Journal of Automata, Languages and Combinatorics **10** (2005) 2/3, 107–157


# BOUNDS FOR TREE AUTOMATA WITH POLYNOMIAL COSTS[1]

Björn Borchardt[2]
*Department of Computer Science, Technische Universität Dresden*
*D–01062 Dresden, Germany*
*e-mail:* `borchard@tcs.inf.tu-dresden.de`

Zoltán Fülöp[1] and Zsolt Gazdag[3]
*Department of Computer Science, University of Szeged*
*Árpád tér 2., H–6720 Szeged, Hungary*
*e-mail:* `{fulop,gazdag}@inf.u-szeged.hu`

and

Andreas Maletti[2]
*Department of Computer Science, Technische Universität Dresden*
*D–01062 Dresden, Germany*
*e-mail:* `maletti@tcs.inf.tu-dresden.de`

ABSTRACT

We consider tree automata with costs over semirings in the sense of (Seidl, 1994). We define the concept of a finitely factorizing semiring and of a monotonic semiring, both as the generalization of well-known particular semirings, and show that the cost-finiteness of tree automata with costs over finitely factorizing and monotonic semirings is decidable. We show that, for tree automata with costs over finitely factorizing and naturally ordered semirings, cost-finiteness and boundedness are equivalent. Hence it is also decidable whether a tree automaton with costs over a finitely factorizing, monotonic, and naturally ordered semiring is bounded with respect to the natural order. With this we generalize the results of (Seidl, 1994) concerning the decidability of the boundedness of tree automata with costs over the classical semiring of natural numbers and the $(\max, +)$-semiring of natural numbers.

*Keywords:* Polynomials over semirings, tree automata with costs, cost-finiteness

[1]Full version of a submission presented at the Workshop on *Weighted Automata: Theory and Applications* (Dresden University of Technology, Germany, June 1–5, 2004).

[2]Research of these authors was financially supported by the German Research Foundation (DFG, GRK 334/3).

[3]Research of these authors was supported by the exchange program of DAAD and MÖB under grant No. UH/2607-45, by the Herbert Quand Foundation, by the Foundation Teaching and Research in Informatics, and by the Hungarian Scientific Foundation (OTKA) under Grant T 046686.

## 1. Introduction

The idea of equipping the transitions of a finite state automaton $M$ with costs (or with weights or multiplicities) was introduced in [30]. Every transition of $M$ has a cost taken from a semiring $(A, \oplus, \odot, \mathbf{0}, \mathbf{1})$. Then, for every input word $w$ and run of $M$ on $w$, the cost of this run is the multiplication of the costs associated with the transitions of this run. Moreover, the cost $c(w)$ of $w$ is the sum of the costs of all successful, i.e., accepting, runs of $w$. We call such a model a finite automaton with costs. One might be interested in the question whether a finite automaton $M$ with costs is bounded in the following sense. Assume that there is a partial order $\preceq$ over the carrier set $A$ of the underlying semiring. Now $M$ is bounded with respect to $\preceq$ if there is an element $a \in A$ (an upper bound) such that $c(w) \preceq a$ for every word $w$ accepted by $M$. Boundedness (or limitedness) theorems (e.g., [17, 23, 26, 27, 22]) have successfully been applied to solve a number of problems, including representation problems and star-height problems [14, 15]. For example, the decidability of the boundedness of distance automata [13], which are restricted finite automata with costs over the $(\min, +)$-semiring, motivated a lot of further research [16, 17, 18, 22, 34, 35].

In this paper we will be interested in the decidability of boundedness of tree automata with costs over a semiring. Therefore we assume that each semiring $(A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ which serves as a cost-set of a tree automaton is computable in the sense that its both operations $\oplus$ and $\odot$ are computable. Next we recall the concept of a tree automaton.

Finite state automata are generalized to tree automata independently in [3] and [33] in such a way that, instead of the input alphabet, they take a ranked alphabet $\Sigma$. Thus the underlying structure, instead of an unoid, is a $\Sigma$-algebra and the inputs to the tree automaton are trees over $\Sigma$. In fact a tree automaton is a tuple $M = (Q, \Sigma, \delta, F)$, where $Q$ is the finite set of states, $\Sigma$ is the input ranked alphabet, $\delta$ is the set of transitions, and $F \subseteq Q$ is the set of final states. A transition of $M$ is a tuple $(q_1, \ldots, q_k, \sigma, q)$, where $\sigma$ is a symbol of rank $k$ from $\Sigma$ and $q_1, \ldots, q_k, q \in Q$. Next we will define the semantics of $M$. An input tree to $M$ is a tree over $\Sigma$ without variables. The transition specified before describes the state behaviour of $M$ at a $\sigma$-node of the input tree. Next we define the concept of the set of computations of $M$ over a $\Sigma$-tree. Note that it would be sufficient to define it for a tree without variables, however for later purposes, we define it more generally. For a $\Sigma$-tree $s$ with variables in $\{x_1, \ldots, x_n\}$ and states $q_1, \ldots, q_n, q \in Q$, we define the set $\Psi^q_{q_1 \cdots q_n}(s)$ of $(q_1, \ldots, q_n, q)$-computations over $s$ to be the set of trees over $\delta$ (as a ranked alphabet) and $\{x_1, \ldots, x_n\}$ which respect the state behaviour of the transitions. By the set $\Psi^q_{q_1 \cdots q_n}$ of $(q_1, \ldots, q_n, q)$-computations, without referring to an input tree, we mean the union of the sets $\Psi^q_{q_1 \cdots q_n}(s)$ for all trees $s$ with variables in $\{x_1, \ldots, x_n\}$. In order to define the semantics of $M$ it is sufficient to consider $(\varepsilon, q)$-computations (or: $q$-computations) over an input tree $s$ without variables. A $q$-computation is accepting if $q \in F$. A tree $s$ is accepted by the tree automaton $M$ if there is an accepting computation over $s$.

Tree automata have also been studied intensively; see among others [4, 28, 32, 5, 9, 10]. Recently tree automata have been generalized to tree automata with costs in

[1, 25, 2]. Note that the computation model was called a tree automaton with costs in none of these papers, however we would like to use this term in our paper.

The idea is similar to what was invented for the string case in [30]. In fact, every transition $(r_1, \ldots, r_k, \sigma, q)$ of $M$ has a cost $a$, which is an element of a semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$. Now let $s = \sigma(s_1, \ldots, s_k)$ be an input tree. The cost of a computation $(r_1, \ldots, r_k, \sigma, q)(\psi_1, \ldots, \psi_k)$ over $s$ is the product $a \odot c(\psi_1) \odot \cdots \odot c(\psi_k)$, where $a \in A$ is the cost of the transition $(r_1, \ldots, r_k, \sigma, q)$ and $c(\psi_i) \in A$ is the cost of the computation $\psi_i$ over $s_i$ for every $1 \leq i \leq k$. Moreover, the cost of $s$ is the sum of the costs of all accepting computations of $M$ over $s$.

In the paper [31], which will be in the focus of our interest, tree automata with costs were defined in a slightly different way. The cost of a transition $(r_1, \ldots, r_k, \sigma, q)$ of a tree automaton $M$ is given by a polynomial over the semiring $\mathcal{A}$ with variables in $\{x_1, \ldots, x_k\}$. Thus the costs of the transitions can be specified as a cost function $c : \delta \longrightarrow P(A, X)$, where $P(A, X)$ is the set of polynomials over $A$ and the set of variables $X$. Then the cost of a computation $(r_1, \ldots, r_k, \sigma, q)(\psi_1, \ldots, \psi_k)$ of $M$ becomes $c((r_1, \ldots, r_k, \sigma, q))(c(\psi_1), \ldots, c(\psi_k))$, where the polynomial $c((r_1, \ldots, r_k, \sigma, q))$ with variables in $\{x_1, \ldots, x_k\}$ is considered as a function of type $A^k \longrightarrow A$. The set of accepting costs of $M$ is denoted by $c(M)$ and is defined to be the set of costs of all accepting computations of $M$. Note that this way of computing the cost is more general than the approach of [1, 25, 2] because the cost of the transition $(r_1, \ldots, r_k, \sigma, q)$ in those papers is described by a special polynomial of the form $a \cdot x_1 \cdot \ldots \cdot x_k$. On the other hand, we think because the author of [31] concentrates only on deciding the boundedness of the set $c(M)$, the cost of an input tree $s$ is not considered in [31].

Now we turn to cost-finiteness and boundedness of tree automata with costs. We call a tree automaton $M$ with costs over a semiring $\mathcal{A}$ cost-finite if the set of all accepting costs is finite. Moreover, we call $M$ bounded with respect to a partial order $\preceq$ on $A$ if an upper bound $a \in A$ of the set of all accepting costs exists. Note that cost-finiteness and boundedness are trivial in finite semirings.

In this paper we deal with the problem of deciding cost-finiteness of a tree automaton with costs. Our motivation was to generalize the decidability results of [31]. In that paper decidability of boundedness with respect to $\leq$ was proved for tree automata with costs over three particular semirings: the classical semiring Nat, the $(\max, +)$-semiring Arct, and the $(\min, +)$-semiring Trop of natural numbers, (cf. Theorems 3.2, 3.4, and 3.5 in [31], respectively). (Certainly, in the semiring Trop the upper bound of the accepting costs is not allowed to be $\infty$.) Moreover, it was shown that, for every tree automaton with costs over $\mathrm{FSet}(\mathbb{N})$, which is the semiring of finite subsets of natural numbers, it is decidable whether the set of the cardinalities of accepting costs (which are also finite sets of natural numbers) is finite (cf. Theorem 3.19 in [31]). Note that in $\mathrm{FSet}(\mathbb{N})$ the operation $\oplus$ is the union and the operation $\odot$ is the pointwise addition of sets of non-negative integers. Let us observe that cost-finiteness (and hence boundedness with respect to $\subseteq$) of a tree automaton $M$ with costs over $\mathrm{FSet}(\mathbb{N})$ implies that $M$ has the mentioned decidable property but the converse implication fails.

It should also be noted that, for the particular semirings Nat, Arct, and Trop, boundedness with respect to $\leq$ is equivalent with cost-finiteness, while for $\mathrm{FSet}(\mathbb{N})$

boundedness with respect to $\subseteq$ is also equivalent with cost-finiteness.

In this research our aim was to give a reasonable class of semirings such that the cost-finiteness of tree automata with costs over a semiring of this class is decidable. We found this class to be the class of finitely factorizing and monotonic semirings. As the main result of this paper, we proved that the cost-finiteness of tree automata with costs over finitely factorizing and monotonic semirings is decidable (cf. Theorem 46). Next we discuss this result in more detail.

First we define finitely factorizing as well as monotonic semirings. A semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ is finitely factorizing if every element $a \in A$ can be decomposed as the $\oplus$ of two further elements in finitely many ways and as the $\odot$ of two non-zero elements also in finitely many ways. For instance, the semirings Nat, Arct, and FSet($\mathbb{N}$) are finitely factorizing, while Trop is not. Moreover, $\mathcal{A}$ is monotonic if it is infinite and there is a partial order $\preceq$ on $A$ such that, roughly speaking, both $\oplus$ and $\odot$ are monotonic with respect to $\preceq$ in the sense that, for every elements $a_1, a_2 \in A$, we have $a_1 \preceq a_1 \oplus a_2$ and if $a_1 \neq \mathbf{0} \neq a_2$ and $a_2 \neq \mathbf{1}$, then also $a_1 \prec a_1 \odot a_2$ and $a_1 \prec a_2 \odot a_1$. For instance, Nat and Arct are monotonic with respect to $\leq$ while FSet($\mathbb{N}$) is not monotonic with respect to $\subseteq$ (but it is monotonic with respect to another partial order as we will see).

In the rest of the discussion we will mainly be interested in the set of $(q,q)$-computations in which the variable $x_1$ occurs exactly once, denoted by $\widehat{\Psi}^q_q$, because such a $(q,q)$-computation, as part of another computation, can be pumped.

In order to decide cost-finiteness of a tree automaton with costs we have to transform it into a special form, which is called reduced. Roughly speaking, a tree automaton $M$ with cost function $c : \delta \longrightarrow P(A,X)$ is reduced, whenever for every state $q \in Q$ the facts that $\widehat{\Psi}^q_q \neq \emptyset$ and that the cost of at least one $q$-computation is not $\mathbf{0}$ or $\mathbf{1}$ imply that there is a computation $\psi \in \widehat{\Psi}^q_q$ such that the variable $x_1$ occurs also in the cost $c(\psi)$ of $\psi$, i.e., the polynomial $c(\psi)$ as a function depends on its argument. The existence of such a computation $\psi$ is very important because, if $\psi$ is part of another computation and we pump it, then the cost of the pumped computation may grow provided the underlying semiring is finitely factorizing and monotonic. We have proved that, for every tree automaton $M$ with cost function $c : \delta \longrightarrow P(A,X)$ over a monotonic semiring $\mathcal{A}$ (in fact an even weaker assumption on $\mathcal{A}$ is sufficient), a reduced tree automaton $M'$ with cost function $c' : \delta' \longrightarrow P(A,X)$ can be constructed such that $M$ and $M'$ are cost-equivalent, i.e., $c(M) = c(M')$ (cf. Lemma 27).

