminus \mathfrak{M}(\mathbb{R}_{\geq 0})$.*

*Proof.* The following example is already presented in [13]. Let $R = \{ab \to baa, cb \to bbc\}$. There are derivations (for each $k \geq 0$):

$$a^k b \to^* ba^{2k}, ab^k \to^* b^k a^{2^k}$$
$$cb^k \to^* b^{2k} c, c^k b \to^* b^{2^k} c^k$$
$$ac^k b \to^* ab^{2^k} c^k \to^* b^{2^k} a^{2^{2^k}} c^k$$

The resulting string has length $2^{2^k}$, thus the derivation also took this number of steps, since each step extends the length by one.

By Lemma 9, there can be no matrix interpretation that is strictly compatible with both rules of $R$.

On the other hand we have $R \in \mathfrak{M}(\mathbb{N}, 2)^2$ by Example 6. □

We modify, and generalize this example. For any $n \geq 1$, define a rewriting system over alphabet $\Sigma_n = \{1, \ldots, n\}$ by

$$R_n = \{i(i-1) \to (i-1)^2 i \mid 2 \leq i \leq n\} \cup \{(i-1) \to (i-2) \mid 3 \leq i \leq n\}.$$

For $n \geq 2$, this system has $2n - 3$ rules. Note that $R_1$ is empty.

**Example 10.** $R_3 = \{32 \to 223, 21 \to 112, 2 \to 1\}$. □

**Lemma 11.** *For each $i$ and $k$, there is a $R_n$-derivation from $i^k(i-1)$ to some word containing $(i-1)^{2^{k-1}}(i-2)$ as a factor and using each of the rules $i(i-1) \to (i-1)^2 i$ and $(i-1) \to (i-2)$ at least $2^{k-1}$ times.*

*Proof.* For each $l$, we have $i(i-1)^l \to^l (i-1)^{2l} i$, and by iteration,

$$i^k(i-1) \to^{2^{k+1}-1} (i-1)^{2^k} i^k.$$

Now we apply rule $(i-1) \to (i-2)$ for $2^{k-1}$ times to get

$$(i-1)^{2^{k-1}}(i-2)^{2^{k-1}} i^k.$$

□

Using Lemma 11 repeatedly, we get

**Lemma 12.** *For each $i$ and $k$, there is a $R_n$-derivation from $i^k(i-1)$ using at least* $\exp(\exp(k))$ *steps of each rule $j(j-1) \to (j-1)^2 j$ and $(j-1) \to (j-2)$, for $j < i$.* □

**Lemma 13.** *If a matrix interpretation is weakly compatible with $R_n$ and strictly compatible with some subset $S \subseteq R_n$, then $R_{n-1} \cap S = \emptyset$.*

*Proof.* By Lemma 12, there is a family of derivations that uses all rules in $R_{n-1}$ more than exponentially often. By Corollary 2, the claim follows. □

**Lemma 14.** $R_{n+2} \notin \mathfrak{M}(\mathbb{R}_{\geq 0})^n$.

*Proof.* $R_2 \notin \mathfrak{M}(\mathbb{R}_{\geq 0})^0$ since $R_2$ is non-empty. By Lemma 13, if $R_{n+1} \vdash^{\mathfrak{M}(\mathbb{R}_{\geq 0})} R'$, then $R_n \subseteq R'$. Then the claim follows by induction. □

**Lemma 15.** $R_{n+1} \in \mathfrak{M}(\mathbb{N}, 2)^n$.

*Proof.* $R_1 \in \mathfrak{M}(\mathbb{R}_{\geq 0}, 2)^0$ since $R_1$ is empty. The following interpretation

$$[n] = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}, [n-1] = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad \text{for } j \leq n-2 : [j] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

shows $R_n \left|\!\xrightarrow{\mathfrak{M}(\mathbb{N},2)}\right. R_{n-1}$. Then the claim follows by induction. □

We obtain as a corollary:

**Theorem 6.** *Each inclusion* $\mathfrak{M}(\mathbb{R}_{\geq 0})^s \subset \mathfrak{M}(\mathbb{R}_{\geq 0})^{s+1}$ *is strict. The "proof length hierarchy"* $(\mathfrak{M}(\mathbb{R}_{\geq 0})^s)_{s=1,2,\ldots}$ *is infinite.*

# 9 Discussion

**Summary.** Termination proofs by weighted (word and tree) automata are being investigated only since 2006. (resp. 2003, if we include the Match Bounds method, which later turned out to be related to the Min/Max semiring.) The focus of investigation mainly was the construction of automata, with the goal of actually implementing and running the algorithms. This has been achieved rather successfully: the various "matrix methods" play a decisive role in the regular Termination Competitions.

With the present paper, we start a systematic investigation into the expressiveness of these methods as *proof systems*.

To this end, we have defined a two-dimensional hierarchy $\mathfrak{M}(W, d)^s$ for termination proofs for string rewriting via weighted word automata, and we proved that the hierarchy is infinite in both directions.

Still, we have no exact information on which levels are actually inhabited (notice the "gap" from $d$ to $d'$ in Lemma 4), and which levels (if any) are decidable. These questions remain as challenging open problems. Other extensions are at hand, and we list a few.

**Decidability.** As noted in the proof of Lemma 5, existence of a one-dimensional interpretation is equivalent to the feasibility of a system of linear inequalities. Therefore, $\mathfrak{M}(\mathbb{R}_{\geq 0}, 1)$ is decidable.