Further, we have given different characterizations of cost-finiteness of a reduced tree automaton $M$ with costs (cf. Theorem 44). The first two of them, which are denoted by (ii) and (iii) in Theorem 44, are necessary only for technical reasons in order to be able to handle the third one, which is denoted by (iv). The characterization (iv), called Condition (linear-trans) in Definition 40, refers to the costs of the transitions of $M$, i.e., to finitely many polynomials only. Using this fact, we could prove that the Condition (linear-trans) is decidable for tree automata with costs over finitely factorizing and monotonic semirings (cf. Lemma 45). So we have obtained that, for every tree automaton $M$ with cost function $c : \delta \longrightarrow P(A,X)$ over a finitely factorizing and monotonic semiring, it is decidable whether $M$ is cost-finite in the following way (cf. Theorem 46). We construct a reduced tree automaton $M'$ which is cost-equivalent with $M$. Hence $M$ is cost-finite if and only if $M'$ is cost-finite. Then we decide whether $M'$ is cost-finite by checking if it satisfies Condition (linear-trans).

Finally, we have considered the connection between cost-finiteness and boundedness of tree automata with costs. It has turned out that the connection is very close provided that the underlying semiring is finitely factorizing and naturally ordered and boundedness is meant with respect to the natural order. The semiring $\mathcal{A}$ is naturally ordered provided that the relation $\sqsubseteq$ on $A$, defined by $a \sqsubseteq b$ if and only if there is a $c \in A$ such that $a \oplus c = b$, is a partial order. For example, the semirings Nat, Arct, Trop, and FSet($\mathbb{N}$) are naturally ordered. Now it is easy to show that, for a tree automaton $M$ with costs over a finitely factorizing and naturally ordered semiring $\mathcal{A}$, the automaton $M$ is cost-finite if and only if it is bounded with respect to the natural order $\sqsubseteq$ (cf. Lemma 49). Thus, we have obtained that, for every tree automaton $M$ with costs over a finitely factorizing, monotonic, and naturally ordered semiring $\mathcal{A}$, it is decidable whether $M$ is bounded with respect to the natural order $\sqsubseteq$ (cf. Theorem 50) because it is sufficient to decide whether $M$ is cost-finite.

Let us note that if a semiring $\mathcal{A}$ is monotonic with respect to $\preceq$, and at the same time, is naturally ordered with respect to $\sqsubseteq$, then the partial orders $\preceq$ and $\sqsubseteq$ may coincide but also differ. For instance, the semirings Nat and Arct are both monotonic and naturally ordered with respect to the order $\leq$, however FSet($\mathbb{N}$) is naturally ordered but not monotonic with respect to $\subseteq$. Luckily, it is monotonic with respect to another partial order (cf. Corollary 56), thus Theorem 50 can also be applied to decide whether a tree automaton with costs over FSet($\mathbb{N}$) is bounded with respect to $\subseteq$.

Due to the above equivalence of cost-finiteness and boundedness with respect to the natural order, we have reobtained two results of [31] about boundedness of tree automata with costs as corollaries of our results. In fact, decidability of boundedness of a tree automaton over Nat with respect to $\leq$ (cf. Theorem 3.2 of [31]) and over Arct with respect to $\leq$ (cf. Theorem 3.4 of [31]) follows from our Theorem 50, because both Nat and Arct are finitely factorizing and monotonic (with respect to $\leq$) and naturally ordered with respect to $\leq$ (cf. Corollary 51 and Corollary 52). Moreover, decidability of boundedness of a tree automaton with costs over FSet($\mathbb{N}$) with respect to $\subseteq$ also follows from Theorem 50, because FSet($\mathbb{N}$) is finitely factorizing, monotonic, and naturally ordered with respect to $\subseteq$ (cf. Corollary 56). Let us recall that the last decidability result is not the same as the one which was shown in Theorem 3.19 of [31].

On the other hand, our decidability result (cf. Theorem 50) cannot be applied to reobtain Theorem 3.5 of [31], i.e., the decidability of the boundedness of tree automata with costs over Trop, because the semiring Trop is neither finitely factorizing nor monotonic. However, our results can be applied to further important semirings, for instance, the semiring $(\mathbb{N}, \mathrm{lcm}, \cdot, 0, 1)$, where lcm is the usual least commom multiple, and also the square matrix semiring $(\mathbb{N}_+^{n\times n} \cup \{\underline{0}, \underline{1}\}, +, \cdot, \underline{0}, \underline{1})$ over the positive integers $\mathbb{N}_+$ as evidenced in Corollary 53 and Corollary 54.

Now we describe the structure of our paper. In Section 2 we introduce some basic concepts about mappings and relations, trees, monoids and semirings, and polynomials over semirings, which will be used as costs of transitions of tree automata. In Section 3 we define monotonic semirings, consider polynomials over finitely factorizing and monotonic semirings and prove those decidability results for them which will be necessary to show the decidability of cost-finiteness of tree automata with costs over semirings having these two properties. In Section 4, we introduce the concept of a tree automaton with costs over a semiring. We define reduced tree automata with costs and show that, for every tree automaton with costs over a monotonic semiring, a cost-equivalent one over the same semiring can be constructed. In Section 5 we give characterizations of cost-finiteness of reduced tree automata over finitely factorizing and monotonic semirings and prove our main decidability result. In Section 6 we show that cost-finiteness and boundedness with respect to the natural order are equivalent for tree automata with costs over finitely factorizing and naturally ordered semirings. Moreover, we show that two results of [31] follow as corollaries from our results.

## 2. Preliminaries

In this section we present the basic notions and notations required in the sequel. We assume that the reader is familiar with elementary set and number theory [12]. The remaining concepts will be defined formally in this and later sections.

The first subsection recalls some basic notations regarding sets, mappings, and relations [29]. In particular, we define equivalence relations and partial orders. The former will be central in our treatment of polynomials in Subsection 2.4, and the latter will be an essential component in the definition of monotonic semirings in Section 3. Finally, we also present the principle of well-founded induction, which we will apply in Section 5.

The next subsection deals exclusively with trees [9, 10] and operations defined thereupon. Subsection 2.3 is devoted to algebraic structures [20, 21] and semirings [24, 19, 11], in particular. The final subsection of this section introduces the core notion of polynomials. We deliberately present a (rather) non-standard definition of polynomials, primarily because this enables us to consider polynomials in arbitrary (not necessarily commutative) semirings.

### *2.1. Sets, Mappings, and Relations*

We denote by $\mathbb{N}$ the set $\{0, 1, 2, \ldots\}$ of non-negative integers and we let $\mathbb{N}_+ = \{1, 2, \ldots\}$ be the set of positive integers. For every $k, n \in \mathbb{N}$ we let $[k, n] = \{\, i \in \mathbb{N} \mid k \leq i \leq n \,\}$ and $[n] = [1, n]$. We observe that $[0] = \emptyset$. Given a finite set $S$ the *cardinality* of $S$, i.e., the number of elements of $S$, is denoted by $\mathrm{card}(S)$; thus $\mathrm{card}([n]) = n$. The set of all subsets of a set $S$, also called the *power set* of $S$, is denoted by $\mathcal{P}(S) = \{\, S' \mid S' \subseteq S \,\}$ and the set of all finite subsets is denoted by $\mathcal{P}_{\mathrm{f}}(S) = \{\, S' \subseteq S \mid S' \text{ is finite} \,\}$. A (total) mapping $f$ from a non-empty set $S_1$ to a non-empty set $S_2$ is denoted by $f : S_1 \longrightarrow S_2$. We occasionally lift mappings from sets to the corresponding power sets, i.e., given a mapping $f : S_1 \longrightarrow S_2$ we define $\hat{f} : \mathcal{P}(S_1) \longrightarrow \mathcal{P}(S_2)$ by $\hat{f}(S_1') = \{\, f(s_1') \mid s_1' \in S_1' \,\}$ for every subset $S_1' \subseteq S_1$. If no confusion might arise, we drop the hat and just write $f$.

The *Cartesian product* of sets $S_1$ and $S_2$ is displayed as $S_1 \times S_2$, and we will shorten the Cartesian product $S \times \cdots \times S$ containing $n$-times the set $S$ simply to $S^n$ for every $n \in \mathbb{N}$. We remark that $S^0 = \{()\}$. A (binary) *relation* (on $S$) is a subset $\rho \subseteq S^2$. We generally write $s_1 \ \rho \ s_2$ for $(s_1, s_2) \in \rho$. Specifically, a relation $\sim \subseteq S^2$ is called *equivalence relation* (on $S$), if it is (i) *reflexive*, i.e., for every $s \in S$ we have $s \sim s$, (ii) *symmetric*, i.e., for every two elements $s_1, s_2 \in S$ if $s_1 \sim s_2$, then $s_2 \sim s_1$, and (iii) *transitive*, i.e., for every three elements $s_1, s_2, s_3 \in S$ the facts $s_1 \sim s_2$ and $s_2 \sim s_3$ imply $s_1 \sim s_3$. For every element $s \in S$ we define the *equivalence class* of $s$ by $[s]_\sim = \{ s' \in S \mid s \sim s' \}$. The *factor set* of $S' \subseteq S$ is defined to be the set $[S']_\sim = \{ [s']_\sim \mid s' \in S' \}$.

A *partial order* (on $S$) is defined to be a relation $\preceq \subseteq S^2$ which is (i) reflexive, (ii) transitive, and (iii) *anti-symmetric*, i.e., for every two $s_1, s_2 \in S$ if $s_1 \preceq s_2$ and $s_2 \preceq s_1$ then $s_1 = s_2$. As usual we write $s_1 \prec s_2$, whenever $s_1 \preceq s_2$ and $s_1 \neq s_2$. A subset $S' \subseteq S$ is *bounded* provided there exists an $s \in S$ such that $s' \preceq s$ for every $s' \in S'$. Moreover, an element $s' \in S'$ is called *minimal element* of $S'$, if $s \preceq s'$ implies $s = s'$ for every element $s \in S'$. In addition, $\preceq$ is termed *well-founded*, if every non-empty subset of $S$ has a minimal element. Clearly, if the set $S$ is finite, then $\preceq$ is well-founded. The next principle is called principle of *well-founded induction* (cf. Theorem 9 of [36]).

**Principle 1.** *Let $\preceq$ be a well-founded partial order on a set $S$. A property $\Pi \subseteq S$ holds for all elements of $S$, i.e., $\Pi = S$, if*

$$(\forall s_1 \in S) : \big((\forall s_2 \in S) : \ s_2 \prec s_1 \Rightarrow s_2 \in \Pi\big) \Rightarrow s_1 \in \Pi.$$

### *2.2. Trees*

The set of all (finite) *sequences* over a set $S$ is denoted by $S^* = \bigcup_{n \in \mathbb{N}} S^n$ with the empty sequence () denoted by $\varepsilon$. The *length* of the sequence $w \in S^*$, denoted by $|w|$, is defined to be the integer $n \in \mathbb{N}$ such that $w \in S^n$. We prefer to drop the tuple notation, and if delimitation is required to give an unambiguous meaning to a sequence, we use $\cdot$ to separate elements of $S$. For example, $1{\cdot}45{\cdot}5 \in \mathbb{N}^*$ denotes the sequence $(1, 45, 5)$. This notation itself is prone to ambiguity, but the intended meaning (delimitation or multiplication) should always be obvious from the context.

Sets which are non-empty and finite are also called *alphabets*, and the elements of alphabets are called *symbols*. A *ranked alphabet* is a pair $(\Sigma, \mathrm{rk}_\Sigma)$ consisting of an alphabet $\Sigma$ and a mapping $\mathrm{rk}_\Sigma : \Sigma \longrightarrow \mathbb{N}$ associating to every symbol of $\Sigma$ a *rank* (an arity). When specifying a ranked alphabet, we usually list the symbols of the alphabet $\Sigma$ with their corresponding ranks annotated in parentheses as superscripts, e.g., $\{\sigma^{(2)}, \alpha^{(0)}\}$ shall denote the ranked alphabet $(\Sigma, \mathrm{rk}_\Sigma)$ with $\Sigma = \{\sigma, \alpha\}$, $\mathrm{rk}_\Sigma(\sigma) = 2$, and $\mathrm{rk}_\Sigma(\alpha) = 0$. For every $k \in \mathbb{N}$ we denote by $\Sigma^{(k)}$ the set of all symbols of $\Sigma$ which have rank $k$, i.e., $\Sigma^{(k)} = \{ \sigma \in \Sigma \mid \mathrm{rk}_\Sigma(\sigma) = k \}$. In the sequel let $\Sigma$ be a ranked alphabet and $V$ be a set disjoint with $\Sigma$, i.e., $V \cap \Sigma = \emptyset$.

The set of (finite, labeled, and ordered) $\Sigma$-*trees* (indexed by $V$), denoted by $T_\Sigma(V)$, is the smallest subset $T \subseteq (\Sigma \cup V \cup \{(,)\} \cup \{,\})^*$ such that (i) $V \cup \Sigma^{(0)} \subseteq T$, and (ii) if $\sigma \in \Sigma^{(k)}$ with $k \in \mathbb{N}_+$ and $s_1, \ldots, s_k \in T$, then $\sigma(s_1, \ldots, s_k) \in T$. We write

$T_\Sigma(\emptyset)$ as $T_\Sigma$. It should be clear that $T_\Sigma = \emptyset$, if and only if $\Sigma^{(0)} = \emptyset$. Since we are not interested in this particular case, we assume that $\Sigma^{(0)} \neq \emptyset$ for every ranked alphabet $\Sigma$ appearing in the following.

We recursively define height : $T_\Sigma(V) \longrightarrow \mathbb{N}_+$ and pos : $T_\Sigma(V) \longrightarrow \mathcal{P}_\mathrm{f}((\mathbb{N}_+)^*)$ for every tree $s \in T_\Sigma(V)$ as follows. (i) If $s \in V \cup \Sigma^{(0)}$, then $\mathrm{height}(s) = 1$ and $\mathrm{pos}(s) = \{\varepsilon\}$. (ii) If $s = \sigma(s_1, \ldots, s_k)$ for some $k \in \mathbb{N}_+$, symbol $\sigma \in \Sigma^{(k)}$, and subtrees $s_1, \ldots, s_k \in T_\Sigma(V)$, then

$$\begin{aligned} \mathrm{height}(\sigma(s_1, \ldots, s_k)) &= 1 + \max\{\, \mathrm{height}(s_i) \mid i \in [k] \,\}, \\ \mathrm{pos}(\sigma(s_1, \ldots, s_k)) &= \{\varepsilon\} \cup \{\, i{\cdot}w_i \mid i \in [k], w_i \in \mathrm{pos}(s_i) \,\}. \end{aligned}$$

Each $w \in \mathrm{pos}(s)$ is called *position* of $s$ and $\mathrm{size}(s) = \mathrm{card}(\mathrm{pos}(s))$. In addition, we recursively define $\mathrm{lab}_s : \mathrm{pos}(s) \longrightarrow V \cup \Sigma$ returning the *label* at a position of $s$ and $s|_{(\cdot)} : \mathrm{pos}(s) \longrightarrow T_\Sigma(V)$ returning the *subtree* at a position of $s$. (i) If $s \in V \cup \Sigma^{(0)}$ (and thus $\mathrm{pos}(s) = \{\varepsilon\}$), then $\mathrm{lab}_s(\varepsilon) = s|_\varepsilon = s$. (ii) If $s = \sigma(s_1, \ldots, s_k)$ for some symbol $\sigma \in \Sigma^{(k)}$ with $k \in \mathbb{N}_+$ and subtrees $s_1, \ldots, s_k \in T_\Sigma(V)$, then

$$\mathrm{lab}_{\sigma(s_1,\ldots,s_k)}(w) = \begin{cases} \sigma & \text{, if } w = \varepsilon \\ \mathrm{lab}_{s_i}(w_i) & \text{, if } w = i{\cdot}w_i \text{ for some } i \in [k] \text{ and } w_i \in \mathrm{pos}(s_i) \end{cases},$$

$$\sigma(s_1, \ldots, s_k)|_w = \begin{cases} \sigma(s_1, \ldots, s_k) & \text{, if } w = \varepsilon \\ s_i|_{w_i} & \text{, if } w = i{\cdot}w_i \text{ for some } i \in [k] \text{ and } w_i \in \mathrm{pos}(s_i) \end{cases}.$$

For notational convenience we identify $z$ with $z()$ whenever $z \in V \cup \Sigma^{(0)}$. We observe that given a tree $s \in T_\Sigma(V)$, then there are a unique integer $k \in \mathbb{N}$ and a unique element $\sigma \in V \cup \Sigma^{(k)}$ such that $s = \sigma(s|_1, \ldots, s|_k)$. The number of *occurrences* of $z \in V \cup \Sigma$ in a tree $s \in T_\Sigma(V)$, denoted by $|s|_z$, is defined by $|s|_z = \mathrm{card}(\{\, w \in \mathrm{pos}(s) \mid \mathrm{lab}_s(w) = z \,\})$. Let $Y$ be a finite subset of $V$ and $s \in T_\Sigma(V)$. The tree $s$ is called *linear* in $Y$ (or likewise *non-deleting* in $Y$), if every $y \in Y$ occurs at most once, i.e., $|s|_y \leq 1$, (or likewise at least once, i.e., $1 \leq |s|_y$) in the tree $s$. Finally, the set $\widehat{T_\Sigma}(Y) \subseteq T_\Sigma(Y)$ is defined by $\widehat{T_\Sigma}(Y) = \{\, s \in T_\Sigma(Y) \mid s \text{ linear and non-deleting in } Y \,\}$.

Given a $\Sigma$-tree $s \in T_\Sigma(V)$, an integer $n \in \mathbb{N}$, (pairwise) distinct elements $v_1, \ldots, v_n \in V$, and $\Sigma$-trees $t_1, \ldots, t_n \in T_\Sigma(V)$ the *tree substitution* of $t_1, \ldots, t_n$ for $v_1, \ldots, v_n$ in $s$, denoted by $s[v_1 \leftarrow t_1, \ldots, v_n \leftarrow t_n]$, is recursively defined as follows. Let $\theta = [v_1 \leftarrow t_1, \ldots, v_n \leftarrow t_n]$. (i) If $s = v_i$ for some $i \in [n]$ then $v_i\,\theta = t_i$, and (ii) if $s = \sigma(s|_1, \ldots, s|_k)$ for some $k \in \mathbb{N}$ and $\sigma \in (V \setminus \{v_1, \ldots, v_n\}) \cup \Sigma^{(k)}$, then $\sigma(s|_1, \ldots, s|_k)\,\theta = \sigma(s|_1\,\theta, \ldots, s|_k\,\theta)$. Any subset of $T_\Sigma(V)$ is called *tree language*. Next we define the OI-substitution of tree languages (cf. [6, 7]). Let $L_1, \ldots, L_n \subseteq T_\Sigma(V)$ be tree languages for some $n \in \mathbb{N}$ and $\theta' = [v_1 \leftarrow L_1, \ldots, v_n \leftarrow L_n]$. The *OI-substitution* of $L_1, \ldots, L_n$ for $v_1, \ldots, v_n$ in $s$ is defined to be the set $s\,\theta' \subseteq T_\Sigma(V)$, which is recursively defined by (i) if $s = v_i$ for some $i \in [n]$, then $v_i\,\theta' = L_i$, and (ii) if $s = \sigma(s|_1, \ldots, s|_k)$ for some $k \in \mathbb{N}$ and $\sigma \in (V \setminus \{v_1, \ldots, v_n\}) \cup \Sigma^{(k)}$, then

$$\sigma(s|_1, \ldots, s|_k)\,\theta' = \{\, \sigma(s_1, \ldots, s_k) \mid (\forall i \in [k]) : \ s_i \in (s|_i)\,\theta' \,\}.$$

Moreover, $L\,\theta' = \bigcup_{s \in L} s\,\theta'$ for $L \subseteq T_\Sigma(V)$ and $L\,\theta = L[v_1 \leftarrow \{t_1\}, \ldots, v_n \leftarrow \{t_n\}]$.

Let $X = \{\, x_i \mid i \in \mathbb{N}_+ \,\}$ be a fixed set of (formal) *variables* disjoint with $\Sigma$. For every integer $k \in \mathbb{N}$ we define $X_k = \{\, x_i \mid i \in [k] \,\}$; thus $X_0 = \emptyset$. We use these variables to occur in trees, so we will frequently consider sets like $T_\Sigma(X)$ and $T_\Sigma(X_k)$. In particular, we abbreviate the substitution $s[x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n]$ by just $s[t_1, \ldots, t_n]$ for every $n \in \mathbb{N}_+$ and $s \in T_\Sigma(X)$ and likewise we also use the abbreviations $s[L_1, \ldots, L_n]$, $L[L_1, \ldots, L_n]$, and $L[t_1, \ldots, t_n]$. Finally, for every tree $s \in T_\Sigma(X_1)$ and integer $n \in \mathbb{N}$ the $n$-th power $s^n$ is recursively defined by $s^0 = x_1$ and $s^{n+1} = s[s^n]$.

### *2.3. Monoids and Semirings*

A *monoid* is an algebraic structure $\mathcal{A} = (A, \otimes, \mathbf{1})$ with *carrier set* $A$, operation $\otimes : A^2 \longrightarrow A$, and *unit element* $\mathbf{1} \in A$ satisfying the axioms of *associativity*, i.e., for every three elements $a_1, a_2, a_3 \in A$ the equality $(a_1 \otimes a_2) \otimes a_3 = a_1 \otimes (a_2 \otimes a_3)$ holds, and *unit*, i.e., for every element $a \in A$ we have $\mathbf{1} \otimes a = a = a \otimes \mathbf{1}$. For every element $a \in A$ and integer $n \in \mathbb{N}$ we adopt the *power notation* $a^n$ abbreviating the $n$-fold product $a \otimes \cdots \otimes a$ of $a$ with itself and we put $a^0 = \mathbf{1}$. A monoid $\mathcal{A} = (A, \otimes, \mathbf{1})$ is called *commutative*, if for every two elements $a_1, a_2 \in A$ the equality $a_1 \otimes a_2 = a_2 \otimes a_1$ is satisfied. The set $D^{\otimes}(a) \subseteq A^2$ of decompositions of $a$ is defined as $D^{\otimes}(a) = \{\, (a_1, a_2) \in A^2 \mid a = a_1 \otimes a_2 \,\}$. Finally, we say that $\mathcal{A}$ is *finitely factorizing*, if for every element $a \in A$ the set $D^{\otimes}(a)$ is finite.

A *semiring* $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ (with one and absorbing zero) is an algebraic structure with *carrier set* $A$, operations $\oplus, \odot : A^2 \longrightarrow A$, often called *addition* and *multiplication*, respectively, and *unit elements* $\mathbf{0}, \mathbf{1} \in A$ such that $(A, \oplus, \mathbf{0})$ is a commutative monoid and $(A, \odot, \mathbf{1})$ is a monoid. Additionally, the axioms of *distributivity*, i.e., for every three elements $a_1, a_2, a_3 \in A$, we have $a_1 \odot (a_2 \oplus a_3) = (a_1 \odot a_2) \oplus (a_1 \odot a_3)$ as well as $(a_1 \oplus a_2) \odot a_3 = (a_1 \odot a_3) \oplus (a_2 \odot a_3)$, and *absorption*, i.e., for every element $a \in A$, the equality $a \odot \mathbf{0} = \mathbf{0} = \mathbf{0} \odot a$ holds, need to be fulfilled.

As usual we assume that the priority of multiplication is greater than the priority of addition; thus we read $a_1 \oplus a_2 \odot a_3$ as $a_1 \oplus (a_2 \odot a_3)$. The operations are also lifted to sets in the usual manner, i.e., for every $A_1, A_2 \subseteq A$ and $\otimes \in \{\oplus, \odot\}$ we let $A_1 \otimes A_2 = \{\, a_1 \otimes a_2 \mid a_1 \in A_1, a_2 \in A_2 \,\}$. Likewise we use $a \otimes A_2$ and $A_1 \otimes a$ as abbreviations for $\{a\} \otimes A_2$ and $A_1 \otimes \{a\}$, respectively. Moreover, we denote $A' \setminus \{\mathbf{0}\}$ shortly by $A'_+$ for every $A' \subseteq A$ and we use the power notation exclusively for the multiplication. The semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ has the property of being

- *idempotent*, if $\mathbf{1} \oplus \mathbf{1} = \mathbf{1}$,
- *naturally ordered*, if the relation $\sqsubseteq \; \subseteq A^2$, defined by $a_1 \sqsubseteq a_2$ if and only if $a_2 \in a_1 \oplus A$, is a partial order on $A$,
- *positive*, if $\mathcal{A}$ is *zero-sum free*, i.e., $A \oplus A_+ \subseteq A_+$, and *zero-divisor free*, i.e., $A_+ \odot A_+ \subseteq A_+$,
- *one-summand free*, if $a_1 \oplus a_2 = \mathbf{1}$ implies $a_1, a_2 \in \{\mathbf{0}, \mathbf{1}\}$,
- *one-product free*, if $a_1 \odot a_2 = \mathbf{1}$ implies $a_1 = \mathbf{1} = a_2$,
- *finitely factorizing*, if the set $D^{\oplus}(a)$ and the set

$$D^{\odot}_{+}(a) = \{\, (a_1, a_2) \in A^2_+ \mid a = a_1 \odot a_2 \,\}$$

of multiplicative decompositions are finite for every $a \in A$, and

- *computable* if the operations $\oplus$ and $\odot$ are computable functions, i.e., there exists a Turing machine which for every $a_1, a_2 \in A$ runs on an initial tape containing a suitable coding of $a_1$ and $a_2$ and halts with the result $a_1 \oplus a_2$ (respectively, $a_1 \odot a_2$) on the tape.

Since every semiring obeying $\mathbf{0} = \mathbf{1}$ has a singleton carrier set, we will generally assume that $\mathbf{0} \neq \mathbf{1}$ for all semirings considered in this paper. The following semirings shall illustrate the notion of a semiring and the properties defined thereupon. In the sequel we will refer to these semirings occasionally in examples. Table 1 attempts to display the properties of those semirings, where we assume that the alphabet $\Sigma$ is non-trivial, i.e., $1 < \mathrm{card}(\Sigma)$.