For larger dimensions, the weak and strong compatibility conditions give rise to a system of inequalities between polynomials, where the unknowns are the matrix entries (the weights of the automaton transitions). Then we can use Tarski's decision method [16], and obtain that for each $d$, $\mathfrak{M}(\mathbb{R}_{\geq 0}, d)$ is decidable.

In fact if the system of polynomial (in)equalities has a solution, then it also has a solution in algebraic numbers. So we don't really need real numbers: for each $d$, $\mathfrak{M}(\mathrm{Alg}_{\geq 0}, d) = \mathfrak{M}(\mathbb{R}_{\geq 0}, d)$.

Except for these immediate observations, we have no information (and no intuition) on decidability of any $\mathfrak{M}(W, d)$.

**Non-strict semirings.** One can use semirings with non-strict addition for termination, e.g. the max/plus semiring, or the max/min semiring [17]. Again, a corresponding hierarchy can be defined but it needs different methods than presented here. If we try the construction of Section 5, using a suitable polynomial identity $\sum[l_i] = \sum[r_i]$ in the arctic semiring, we can no longer infer from $\forall i : [l_i] \geq [r_i]$ that $\forall i : [l_i] = [r_i]$, since arctic addition is not strictly monotonic in its arguments. For the proof step hierarchy we cannot use the methods of Section 8, because of the following: Arctic interpretations give a linear bound on derivational complexity, and by the reasoning in Lemma 6, even a combination of such interpretations might not achieve more than linear derivation lenghts. So, the "arctic termination hierarchy" is a subject of further study.

**Term rewriting.** The method of interpretation via weighted automata has been generalized to term rewriting [6]. The definition of our hierarchy can be generalized as well. Still we note that matrix interpretations for term rewriting use a rather restricted form of weighted tree automata.

**Parallel composition of proofs.** Our hierarchy uses the concept of combining termination proofs *sequentially.* There are methods of proving termination that correspond to a *parallel* composition: after the Dependency Pairs transformation [1], the resulting relative termination problem can be decomposed into several independent sub-problems, corresponding to the strictly connected components of the dependency graph [12]. In all, a termination proof thus gets a tree structure. While we presently compare proof sequences by length, proof trees should be compared structurally, e.g. with respect to embedding.

# References

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

[2] Berstel, Jean and Reutenauer, Christophe. *Noncommutative Rational Series With Applications.* `http://www-igm.univ-mlv.fr/~berstel/LivreSeries/LivreSeries.html`, 2009.

[3] Book, Ronald V and Otto, Friedrich. *String Rewriting Systems.* Springer, 1993.

[4] Drensky, Vesselin and Formanek, Edward. *Polynomial Identity Rings.* Birkhäuser, 2004.

[5] Droste, Manfred, Kuich, Werner, and Vogler, Heiko, editors. *Handbook of Weighted Automata.* Springer, 2009. in preparation.

[6] Endrullis, Jörg, Waldmann, Johannes, and Zantema, Hans. Matrix interpretations for proving termination of term rewriting. In Furbach, Ulrich and

Shankar, Natarajan, editors, *IJCAR*, volume 4130 of *Lecture Notes in Computer Science*, pages 574–588. Springer, 2006.

[7] Fuchs, Laszlo. *Partially Ordered Algebraic Systems.* Addison-Wesley, 1962.

[8] Gebhardt, Andreas, Hofbauer, Dieter, and Waldmann, Johannes. Matrix evolutions. In Hofbauer, Dieter and Serebrenik, Alexander, editors, *Proc. Workshop on Termination, Paris*, 2007.

[9] Gebhardt, Andreas and Waldmann, Johannes. Weighted automata define a hierarchy of terminating string rewriting systems. In Droste, Manfred and Vogler, Heiko, editors, *Proc. Weighted Automata Theory and Applications, Dresden*, pages 34–35, 2008.

[10] Geser, Alfons. *Relative termination.* Dissertation, Fakultät für Mathematik und Informatik, Universität Passau, Germany, 1990. 105 pages. Also available as: Report 91-03, Ulmer Informatik-Berichte, Universität Ulm, 1991.

[11] Golan, Jonathan S. *Semirings and their Applications.* Kluwer, 1999.

[12] Hirokawa, Nao and Middeldorp, Aart. Dependency pairs revisited. In van Oostrom, Vincent, editor, *RTA*, volume 3091 of *Lecture Notes in Computer Science*, pages 249–268. Springer, 2004.

[13] Hofbauer, Dieter and Waldmann, Johannes. Termination of string rewriting with matrix interpretations. In Pfenning, Frank, editor, *RTA*, volume 4098 of *Lecture Notes in Computer Science*, pages 328–342. Springer, 2006.

[14] Kanel-Belov, Alexei and Rowen, Louis Halle. *Computational Aspects of Polynomial Identities.* AK Peters, 2005.

[15] Kuich, Werner. Semirings and formal power series. In *Handbook of Formal Languages*, volume 1, pages 609–677. Springer, 1997.

[16] Tarski, Alfred. A decision method for elementary algebra and geometry. Manuscript. Santa Monica, CA: RAND Corp., 1948.

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

[18] Zantema, Hans. Termination. In Terese, editor, *Term Rewriting Systems*, pages 181–259. Cambridge University Press, 2003.

*Received 1st August 2008*