- The *Boolean semiring* $\mathrm{Bool} = (\{0,1\}, \vee, \wedge, 0, 1)$ where $\vee$ is disjunction and $\wedge$ is conjunction.
- The *semiring of the non-negative integers* $\mathrm{Nat} = (\mathbb{N}, +, \cdot, 0, 1)$ with the usual operations of addition and multiplication.
- The *arctic semiring* $\mathrm{Arct} = (\mathbb{N} \cup \{-\infty\}, \max, +, (-\infty), 0)$ with the standard maximum operation extended such that $(-\infty)$ behaves like a neutral element and $+$ extended to an absorbing element $(-\infty)$.
- The *tropical semiring* $\mathrm{Trop} = (\mathbb{N} \cup \{+\infty\}, \min, +, (+\infty), 0)$ similar to the semiring Arct, but with the common minimum operation.
- The semiring $\mathrm{Lcm} = (\mathbb{N}, \mathrm{lcm}, \cdot, 0, 1)$ with $\mathrm{lcm}(0, n) = n = \mathrm{lcm}(n, 0)$ for every $n \in \mathbb{N}$ and otherwise lcm is the usual least common multiple.
- The *finite-language semiring* $\mathrm{FLang}(\Sigma) = (\mathcal{P}_{\mathrm{f}}(\Sigma^*), \cup, \circ, \emptyset, \{\varepsilon\})$ over the alphabet $\Sigma$ with the operations of union and concatenation.
- The *finite subsets semiring of* [31] $\mathrm{FSet}(\mathbb{N}) = (\mathcal{P}_{\mathrm{f}}(\mathbb{N}), \cup, +, \emptyset, \{0\})$ with the operations of union and addition, where the addition is extended to sets as usual.
- For every $n \in \mathbb{N}_+$, the semiring $\mathrm{Mat}_n(\mathbb{N}_+) = (\mathbb{N}_+^{n \times n} \cup \{\underline{0}, \underline{1}\}, +, \cdot, \underline{0}, \underline{1})$ of *square matrices over* $\mathbb{N}_+$ with the common matrix addition and multiplication, where $\underline{0}$ is the $n \times n$ zero matrix and $\underline{1}$ is the $n \times n$ unit matrix.

In the following observation we state an important property of finitely factorizing semirings, namely that they preserve infinite subsets. Obviously multiplying an infinite set with the singleton set $\{\mathbf{0}\}$ creates an exception, but the next observation shows that this is in fact the only exception.

**Observation 2.** *Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be a finitely factorizing semiring and $\otimes \in \{\oplus, \odot\}$ be an operation of the semiring. Moreover, let $B = A$ if $\otimes = \oplus$ otherwise $B = A_+$. Finally, let $A_1, A_2 \subseteq A$ be such that $A_1 \cap B \neq \emptyset \neq B \cap A_2$.*

*(i) If $A_1$ is infinite, then for every $a_2 \in B$ the set $A_1 \otimes a_2$ is infinite.*

*(ii) If $A_2$ is infinite, then for every $a_1 \in B$ the set $a_1 \otimes A_2$ is infinite.*

*(iii) If $A_1$ or $A_2$ is infinite, then the set $A_1 \otimes A_2$ is infinite.*

| | idempotent | naturally ordered | positive | one-summand free | one-product free | finitely factorizing |
|---|---|---|---|---|---|---|
| Bool | yes | yes | yes | yes | yes | yes |
| Nat | NO | yes | yes | yes | yes | yes |
| Arct | yes | yes | yes | yes | yes | yes |
| Trop | yes | yes | yes | NO | yes | NO |
| Lcm | yes | yes | yes | yes | yes | yes |
| FLang($\Sigma$) | yes | yes | yes | yes | yes | yes |
| FSet($\mathbb{N}$) | yes | yes | yes | yes | yes | yes |
| $\mathrm{Mat}_n(\mathbb{N}_+)$ | NO | yes | yes | yes | yes | yes |

Table 1: Properties of some semirings.

*Proof.* We first prove Statement (i); Statement (ii) can then be proved similarly. In order to derive a contradiction, let us assume that there is a semiring element $a_2 \in B$ such that $A_1 \otimes a_2$ is finite. By the pigeon-hole principle, there is an element $a' \in A_1 \otimes a_2$ such that for infinitely many $a_1 \in (A_1)_+$ the equality $a' = a_1 \otimes a_2$ holds. However, this is a contradiction, because $\mathcal{A}$ is finitely factorizing.

Lastly we prove Statement (iii). Therefore assume that $A_1$ is infinite. Then the statement follows from Statement (i), because $B \cap A_2$ is non-empty and for every $a_2 \in A_2$ we have that $A_1 \otimes a_2 \subseteq A_1 \otimes A_2$. The case in which $A_2$ is infinite can be handled alike using Statement (ii). □

Trivially, any finite semiring is finitely factorizing, but the next observation derives that every infinite, but finitely factorizing semiring is necessarily positive.

**Observation 3.** *Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be an infinite, but finitely factorizing semiring. Then $\mathcal{A}$ is positive.*

*Proof.* Firstly we prove zero-sum freeness, i.e., $A \oplus A_+ \subseteq A_+$. In order to derive a contradiction, let $a_1 \oplus a_2 = \mathbf{0}$ for some $a_1 \in A$ and $a_2 \in A_+$. Consequently, $a \odot (a_1 \oplus a_2) = a \odot a_1 \oplus a \odot a_2 = \mathbf{0}$ for every $a \in A$. Since $A$ is infinite, also $A \odot a_1$ is infinite by Observation 2(i). We observe that $\{ (a \odot a_1, a \odot a_2) \mid a \in A \} \subseteq D^{\oplus}(\mathbf{0})$, thus $D^{\oplus}(\mathbf{0})$ is infinite, because $A \odot a_1$ is infinite. However, this contradicts the finitely factorizing property, hence $\mathcal{A}$ is zero-sum free.

Finally, we need to prove zero-divisor freeness, i.e., $A_+ \odot A_+ \subseteq A_+$. Let $a_1 \odot a_2 = \mathbf{0}$ for some $a_1, a_2 \in A_+$. Then also $a \odot a_1 \odot a_2 = \mathbf{0}$ for every $a \in A$. Clearly, $A \odot a_1$ is infinite by Observation 2(i), because $A$ is infinite and $a_1 \in A_+$. Moreover, $\{ (a \odot a_1, a_2) \mid a \in A \} \subseteq D^{\odot}_{+}(\mathbf{0}) \cup \{(\mathbf{0}, a_2)\}$. Consequently, $D^{\odot}_{+}(\mathbf{0})$ is also infinite, which constitutes a contradiction to the finitely factorizing property.

Summing up, we have that $\mathcal{A}$ is zero-sum free and zero-divisor free, hence $\mathcal{A}$ is positive. □

*2.4. Polynomials over Semirings*

Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be a computable semiring, of which the carrier set $A$ is disjoint with $X \cup \{+, \cdot\}$. This assumption remains valid for the rest of the paper. Moreover, let $n \in \mathbb{N}$. A *monomial* (over $X_n$ with coefficients of $A$) is defined to be an element of $T_{\{\cdot^{(2)}\}}(A \cup X_n)$. To avoid cumbersome notation, we abbreviate the set $T_{\{\cdot^{(2)}\}}(A \cup X_n)$ of all monomials by $A[X_n]$. Note that $X_n \subseteq A[X_n]$ as well as $A \subseteq A[X_n]$. A *polynomial* (over $X_n$ with coefficients of $A$) is an element of the set $T_{\{\cdot^{(2)}, +^{(2)}\}}(A \cup X_n)$. In the sequel, we will also use $P(A, X_n)$ to denote the set $T_{\{\cdot^{(2)}, +^{(2)}\}}(A \cup X_n)$ of polynomials. Apparently, $A[X_n] \subseteq P(A, X_n)$. By convention we usually write the binary symbols infix.

Since polynomials are trees, we can substitute polynomials into polynomials. We note that the substitution of monomials into a monomial again yields a monomial. Let $p_1, p_2 \in P(A, X_n)$ be polynomials. The addition of $p_1$ and $p_2$ is defined to be the polynomial $p = p_1 + p_2$ and the multiplication of $p_1$ and $p_2$ is likewise defined to be the polynomial $p = p_1 \cdot p_2$. Thereby the multiplication of two monomials gives a monomial.

Next we connect the syntactic entity of a polynomial with an operation on the semiring $\mathcal{A}$. Therefore, let $p \in P(A, X_n)$ be a polynomial. The ($n$-ary) *polynomial function* induced by $p$ is the mapping $\overline{p} : A^n \longrightarrow A$ recursively defined for every $n$ semiring elements $a_1, \ldots, a_n \in A$ and every $p_1, p_2 \in P(A, X_n)$ by

(i) $\overline{a}(a_1, \ldots, a_n) = a$ for every coefficient $a \in A$,

(ii) $\overline{x_j}(a_1, \ldots, a_n) = a_j$ for every index $j \in [n]$,

(iii) $\overline{p_1 + p_2}(a_1, \ldots, a_n) = \overline{p_1}(a_1, \ldots, a_n) \oplus \overline{p_2}(a_1, \ldots, a_n)$, and

(iv) $\overline{p_1 \cdot p_2}(a_1, \ldots, a_n) = \overline{p_1}(a_1, \ldots, a_n) \odot \overline{p_2}(a_1, \ldots, a_n)$.

Finally, two polynomials $p_1, p_2 \in P(A, X_n)$ are said to be *equivalent*, denoted by $p_1 \equiv p_2$, if their induced polynomial functions coincide, i.e., $\overline{p_1} = \overline{p_2}$. Moreover, given two sets of polynomials $P_1, P_2 \subseteq P(A, X_n)$ we write $P_1 \equiv P_2$ to denote $[P_1]_\equiv = [P_2]_\equiv$. The polynomial function induced by $p$ is lifted to a mapping on sets in the usual way. We define the mapping $\overline{p} : \mathcal{P}(A)^n \longrightarrow \mathcal{P}(A)$ for every $n$ subsets $A_1, \ldots, A_n \subseteq A$ by $\overline{p}(A_1, \ldots, A_n) = \{\, \overline{p}(a_1, \ldots, a_n) \mid (\forall i \in [n]) : a_i \in A_i \,\}$. Note that this corresponds to IO-substitution [6, 7].

In the following we will drop the overlining from the notation $\overline{p}$ because it can be deduced from the context whether the polynomial $p$ or the polynomial function $\overline{p}$ is referred to. Next we show that the introduced equivalence is stable under substitutions (cf. ,e.g., [8]).

**Theorem 4.** *Let $p', p'', p_1, p_2 \in P(A, X_n)$ be polynomials such that $p' \equiv p''$ and $p_1 \equiv p_2$. Then for every variable $x \in X_n$ also $p'[x \leftarrow p_1] \equiv p''[x \leftarrow p_2]$.*

*Proof.* Clearly, we have to show that

$$(p'[x \leftarrow p_1])(a_1, \ldots, a_n) = (p''[x \leftarrow p_2])(a_1, \ldots, a_n)$$

for every $n$ semiring elements $a_1, \ldots, a_n \in A$. Let $x = x_i$ with $i \in [n]$. We observe that

$$\begin{aligned}(p'[x \leftarrow p_1])(a_1, \ldots, a_n) &= p'(a_1, \ldots, a_{i-1}, p_1(a_1, \ldots, a_n), a_{i+1}, \ldots, a_n) \\ &= p''(a_1, \ldots, a_{i-1}, p_2(a_1, \ldots, a_n), a_{i+1}, \ldots, a_n) \\ &= (p''[x \leftarrow p_2])(a_1, \ldots, a_n),\end{aligned}$$

where we used $p' \equiv p''$ and $p_1 \equiv p_2$ in the second line. The proofs of the equalities in line 1 and line 3 are straightforward using induction over the structure of $p'$ and $p''$, respectively, so we leave these proofs to the reader. $\square$

Since $(p_1 + p_2) + p_3 \equiv p_1 + (p_2 + p_3)$ and $(p_1 \cdot p_2) \cdot p_3 \equiv p_1 \cdot (p_2 \cdot p_3)$ due to the associativity of $\oplus$ and $\odot$, we usually just write $p_1 + p_2 + p_3$ and $p_1 \cdot p_2 \cdot p_3$ for every three polynomials $p_1, p_2, p_3 \in P(A, X_n)$. Moreover, we assume that $\cdot$ has a stronger binding priority than $+$. Finally, we omit the symbol $\cdot$ altogether whenever convenient. A polynomial of the form $m_1 + \cdots + m_k$ for some integer $k \in \mathbb{N}_+$ and monomials $m_1, \ldots, m_k \in A[X_n]$ is said to be in *normal form*. The next lemma shows that every polynomial admits an equivalent polynomial in normal form.

**Lemma 5.** *For every polynomial $p \in P(A, X_n)$ a polynomial $p' \in P(A, X_n)$ can effectively be constructed such that $p'$ is in normal form and $p \equiv p'$.*

*Proof.* Set $i = 0$ and $p_0 = p$.

**Iteration:** If $p_i$ is in normal form, then let $p' = p_i$, and halt. Otherwise, there exists a position $w \in \mathrm{pos}(p_i)$ such that $p_i|_w = p'_1 \cdot (p'_2 + p'_3)$ (or $p_i|_w = (p'_1 + p'_2) \cdot p'_3$) for some polynomials $p'_1, p'_2, p'_3 \in P(A, X_n)$. Then we obtain $p_{i+1}$ by replacing the subtree $p_i|_w$ in $p_i$ by the tree $(p'_1 \cdot p'_2) + (p'_1 \cdot p'_3)$ (or by the tree $(p'_1 \cdot p'_3) + (p'_2 \cdot p'_3)$). Next set $i := i + 1$ and continue the iteration.

We leave the proof of termination to the reader. If the iteration terminates, then $p \equiv p'$ because for every $i$ we have that $p_i \equiv p_{i+1}$. In fact, by distributivity $p'_1 \cdot (p'_2 + p'_3) \equiv (p'_1 \cdot p'_2) + (p'_1 \cdot p'_3)$ and $(p'_1 + p'_2) \cdot p'_3 \equiv (p'_1 \cdot p'_3) + (p'_2 \cdot p'_3)$, hence by the Replacement Theorem (cf. Theorem 4) we have $p_i \equiv p_{i+1}$. $\square$

A polynomial $p \in P(A_+, X_n)$ is called *zero-free*. Note that a zero-free polynomial $p$ might be equivalent to $\mathbf{0}$. However, if the semiring $\mathcal{A}$ is positive, then for every zero-free polynomial $p$ over $\mathcal{A}$, we have $p \not\equiv \mathbf{0}$.

**Observation 6.** *Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be a positive semiring, $p \in P(A_+, X_n)$ be a zero-free polynomial for some integer $n \in \mathbb{N}$, and $a_1, \ldots, a_n \in A_+$. Then $p(a_1, \ldots, a_n) \neq \mathbf{0}$.*

*Proof.* The statement follows easily from the facts $A_+ \oplus A_+ \subseteq A_+$ and $A_+ \odot A_+ \subseteq A_+$. We leave the details to the reader. $\square$

**Lemma 7.** *There is an algorithm which for every polynomial $p \in P(A, X_n)$ returns either $p \equiv \mathbf{0}$ or a zero-free polynomial $p' \in P(A_+, X_n)$ in normal form such that $p \equiv p'$.*

*Proof.* By Lemma 5 we can assume that an equivalent polynomial in normal form can be computed, i.e., $p \equiv m_1 + \ldots + m_k$ for some integer $k \in \mathbb{N}_+$ and monomials $m_1, \ldots, m_k \in A[X_n]$. If $1 \leq |m_i|_{\mathbf{0}}$ for every index $i \in [k]$, then obviously we can return $p \equiv \mathbf{0}$. Otherwise we return $p' = \sum_{i \in [k], |m_i|_{\mathbf{0}} = 0} m_i$, where the sum is only syntactic and the order of the monomials is obviously irrelevant due to the commutativity of the semiring addition. Clearly, $p'$ is zero-free and $p \equiv p'$. □

Now we consider the variables and the degree of a polynomial $p \in P(A, X_n)$. Therefore, we define the mappings

$$\mathrm{var}\colon\ P(A, X_n) \longrightarrow \mathcal{P}(X_n) \quad \text{and} \quad \deg\colon\ P(A, X_n) \longrightarrow \mathbb{N} \cup \{-\infty\}$$

by $\mathrm{var}(p) = \{\, x \in X_n \mid 1 \leq |p|_x \,\}$ and recursively for every coefficient $a \in A_+$, variable $x \in X_n$, and two polynomials $p_1, p_2 \in P(A, X_n)$ by

$$\begin{aligned} &\deg(\mathbf{0}) = -\infty \qquad &&\deg(a) = 0 \qquad &&\deg(p_1 + p_2) = \max(\deg(p_1), \deg(p_2)) \\ & &&\deg(x) = 1 \qquad &&\deg(p_1 \cdot p_2) = \deg(p_1) + \deg(p_2). \end{aligned}$$

The value $\deg(p)$ is called the (syntactic) *degree* of $p$. The *semantic degree* of $p$ is defined to be $\deg_{\mathrm{s}}(p) = \min\{\, \deg(p') \mid p' \in [p]_{\equiv} \,\}$. Note that $p \equiv \mathbf{0}$, if and only if $\deg_{\mathrm{s}}(p) = -\infty$. By a *linear polynomial* we mean a polynomial having semantic degree at most 1.

**Observation 8.** *Let $x, y \in X_n$ with $x \neq y$ be variables, $a \in A$ be a semiring element, and $p \in P(A, X_n)$ be a polynomial.*

$$x \in \mathrm{var}(p[y \leftarrow a]) \qquad \Longleftrightarrow \qquad x \in \mathrm{var}(p)$$

The next lemma will be central in the first part of Section 5. Roughly speaking, it states that non-constant polynomials preserve infinite sets in a finitely factorizing semiring, i.e., given an infinite set $A'$ of semiring elements and a non-constant polynomial $p$, then also $p(A')$ forms an infinite set. This lifts Observation 2 to the level of polynomials.

**Lemma 9.** *Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be a finitely factorizing semiring, and let $p \in P(A_+, X_1)$ be a zero-free polynomial with $\mathrm{var}(p) \neq \emptyset$. Furthermore, let $A' \subseteq A$. The set $A'$ is infinite, if and only if the set $p(A')$ is infinite.*

*Proof.* The statement is trivial, if $A$ is finite. Thus assume that $A$ is infinite. Then, according to Observation 3, the semiring $\mathcal{A}$ is positive. Clearly, if $p(A')$ is infinite, then $A'$ is also infinite. It remains to show that $p(A')$ is infinite, whenever $A'$ is infinite. We perform induction on the structure of $p$ to prove this statement.

**Induction base:** If $p = x_1$, then the claim trivially holds, because $p(A') = A'$.

**Induction step:** Let $p = p_1 + p_2$ for some zero-free polynomials $p_1, p_2 \in P(A_+, X_1)$. Then by definition $p(A') = \{\, p_1(a) \oplus p_2(a) \mid a \in A' \,\}$. Moreover, since $x_1 \in \mathrm{var}(p)$, we have $x_1 \in \mathrm{var}(p_1)$ or $x_1 \in \mathrm{var}(p_2)$ again by definition. Hence by induction hypothesis $p_1(A')$ or $p_2(A')$ is infinite. Thus also $\{\, (p_1(a), p_2(a)) \mid a \in A' \,\}$ is infinite. Consequently, $p(A')$ is infinite, because a finite $p(A')$ contradicts to the finitely factorizing property.

Let $p = p_1 \cdot p_2$ for some zero-free polynomials $p_1, p_2 \in P(A_+, X_1)$. Again by definition $p(A') = \{\, p_1(a) \odot p_2(a) \mid a \in A' \,\}$. Since $x_1 \in \mathrm{var}(p)$, we conclude that $x_1 \in \mathrm{var}(p_1)$ or $x_1 \in \mathrm{var}(p_2)$ by definition. For the sake of notational convenience assume that $x_1 \in \mathrm{var}(p_1)$; the arguments presented are symmetric, so they apply to $x_1 \in \mathrm{var}(p_2)$ as well. By induction hypothesis $p_1(A')$ is infinite, hence $\{\, (p_1(a), p_2(a)) \mid a \in A' \,\}$ is also infinite. It follows that also $\{\, (p_1(a), p_2(a)) \mid a \in A'_+ \,\}$ is infinite. Since $p_2$ is zero-free, $\mathcal{A}$ is positive, and $a \neq \mathbf{0}$, we conclude that $p_2(a) \neq \mathbf{0}$ by Observation 6. Thus $p(A')$ must be infinite, else we arrive at a contradiction to the finitely factorizing property. ☐

We can apply the previous lemma in order to derive a corollary, which illuminates the interrelation between the syntactic and the semantic degree of a polynomial and the set of variables of that polynomial.

**Corollary 10.** *Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be an infinite, but finitely factorizing semiring. Moreover, let $p \in P(A_+, X_n)$ be a zero-free polynomial. Then the following statements are equivalent.*

*(i)* $\mathrm{var}(p) \neq \emptyset$.

*(ii)* $1 \leq \deg(p)$.

*(iii)* $1 \leq \deg_{\mathrm{s}}(p)$.

*(iv) For every $a \in A$, we have $p \not\equiv a$.*

*Proof.* The chain of implications (iv) ⇒ (iii) ⇒ (ii) ⇒ (i) holds by definition, so we just prove (i) ⇒ (iv). By assumption we have $x_i \in \mathrm{var}(p)$ for some index $i \in [n]$. Consequently,

$$p' = p[x_1 \leftarrow \mathbf{1}, \ldots, x_{i-1} \leftarrow \mathbf{1}, x_i \leftarrow x_1, x_{i+1} \leftarrow \mathbf{1}, \ldots, x_n \leftarrow \mathbf{1}]$$

is a zero-free polynomial of $P(A_+, X_1)$ and $x_1 \in \mathrm{var}(p')$ by several applications of Observation 8. Hence Lemma 9 yields that $p'(A)$ is infinite, because $A$ is infinite. Moreover it should be clear that

$$p'(A) = p(\{\mathbf{1}\}, \ldots, \{\mathbf{1}\}, A, \{\mathbf{1}\}, \ldots, \{\mathbf{1}\}) \subseteq p(A, \ldots, A).$$

We conclude that $p(A, \ldots, A)$ is infinite; thus for every semiring element $a \in A$ we have $p \not\equiv a$. ☐

**Lemma 11.** *Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be an infinite, finitely factorizing semiring and $p \in P(A, X_n)$ be a polynomial. Then it is decidable whether*

*(i) $p \in \bigcup_{a \in A} [a]_{\equiv}$, i.e., $\deg_{\mathrm{s}}(p) \leq 0$, and*

*(ii) for some given semiring element $a \in A$ we have $p \equiv a$.*

*Proof.* By Lemma 7 either $p \equiv \mathbf{0}$, which decides $p \equiv a$ if $a = \mathbf{0}$, or there exists an effectively computable zero-free polynomial $p' \in P(A_+, X_n)$ with $p \equiv p'$. In the latter case we observe that $p \not\equiv a$ for every $a \in A$, if and only if $\mathrm{var}(p') \neq \emptyset$ by Corollary 10. Hence $p \in \bigcup_{a \in A} [a]_{\equiv}$ if and only if $\mathrm{var}(p') = \emptyset$. Finally we consider the case $\mathrm{var}(p') = \emptyset$, but then $p' \in P(A_+, \emptyset)$ and we can compute whether $p' \equiv a$. ☐

### 3. Monotonic Semirings

In this section we will introduce monotonic semirings, which we will use in our cost-finiteness results in Section 5. Roughly speaking, in a monotonic semiring there exists a partial order and the addition or multiplication of some semiring element $a \in A$ with another semiring element $a' \in A_+$ might only yield a result which is greater or equal than $a$ with respect to the partial order on $A$. Moreover, for multiplication we demand strictness in all possible cases. We will later use the partial order and the monotonicity in order to show that a tree automaton with cost function might produce arbitrarily large costs (semiring elements), precisely speaking in Section 5 we try to decide whether the set of accepting costs is finite. Clearly, finite semirings are obviously not interesting with respect to this question.

**Definition 12.** *An infinite semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ is* monotonic*, if there is a partial order $\preceq$ on $A$ such that the following conditions hold.*

*(i) For every $a_1, a_2 \in A$ the inequality $a_1 \preceq a_1 \oplus a_2$ holds and*

*(ii) for every $a_1, a_2 \in A_+$ with $a_2 \neq \mathbf{1}$ we have $a_1 \prec a_1 \odot a_2$ and $a_1 \prec a_2 \odot a_1$.*

Note that the only finite semiring fulfilling Conditions (i) and (ii) is the Boolean semiring Bool. In the following we add the partial order to the signature of the semiring and write $(A, \oplus, \odot, \mathbf{0}, \mathbf{1}, \preceq)$ for a semiring which is monotonic with respect to the partial order $\preceq$.

For example, the semiring of the non-negative integers $\mathrm{Nat} = (\mathbb{N}, +, \cdot, 0, 1, \leq)$, the arctic semiring $\mathrm{Arct} = (\mathbb{N} \cup \{-\infty\}, \max, +, (-\infty), 0, \leq)$, and for some alphabet $\Sigma$ the following subsemiring of the finite-language semiring

$$\mathrm{FLang}_\varepsilon(\Sigma) = (\{\, L \in \mathcal{P}_\mathrm{f}(\Sigma^*) \mid \varepsilon \in L \,\} \cup \{\emptyset\}, \cup, \circ, \emptyset, \{\varepsilon\}, \subseteq)$$

are all monotonic. Moreover, for every $n \in \mathbb{N}_+$ the semiring $\mathrm{Mat}_n(\mathbb{N}_+)$ is monotonic with respect to the natural order $\sqsubseteq$ and at the end of Section 6 we also show that Lcm, $\mathrm{FSet}(\mathbb{N})$, and $\mathrm{FLang}(\Sigma)$ are monotonic. Finally, the real number semiring $(\{\, a \in \mathbb{R}_+ \mid 1 \leq a \,\} \cup \{0\}, +, \cdot, 0, 1, \leq)$ is also monotonic, but certainly not finitely factorizing. So far, monotonic semirings seem to be quite restricted semirings, but in the next lemma we will characterize exactly which naturally ordered semirings are monotonic. Basically, only Condition (ii) remains.

**Lemma 13.** *Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be an infinite and naturally ordered semiring with the natural order $\sqsubseteq$. Then the following statements are equivalent.*

*(i) $\mathcal{A}$ is monotonic with respect to $\sqsubseteq$.*

*(ii) For every $a_1, a_2 \in A_+$ with $a_2 \neq \mathbf{1}$ we have $a_1 \sqsubset a_1 \odot a_2$ and $a_1 \sqsubset a_2 \odot a_1$.*

*Proof.* The implication (i) $\Rightarrow$ (ii) is obvious. We deduce for every $a_1, a_2 \in A$ the property $a_1 \sqsubseteq a_1 \oplus a_2$ from the fact that $\sqsubseteq$ is the natural order. This shows Condition (i) of Definition 12. Finally Condition (ii) of Definition 12 is given by the assumption. $\square$

In Subsection 2.4 we have already seen that positivity of the semiring is important for a number of results about polynomials. Later it will turn out that the properties one-summand freeness and one-product freeness are also very important. The lemma to follow will state that monotonic semirings have all of the aforementioned properties. For the rest of this section, let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1}, \preceq)$ be a monotonic semiring.

**Lemma 14.** *The following properties are satisfied:*

*(i)* $\mathbf{0} \preceq a$ *for every* $a \in A$ *and* $\mathbf{1} \preceq a$ *for every* $a \in A_+$*.*

*(ii)* $\mathcal{A}$ *is positive.*

*(iii)* $\mathcal{A}$ *is one-summand and one-product free.*

*Proof.* Let $a_1, a_2 \in A$.

(i) By Condition (i) of Definition 12, we obtain $\mathbf{0} \preceq \mathbf{0} \oplus a_2 = a_2$. If $a_2 \neq \mathbf{0}$, then by Condition (ii) of Definition 12, $\mathbf{1} \preceq \mathbf{1} \odot a_2 = a_2$.

(ii) Firstly we show zero-sum freeness. Assume that $a_1 \oplus a_2 = \mathbf{0}$. We show $a_1 = \mathbf{0}$ and $a_2 = \mathbf{0}$ by contradiction. For this, assume $a_2 \in A_+$. Then, by Item (i) and Condition (i) of Definition 12, we have $\mathbf{0} \prec a_2 \preceq a_1 \oplus a_2$, which contradicts $\mathbf{0} = a_1 \oplus a_2$. Thus $a_2 = \mathbf{0}$ and hence $\mathcal{A}$ is zero-sum free.

Finally, we consider zero-divisor freeness. Let $a_1 \odot a_2 = \mathbf{0}$. If $a_1, a_2 \in A_+$, then, by Item (i) and Condition (ii) of Definition 12, $\mathbf{0} \prec a_1 \preceq a_1 \odot a_2$. This contradicts $a_1 \odot a_2 = \mathbf{0}$. Hence $a_1 = \mathbf{0}$ or $a_2 = \mathbf{0}$, and thus $\mathcal{A}$ is zero-divisor free.

(iii) Let us show that $a_1 \oplus a_2 = \mathbf{1}$ implies $a_1, a_2 \in \{\mathbf{0}, \mathbf{1}\}$ by contradiction. Assume that $a_1 \oplus a_2 = \mathbf{1}$ and $a_1 \notin \{\mathbf{0}, \mathbf{1}\}$. Then, by Item (i) and Condition (i) of Definition 12, we obtain $\mathbf{1} \prec a_1 \preceq a_1 \oplus a_2$, which contradicts $a_1 \oplus a_2 = \mathbf{1}$. Consequently, $\mathcal{A}$ is one-summand free.

Let us show by contradiction that $a_1 \odot a_2 = \mathbf{1}$ implies $a_1 = \mathbf{1}$ and $a_2 = \mathbf{1}$. Assume that $a_1 \neq \mathbf{1}$ or $a_2 \neq \mathbf{1}$ and $a_1 \odot a_2 = \mathbf{1}$. Apparently, $a_1, a_2 \in A_+$. Hence by Item (i) and Condition (ii) of Definition 12 we obtain $\mathbf{1} \preceq a_2 \prec a_1 \odot a_2$ or $\mathbf{1} \preceq a_1 \prec a_1 \odot a_2$. This contradicts to $a_1 \odot a_2 = \mathbf{1}$. Consequently, $\mathcal{A}$ is one-product free.

□

Now we show that some of the conditions in the definition of monotonic semirings (cf. Definition 12) regarding the partial order $\preceq$ can be lifted from semiring elements to polynomials, if the semiring is additionally finitely factorizing. In particular, the statements in Items (ii) and (iii) will be applied in Section 5 to show that certain polynomials can be used to generate an infinite number of ever-increasing semiring elements as in $a \prec p(a) \prec p^2(a) \prec \cdots$.

**Lemma 15.** *Let* $\mathcal{A}$ *be finitely factorizing,* $a_1, \ldots, a_n \in A_+$ *be semiring elements,* $p \in P(A_+, X_n)$ *be a zero-free polynomial, and* $m \in A_+[X_n]$ *be a zero-free monomial for some positive integer* $n \in \mathbb{N}_+$*.*

*(i) If* $1 \leq \deg_s(p)$*, then* $a_i \preceq p(a_1, \dots, a_n)$ *for every index* $i \in [n]$ *such that* $x_i \in \mathrm{var}(p)$*.*

*(ii) If* $2 \leq \deg(m)$ *and* $\mathbf{1} \prec a_i$ *for every index* $i \in [n]$ *such that* $x_i \in \mathrm{var}(m)$*, then also* $a_i \prec m(a_1, \dots, a_n)$ *for every index* $i \in [n]$ *such that* $x_i \in \mathrm{var}(m)$*.*

*(iii) If* $2 \leq \deg_s(p)$ *and* $\mathbf{1} \prec a_1$*, then* $a_1 \prec p(a_1, \dots, a_1)$*.*

*Proof.* We prove the items separately.

(i) Let $i \in [n]$ be such that $x_i \in \mathrm{var}(p)$. Hence $1 \leq |p|_{x_i}$. It remains to prove $a_i \preceq p(a_1, \dots, a_n)$.

**Induction base:** Let $p = x_i$. Then $p(a_1, \dots, a_n) = a_i$ fulfilling the property.

**Induction step:** Let $p = p_1 \times p_2$ for some polynomials $p_1, p_2 \in P(A_+, X_n)$ and $\times \in \{+, \cdot\}$; thus $p(a_1, \dots, a_n) = p_1(a_1, \dots, a_n) \otimes p_2(a_1, \dots, a_n)$, where $\otimes = \oplus$ if $\times = +$ and $\otimes = \odot$ otherwise. Since $1 \leq |p|_{x_i}$ we have $1 \leq |p_1|_{x_i}$ or $1 \leq |p_2|_{x_i}$, hence $1 \leq \deg_s(p_1)$ or $1 \leq \deg_s(p_2)$ by Corollary 10. Without loss of generality assume the former; the presented argumentation is symmetric. Then by induction hypothesis we have $a_i \preceq p_1(a_1, \dots, a_n)$. Besides, $p_2(a_1, \dots, a_n) \in A_+$ by Observation 6, hence $a_i \preceq p_1(a_1, \dots, a_n) \otimes p_2(a_1, \dots, a_n)$ by Condition (i) or (ii) of Definition 12.

(ii) By assumption $2 \leq \deg(m)$, and consequently, there exist $m_1, m_2 \in A_+[X_n]$ such that $m \equiv m_1 \cdot m_2$ and $1 \leq \deg(m_k)$ for every $k \in [2]$. Moreover, there exist variables $x_j, x_{j'} \in X_n$ for some indices $j, j' \in [n]$ and monomials $m_{1,1}, m_{1,2}, m_{2,1}, m_{2,2} \in A_+[X_n]$ such that $m_1 \equiv m_{1,1} \cdot x_j \cdot m_{1,2}$ and $m_2 \equiv m_{2,1} \cdot x_{j'} \cdot m_{2,2}$. Hence, $m \equiv m_{1,1} \cdot x_j \cdot m_{1,2} \cdot m_{2,1} \cdot x_{j'} \cdot m_{2,2}$. By Corollary 10 and Item (i) we have $a_j \preceq (x_j \cdot m_{1,2} \cdot m_{2,1})(a_1, \dots, a_n)$ and for every $a \in \{a_j, a_{j'}\}$, since $(x_j \cdot m_{1,2} \cdot m_{2,1})(a_1, \dots, a_n) \in A_+$ and $\mathbf{1} \prec a$,

$$\begin{aligned} a &\prec (x_j \cdot m_{1,2} \cdot m_{2,1})(a_1, \dots, a_n) \odot a_{j'} \\ &= (x_j \cdot m_{1,2} \cdot m_{2,1} \cdot x_{j'})(a_1, \dots, a_n). \end{aligned}$$

We complete the proof using monotonicity as follows.

$$\begin{aligned} a &\prec (x_j \cdot m_{1,2} \cdot m_{2,1} \cdot x_{j'})(a_1, \dots, a_n) \\ &\preceq (m_{1,1} \cdot x_j \cdot m_{1,2} \cdot m_{2,1} \cdot x_{j'} \cdot m_{2,2})(a_1, \dots, a_n) \\ &= m(a_1, \dots, a_n). \end{aligned}$$

Hence we have proved the statement in case $i = j$ or $i = j'$. Otherwise $x_i \in \mathrm{var}(m_{k,l})$ for some $k, l \in [2]$. We show the case $x_i \in \mathrm{var}(m_{1,2})$; the remaining cases are similar. If $x_i \in \mathrm{var}(m_{1,2})$ then $a_i \preceq m_{1,2}(a_1, \dots, a_n)$ by Item (i). Moreover, $\mathbf{1} \prec a_j$. Consequently,

$$\begin{aligned} a_i &\prec (x_j \cdot m_{1,2} \cdot m_{2,1} \cdot x_{j'})(a_1, \dots, a_n) \\ &\preceq (m_{1,1} \cdot x_j \cdot m_{1,2} \cdot m_{2,1} \cdot x_{j'} \cdot m_{2,2})(a_1, \dots, a_n) \\ &= m(a_1, \dots, a_n). \end{aligned}$$

(iii) Since for each polynomial there exists an equivalent polynomial in normal form, where according to Item (ii) the property holds for every monomial with degree greater than 1, we can readily conclude the stated by Condition (i) in Definition 12.

□

After having shown that polynomials of semantic degree at least 2 yields a result strictly greater than $a$, if supplied with $a$ for all variables, we will now show that this property already holds for certain linear polynomials. In fact, later we will exclude exactly those linear polynomials, because of this property.

**Lemma 16.** *Let $p \in P(A, X_1)$ be a polynomial with $\deg_s(p) = 1$. Moreover, let $p \notin \bigcup_{a' \in A}[x_1 + a']_\equiv$. Then for every $a \in A_+$ we have $a \prec p(a)$.*

*Proof.* Since $p \notin \bigcup_{a' \in A}[x_1 + a']_\equiv$ while $\deg_s(p) = 1$, we have that

$$p \equiv a_1 \cdot x_1 \cdot b_1 + \ldots + a_n \cdot x_1 \cdot b_n + a_0$$

for some integer $n \in \mathbb{N}_+$, semiring elements $a_1, \ldots, a_n, b_1, \ldots, b_n \in A_+$, $a_0 \in A$ due to Lemma 5. Moreover, there either exists an index $i \in [n]$ such that (i) $a_i \neq \mathbf{1}$ or $b_i \neq \mathbf{1}$, or (ii) $a_i = b_i = \mathbf{1}$ for every index $i \in [n]$, $2 \leq n$, and $\mathcal{A}$ is not idempotent.

Case (i): Let $a_i \neq \mathbf{1}$ or $b_i \neq \mathbf{1}$ for some index $i \in [n]$. Clearly, $a \prec a_i \odot a \odot b_i \preceq p(a)$ by Conditions (i) and (ii) of Definition 12. Thus we have proved the statement in this case.

Case (ii): Let $a_i = b_i = \mathbf{1}$ for all indices $i \in [n]$, $2 \leq n$, and $\mathbf{1} \neq \mathbf{1} \oplus \mathbf{1}$. Then

$$p \equiv (\mathbf{1} \oplus \mathbf{1}) \cdot x_1 + \sum_{i \in [n-2]} x_1 + a_0$$

which reduces the stated to the previous case, because $\mathbf{1} \neq \mathbf{1} \oplus \mathbf{1}$. □

Note that idempotency is decidable, because a semiring is idempotent, if and only if $\mathbf{1} \oplus \mathbf{1} = \mathbf{1}$, which is certainly decidable in a computable semiring. Moreover, we obtain another characterization of idempotency, which we will present in the observation to follow.

**Observation 17.** *The following two statements are equivalent.*

*(i) For every $a \in A$ it holds that $a \oplus a = a$.*

*(ii) There is an $a \in A_+$ satisfying $a \oplus a = a$.*

*Proof.* Certainly, (i) implies (ii). So we still have to prove (ii) $\Rightarrow$ (i). Let $a \oplus a = a$ and $\mathbf{1} \oplus \mathbf{1} \neq \mathbf{1}$. By monotonicity we conclude $\mathbf{1} \prec \mathbf{1} \oplus \mathbf{1}$. Then $a \prec (\mathbf{1} \oplus \mathbf{1}) \odot a = a \oplus a$ which contradicts to the assumption. Hence $\mathbf{1} \oplus \mathbf{1} = \mathbf{1}$ and by distributivity $a' \oplus a' = a'$ for every $a' \in A$. □

Another important property of monotonic and finitely factorizing semirings is the following. Given two zero-free polynomials $p_1, p_2 \in P(A_+, X_n)$ such that

$\mathrm{var}(p_1) \neq \mathrm{var}(p_2)$. Then also $p_1 \not\equiv p_2$ follows. We justify the case where for some index $i \in [n]$ we have $x_i \in \mathrm{var}(p_1) \setminus \mathrm{var}(p_2)$. The remaining case is analogous. Let $p_1' = p_1[x_1 \leftarrow \mathbf{1}, \ldots, x_{i-1} \leftarrow \mathbf{1}, x_{i+1} \leftarrow \mathbf{1}, \ldots, x_n \leftarrow \mathbf{1}]$ and $p_2' = p_2[x_1 \leftarrow \mathbf{1}, \ldots, x_{i-1} \leftarrow \mathbf{1}, x_{i+1} \leftarrow \mathbf{1}, \ldots, x_n \leftarrow \mathbf{1}]$. By several applications of Observation 8, $\mathrm{var}(p_1') = \{x_i\}$, whereas $\mathrm{var}(p_2') = \emptyset$. By Corollary 10 we have $1 \leq \deg_{\mathrm{s}}(p_1')$ and $\deg_{\mathrm{s}}(p_2') = 0$. Hence $p_2' \equiv a$ for some semiring element $a \in A$. However, $a' \preceq p_1'(a', \ldots, a')$ for every $a' \in A$ by (i) of Lemma 15. This clearly yields $p_1 \not\equiv p_2$.

We already started to show decidability results for the equivalence of certain polynomials (e.g., Lemma 11). Now we continue this with the next lemma, in which, roughly speaking, we show that the equivalence to a variable $x$ is also decidable in a monotonic and finitely factorizing semiring.

**Lemma 18.** *Let $\mathcal{A}$ be finitely factorizing, $p \in P(A, X_n)$ a polynomial, and $x \in X_n$ a variable for some $n \in \mathbb{N}_+$. It is decidable whether $p \equiv x$.*

*Proof.* Let us define, for every $a \in A$, the function $h_a : P(A, X_n) \longrightarrow \{0, 1\}$ by letting

$$h_a(p) = \begin{cases} 1 & , \text{if } p \equiv a \\ 0 & , \text{otherwise.} \end{cases}$$

The mapping $h_a$ is computable by Lemma 11. Let us define the mapping $h_x : P(A, X_n) \longrightarrow \{0, 1\}$ in the following way. For every coefficient $a \in A$, variable $z \in X_n \setminus \{x\}$, and polynomials $p_1, p_2 \in P(A, X_n)$

$$h_x(a) = 0 \qquad h_x(x) = 1$$

$$h_x(z) = 0 \qquad h_x(p_1 \cdot p_2) = \big(h_x(p_1) \wedge h_{\mathbf{1}}(p_2)\big) \vee \big(h_x(p_2) \wedge h_{\mathbf{1}}(p_1)\big)$$

$$h_x(p_1 + p_2) = \Big(\big(h_x(p_1) \wedge h_{\mathbf{0}}(p_2)\big) \vee \big(h_x(p_2) \wedge h_{\mathbf{0}}(p_1)\big)\Big) \bigcirc \big(h_x(p_1) \wedge h_x(p_2)\big),$$

where $\bigcirc = \vee$, if $\mathcal{A}$ is idempotent, else $b_1 \bigcirc b_2 = b_1$ for every $b_1, b_2 \in \{0, 1\}$. Certainly, $h_x$ is also computable. Hence, in order to show that $p \equiv x$ is decidable, it is sufficient to prove that $p \equiv x$ if and only if $h_x(p) = 1$.

A straightforward induction shows that if $h_x(p) = 1$, then $p \equiv x$. Now we show by induction on the structure of $p$ that $p \equiv x$ implies $h_x(p) = 1$.

**Induction base:** Let $p = a$ for some semiring element $a \in A$. Then clearly $p \not\equiv x$, because by Corollary 10 we have $1 \leq \deg_{\mathrm{s}}(x)$ while $\deg_{\mathrm{s}}(a) \leq 0$. Hence our statement follows. Next let $p = z$ for some variable $z \in X_n$. Then $p \equiv x$ implies $z = x$, from which $h_x(p) = 1$ follows.

**Induction step:** Let $p = p_1 \cdot p_2$. We show that either (i) $p_1 \equiv x$ and $p_2 \equiv \mathbf{1}$ or (ii) $p_1 \equiv \mathbf{1}$ and $p_2 \equiv x$. This together with the induction hypothesis implies $h_x(p) = 1$.

Trivially, $p_1 \not\equiv \mathbf{0}$ and $p_2 \not\equiv \mathbf{0}$, else $p \equiv \mathbf{0}$. Thus, by Lemma 7, there are zero-free polynomials $p_1', p_2' \in P(A_+, X_n)$ such that $p_1 \equiv p_1'$ and $p_2 \equiv p_2'$. Since $p \equiv p_1' \cdot p_2'$ and $p \equiv x$, we conclude that $x \in \mathrm{var}(p_1')$ or $x \in \mathrm{var}(p_2')$. For the rest of the proof we assume the former and note that the proof using the latter assumption is absolutely symmetric. Consequently, by Corollary 10 we have $1 \leq \deg_{\mathrm{s}}(p'_1) = \deg_{\mathrm{s}}(p_1)$.

Further, we immediately obtain that $\mathrm{var}(p'_1) = \{x\}$ and $\mathrm{var}(p'_2) \subseteq \{x\}$, else $p \not\equiv p'_1 \cdot p'_2$. Moreover, assume that $1 \leq \deg_{\mathrm{s}}(p'_2)$. Then, by Lemma 15(i), $a \preceq p'_1(a, \ldots, a)$ and $a \preceq p'_2(a, \ldots, a)$ for every $a \in A \setminus \{\mathbf{0}, \mathbf{1}\}$ and

$$a \prec p'_1(a, \ldots, a) \odot p'_2(a, \ldots, a) = p(a, \ldots, a).$$

This constitutes a contradiction, so $\deg_{\mathrm{s}}(p'_2) = 0$. Further $p'_2 \equiv \mathbf{1}$, else $p'_2 \equiv a'$ for some $a' \in A \setminus \{\mathbf{0}, \mathbf{1}\}$ and $a' \prec p'_1(a', \ldots, a') \odot p'_2(a', \ldots, a') = p(a', \ldots, a')$ which is again contradictory. Therefore $p_2 \equiv p'_2 \equiv \mathbf{1}$ and using $p \equiv p'_1 \cdot p'_2$ we obtain $p \equiv p'_1 \equiv p_1$ and $p_1 \equiv x$.

Finally, let $p = p_1 + p_2$. We prove that either (i) $p_1 \equiv x$ and $p_2 \equiv \mathbf{0}$, or (ii) $p_1 \equiv \mathbf{0}$ and $p_2 \equiv x$, or (iii) $p_1 \equiv x \equiv p_2$ and $\mathcal{A}$ is idempotent. This, by the induction hypothesis, implies $h_x(p) = 1$.

Trivially, $p_1 \not\equiv \mathbf{0}$ or $p_2 \not\equiv \mathbf{0}$, else $p \equiv \mathbf{0}$. In order to prove that either (i) or (ii) or (iii) holds, we distinguish three different cases.

<u>Case 1:</u> $p_1 \not\equiv \mathbf{0}$ and $p_2 \equiv \mathbf{0}$. Then, from $p = p_1 + p_2$ we obtain $p \equiv p_1 \equiv x$, hence (i) holds.

<u>Case 2:</u> $p_1 \equiv \mathbf{0}$ and $p_2 \not\equiv \mathbf{0}$. Analogously with Case 1, now (ii) holds.

<u>Case 3:</u> $p_1 \not\equiv \mathbf{0}$ and $p_2 \not\equiv \mathbf{0}$. We show that (iii) holds. By Lemma 7, there are zero-free polynomials $p'_1, p'_2 \in P(A_+, \{x\})$ such that $p_1 \equiv p'_1$ and $p_2 \equiv p'_2$. We can also conclude that $\mathrm{var}(p_1) \subseteq \{x\}$ and $\mathrm{var}(p_2) \subseteq \{x\}$.

We show that $1 \leq \deg(p'_1)$ and $1 \leq \deg(p'_2)$. Assume, on the contrary, that $\deg(p'_1) = 0$. Since $p'_1 \not\equiv \mathbf{0}$, we have $p'_1 \equiv a$ for some $a \in A_+$. On the other hand, $p \equiv x$ implies that $p(\mathbf{0}, \ldots, \mathbf{0}) = \mathbf{0}$ and thus from $p \equiv p'_1 + p'_2$ it follows that $p'_1(\mathbf{0}, \ldots, \mathbf{0}) \oplus p'_2(\mathbf{0}, \ldots, \mathbf{0}) = \mathbf{0}$. This is a contradiction, because $p'_1(\mathbf{0}, \ldots, \mathbf{0}) = a$ and, by Lemma 14, $\mathcal{A}$ is zero-sum free.

Then, by Corollary 10 we have $1 \leq \deg_{\mathrm{s}}(p'_1)$ and $1 \leq \deg_{\mathrm{s}}(p'_2)$. Clearly, $\deg_{\mathrm{s}}(p'_1) = 1$ and $\deg_{\mathrm{s}}(p'_2) = 1$, because otherwise, by Lemma 15(iii), $a \prec p'_1(a, \ldots, a)$ or $a \prec p'_2(a, \ldots, a)$, respectively, for every $a \in A \setminus \{\mathbf{0}, \mathbf{1}\}$ and

$$a \prec p'_1(a, \ldots, a) \oplus p'_2(a, \ldots, a) = p(a, \ldots, a),$$

which contradicts to $p \equiv x$.

Moreover, $p'_1, p'_2 \in \bigcup_{a' \in A} [x + a']_{\equiv}$, else by the previous chain of reasoning using Lemma 16, we again have a contradiction. Thus $p'_1 \equiv x + a_1$ and $p'_2 \equiv x + a_2$ for some $a_1, a_2 \in A$. We can show easily that $a_1 = a_2 = \mathbf{0}$ as follows. Assume, on the contrary, that $a_1 \neq \mathbf{0}$. Then $p'_1(\mathbf{0}, \ldots, \mathbf{0}) \oplus p'_2(\mathbf{0}, \ldots, \mathbf{0}) = a_1 \oplus p'_2(\mathbf{0}, \ldots, \mathbf{0}) \neq \mathbf{0}$, because $\mathcal{A}$ is zero-sum free. On the other hand $p(\mathbf{0}, \ldots, \mathbf{0}) = \mathbf{0}$ because $p \equiv x$. This is a contradiction because $p \equiv p'_1 + p'_2$. Hence $p'_1 \equiv x$ and $p'_2 \equiv x$.

Finally, we have to prove that $\mathcal{A}$ is idempotent, which readily follows from

$$p'_1(a, \ldots, a) \oplus p'_2(a, \ldots, a) = a \oplus a = a$$

with the help of Observation 17. Thus (iii) holds. □

Next we show that, for every semiring element $a$, the equivalence of a polynomial to $x+a$ is also decidable in a monotonic and finitely factorizing semiring.

**Lemma 19.** *Let $\mathcal{A}$ be finitely factorizing, $p \in P(A, X_n)$ be a polynomial, $x \in X_n$ be a variable for some $n \in \mathbb{N}_+$, and $a \in A$ be a semiring element. It is decidable whether (i) $p \in \bigcup_{a' \in A}[x+a']_\equiv$ and (ii) $p \equiv x + a$.*

*Proof.* Roughly speaking, firstly one decides whether $p \equiv \mathbf{0}$ by Lemma 11. If not, then by Lemma 7 a zero-free normal form of $p \equiv m_1 + \ldots + m_k$ is effectively computable for some integer $k \in \mathbb{N}_+$ and monomials $m_1, \ldots, m_k \in A_+[X_n]$. Then $p \in \bigcup_{a' \in A}[x+a']_\equiv$ if and only if $\sum_{i \in [k], \mathrm{var}(m_i) \neq \emptyset} m_i \equiv x$, which is decidable by Lemma 18. And $p \equiv x+a$ if and only if $p \in \bigcup_{a' \in A}[x+a']_\equiv$ and $\sum_{i \in [k], \mathrm{var}(m_i) = \emptyset} m_i \equiv a$, where the latter is decidable by Lemma 11. We leave the details of the proof to the reader. □

The next lemma shows that linear polynomials of the form $x_1 + a$ are closed under substitution, and moreover, given such a linear polynomial, we can only decompose it into such polynomials. Thus this class is closed under compositions and decompositions. This property is central in Section 5, where we will apply it together with the last lemma of this section in order to show the following. Roughly speaking, if a computation of a tree automaton with costs is assigned such a linear cost polynomial, then the cost polynomial of each transition of the automaton within that computation must also be very restricted.

**Lemma 20.** *Let $\mathcal{A}$ be finitely factorizing and $p_1, p_2 \in P(A, X_1)$ polynomials. Then $p_1, p_2 \in [x_1]_\equiv$ if and only if $p_1[p_2] \in [x_1]_\equiv$. Moreover, if $\mathcal{A}$ is idempotent, then*

$$p_1, p_2 \in \bigcup_{a \in A}[x_1 + a]_\equiv \iff p_1[p_2] \in \bigcup_{a \in A}[x_1 + a]_\equiv.$$

*Proof.* The proof of the forward direction of both statements is routine therefore we give the proofs of the backward directions only. Let $C = \bigcup_{a \in A}[x_1 + a]_\equiv$. First we prove the second statement. Therefore, let us assume that $\mathcal{A}$ is idempotent and that $p_1[p_2] \equiv x_1 + a$ for some $a \in A$. Apparently, $p_1 \not\equiv \mathbf{0} \not\equiv p_2$, so without loss of generality we can furthermore assume that $p_1$ and $p_2$ are zero-free by Lemma 7. Moreover, $1 \le \deg_{\mathrm{s}}(p_1[p_2])$ by Corollary 10. We continue with a case distinction.

<u>Case 1:</u> Let $\deg_{\mathrm{s}}(p_1) = 0$ or $\deg_{\mathrm{s}}(p_2) = 0$. Then $p_1 \equiv a_1$ or $p_2 \equiv a_2$ for some $a_1, a_2 \in A$. Consequently, $\mathrm{var}(p_1[p_2]) = \emptyset$ and thus $\deg_{\mathrm{s}}(p_1[p_2]) = 0$ contradicting to the assumption $1 \le \deg_{\mathrm{s}}(p_1[p_2])$.

<u>Case 2:</u> Let $1 \le \deg_{\mathrm{s}}(p_1)$ and $1 \le \deg_{\mathrm{s}}(p_2)$. Consequently, by Lemma 15 we have both $a' \preceq p_1(a')$ and $a' \preceq p_2(a')$ for every $a' \in A_+$.

<u>Subcase 2.1:</u> Let $2 \le \deg_{\mathrm{s}}(p_1)$ or $2 \le \deg_{\mathrm{s}}(p_2)$. Let $a'' \in A \setminus \{\mathbf{0}, \mathbf{1}\}$ be a non-unit semiring element. We now select a semiring element $a' \in A$ such that we are able to derive a contradiction. Therefore, let

$$a' = \begin{cases} \mathbf{1} \oplus a'' & \text{, if } a \in \{\mathbf{0}, \mathbf{1}\} \\ a & \text{, otherwise} \end{cases}.$$

We immediately observe that $(p_1[p_2])(a') = a'$, because $p_1[p_2] \equiv x_1 + a$. However, using Condition (i) of Definition 12, Item (i) of Lemma 14, and Item (i) of Lemma 15 we conclude

$$\mathbf{1} \prec a' \preceq p_2(a') \preceq p_1(p_2(a')) = (p_1[p_2])(a'),$$

where at least one of the last two inequalities is strict due to Item (iii) in Lemma 15. In case $2 \leq \deg_s(p_1)$ it is the last one, whereas it is the former one, if $2 \leq \deg_s(p_2)$. Thus we arrived at a contradiction, which only leaves one case.

Subcase 2.2: Let $\deg_s(p_1) = 1 = \deg_s(p_2)$. For a contradiction assume that $p_1 \notin C$ or $p_2 \notin C$. Clearly, $(p_1[p_2])(\mathbf{1} \oplus a) = \mathbf{1} \oplus a$ due to $p_1[p_2] \equiv x_1 + a$ and the fact that $\mathcal{A}$ is idempotent. Thus, by Item (i) of Lemma 15,

$$\mathbf{0} \prec \mathbf{1} \oplus a \preceq p_2(\mathbf{1} \oplus a) \preceq p_1(p_2(\mathbf{1} \oplus a)) = (p_1[p_2])(\mathbf{1} \oplus a),$$

where again one of the last two inequalities is strict by Lemma 16. Provided that $p_1 \notin C$ it is the latter, otherwise the former. Anyway we derived a contradiction, hence $p_1, p_2 \in C$.

Next we prove the first statement. Therefore, let us assume that $p_1[p_2] \equiv x_1$. Now $p_1 \not\equiv \mathbf{0} \not\equiv p_2$ and the same case distinction can be made as above out of which Case 1 leads to a contradiction in the same way as before because there we did not use that $\mathcal{A}$ is idempotent. Moreover, Case 2 (including Subcases 2.1 and 2.2) is also sound because its proof also works for $a = \mathbf{0}$ (even if $\mathcal{A}$ is not idempotent). Thus we obtain that $p_1, p_2 \in C$. Let us assume that $p_1 \equiv x_1 + a'$ or $p_2 \equiv x_1 + a'$ for some $a' \neq \mathbf{0}$. Apparently, $(p_1[p_2])(\mathbf{0}) = \mathbf{0}$ by $p_1[p_2] \equiv x_1$, but $a' \preceq p_1(\mathbf{0})$ or $a' \preceq p_2(\mathbf{0})$. This yields by Condition (i) of Definition 12

$$\mathbf{0} \prec a' \preceq p_1(p_2(\mathbf{0})) = (p_1[p_2])(\mathbf{0}),$$

which is again contradictory. Thus, $p_1, p_2 \in [x_1]_\equiv$ which finishes the proof of the first statement and thus of the lemma. □

**Lemma 21.** *Let $\mathcal{A}$ be finitely factorizing and idempotent, and let $p_1 \in P(A, X_n)$ be a polynomial and $a_1, \ldots, a_n \in A \setminus \{\mathbf{0}, \mathbf{1}\}$ be non-zero semiring elements for some integer $n \in \mathbb{N}_+$. Finally, let $i \in [n]$. Then*

$$p_1[x_1 \leftarrow a_1, \ldots, x_{i-1} \leftarrow a_{i-1}, x_{i+1} \leftarrow a_{i+1}, \ldots, x_n \leftarrow a_n] \in \bigcup_{a \in A} [x_i + a]_\equiv,$$

*if and only if $p_1 \equiv x_i + p'$ with $p' \in P(A, X_n \setminus \{x_i\})$.*

*Proof.* Sufficiency is readily seen, so it remains to prove necessity. Apparently, $p_1 \not\equiv \mathbf{0}$ and there exists a normal form $p_1 \equiv m_1 + \cdots + m_k$ for some integer $k \in \mathbb{N}_+$ and zero-free monomials $m_1, \ldots, m_k \in A_+[X_n]$ by Lemma 7. Let

$$p_2 = p_1[x_1 \leftarrow a_1, \ldots, x_{i-1} \leftarrow a_{i-1}, x_{i+1} \leftarrow a_{i+1}, \ldots, x_n \leftarrow a_n].$$

and $p_2 \equiv x_i + a$ for some $a \in A$. Further, let $j \in [k]$ be an arbitrary index such that $x_i \in \operatorname{var}(m_j)$. We will prove that $m_j \equiv x_i$, thereby proving the statement.

Apparently, we have that $1 \leq \deg_s(m_j)$ by Corollary 10. Furthermore, assume that $2 \leq \deg_s(m_j)$. Then by Lemma 15 we conclude that

$$\begin{aligned} a' \oplus a &\prec m_j(a_1, \ldots, a_{i-1}, a' \oplus a, a_{i+1}, \ldots, a_n) \\ &\preceq p_1(a_1, \ldots, a_{i-1}, a' \oplus a, a_{i+1}, \ldots, a_n) \\ &= p_2(a_1, \ldots, a_{i-1}, a' \oplus a, a_{i+1}, \ldots, a_n) \\ &= a' \oplus a \end{aligned}$$

for every $a' \in A \setminus \{\mathbf{0}, \mathbf{1}\}$. However, this is contradictory, hence $\deg_s(m_j) = 1$. Thus $m_j \equiv m'_1 \cdot x_i \cdot m'_2$ for some zero-free monomials $m'_1, m'_2 \in A_+[X_n]$. If $m'_1 \not\equiv \mathbf{1}$ or $m'_2 \not\equiv \mathbf{1}$, then $m_j \notin [x_i]_\equiv$ by Lemma 18 because $h_{x_i}(m'_1 \cdot x_i \cdot m'_2) = 0$. Consequently, with the help of Lemma 16 we conclude

$$\begin{aligned} \mathbf{1} \oplus a &\prec m_j(a_1, \ldots, a_{i-1}, \mathbf{1} \oplus a, a_{i+1}, \ldots, a_n) \\ &\preceq p_1(a_1, \ldots, a_{i-1}, \mathbf{1} \oplus a, a_{i+1}, \ldots, a_n). \end{aligned}$$

This, however, contradicts to $p_2(a_1, \ldots, a_{i-1}, \mathbf{1} \oplus a, a_{i+1}, \ldots, a_n,) = \mathbf{1} \oplus a \oplus a = \mathbf{1} \oplus a$. Hence $m'_1 \equiv \mathbf{1} \equiv m'_2$ proving the statement. □

## 4. Tree Automata with Costs

A *finite tree automaton* is a quadruple $M = (Q, \Sigma, \delta, F)$ where $Q$ and $F \subseteq Q$ are finite sets of *states* and *final states*, respectively, $\Sigma$ is a ranked alphabet, and $\delta \subseteq \bigcup_{k \in \mathbb{N}} \delta^{(k)}$ is a ranked alphabet of *transitions*, where for every integer $k \in \mathbb{N}$ we have $\delta^{(k)} \subseteq Q^k \times \Sigma^{(k)} \times Q$. In transitions we drop the tuple notation for the first component of the triple and just write $q_1 \ldots q_k$ to mean $(q_1, \ldots, q_k)$. A transition $(q_1 \ldots q_k, \sigma, q) \in \delta^{(k)}$ for some integer $k \in \mathbb{N}$, $k$-ary symbol $\sigma \in \Sigma^{(k)}$, and states $q_1, \ldots, q_k, q \in Q$ is also called *$q$-transition*. Moreover, let

$$\delta^q_{q_1 \ldots q_k} = \{\, (q_1 \ldots q_k, \sigma, q) \in \delta^{(k)} \mid \sigma \in \Sigma^{(k)} \,\} \quad \text{and} \quad \delta^q = \bigcup_{k \in \mathbb{N}, q_1, \ldots, q_k \in Q} \delta^q_{q_1 \ldots q_k}.$$

Let us now define the semantics of these devices. For every integer $n \in \mathbb{N}$, tree $s \in T_\Sigma(X_n)$, and states $q, q_1, \ldots, q_n \in Q$, we define the set $\Psi^q_{q_1 \ldots q_n}(s)$ of $(q_1 \ldots q_n, q)$-*computations* over $s$ as a subset of $T_\delta(X_n)$ by induction on the tree $s$ as follows.

(i) Let $s = x_j$ for some index $j \in [n]$. If $q = q_j$, then $\Psi^q_{q_1 \ldots q_n}(s) = \{x_j\}$, otherwise $\Psi^q_{q_1 \ldots q_n}(s) = \emptyset$.

(ii) Let $s = \sigma(s|_1, \ldots, s|_k)$ for some $k \in \mathbb{N}$ and input symbol $\sigma \in \Sigma^{(k)}$. Then $\Psi^q_{q_1 \ldots q_n}(s)$ is the set of all trees $\tau(\psi_1, \ldots, \psi_k)$, where for every index $j \in [k]$ there is a state $r_j \in Q$, such that $\psi_j \in \Psi^{r_j}_{q_1 \ldots q_n}(s|_j)$ and $\tau = (r_1 \ldots r_k, \sigma, q) \in \delta^{(k)}$.

The set of $(q_1 \ldots q_n, q)$-computations is defined as $\Psi^q_{q_1 \ldots q_n} = \bigcup_{s \in T_\Sigma(X_n)} \Psi^q_{q_1 \ldots q_n}(s)$ and the set of linear and non-deleting $(q_1 \ldots q_n, q)$-computations is defined as $\widehat{\Psi}^q_{q_1 \ldots q_n} = \bigcup_{s \in \widehat{T_\Sigma}(X_n)} \Psi^q_{q_1 \ldots q_n}(s)$. In case $n = 0$, we write $\varepsilon$ for the sequence $q_1 \ldots q_n$

and we speak about an $(\varepsilon, q)$-computation or just $q$-computation. The tree language *accepted by $M$ in state $q \in Q$* is defined to be

$$L(M)_q = \{\, s \in T_\Sigma \mid \Psi^q_\varepsilon(s) \neq \emptyset \,\}$$

and the tree language *accepted by $M$* is $L(M) = \bigcup_{q \in F} L(M)_q$.

The *trace graph* [31] of $M$ is the directed graph $G(M) = (Q, E)$, where the set $E$ of labeled edges consists of all triples $(q', \langle \tau, j\rangle, q)$ with $q, q' \in Q$ being states and the label $\langle \tau, j \rangle$ consists of a transition $\tau = (q_1 \dots q_k, \sigma, q) \in \delta^{(k)}$ for some integer $k \in \mathbb{N}$ and an integer $j \in \mathbb{N}$ such that $j \in [k]$ and $q' = q_j$. Let $q, q' \in Q$. We call $q$ *reachable* from $q'$ in $G(M)$, if $q = q'$ or there exists an integer $k \in \mathbb{N}_+$ and a sequence $(q_0, \langle \tau_1, j_1\rangle, q_1), (q_1, \langle \tau_2, j_2\rangle, q_2), \dots, (q_{k-1}, \langle \tau_k, j_k\rangle, q_k)$ of edges in $E$ such that $q_0 = q'$ and $q_k = q$. Moreover, we define the relation $\sim_M$ on $Q$ as follows: for two vertices $q, q' \in Q$ we have $q \sim_M q'$, if and only if both $q$ is reachable from $q'$ and $q'$ is reachable from $q$, i.e., if $q$ and $q'$ are strongly connected in $G(M)$. Then certainly $\sim_M$ is an equivalence relation on $Q$.

Now we define the relation $\leq_M$ over the factor set $[Q]_{\sim_M}$ as follows: for $q, q' \in Q$, we let $[q]_{\sim_M} \leq_M [q']_{\sim_M}$, if and only if $q'$ is reachable from $q$ in $G(M)$. Certainly, $\leq_M$ is a partial order on $[Q]_{\sim_M}$. Occasionally we will also use the partial order $\leq_M$ defined on states as follows. We let $q \leq_M q'$ for states $q, q' \in Q$, if and only if $[q]_{\sim_M} \leq_M [q']_{\sim_M}$ and $q \notin [q']_{\sim_M} \setminus \{q'\}$.

Given a tree automaton $M$ we can construct another tree automaton $M'$ with $L(M) = L(M')$ such that $M'$ has no useless states [31], i.e., states which do not occur in any $q$-computation of any final state $q \in F$.

Let us now add costs to the finite tree automaton $M = (Q, \Sigma, \delta, F)$ in the way as it was done in [31]. For this let us consider a semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$. A *cost function* for $M$ is a mapping $c : \delta \longrightarrow P(A, X)$ such that for every $k \in \mathbb{N}$ and $\tau \in \delta^{(k)}$ we have $c(\tau) \in P(A, X_k)$. The cost function $c$ is *linear*, if for every $\tau \in \delta$ the polynomial $c(\tau)$ is linear. We extend $c$ to a mapping $c : T_\delta(X) \longrightarrow P(A, X)$ in the following way. Let $\psi \in T_\delta(X)$. (i) If $\psi = x$ with $x \in X$, then $c(\psi) = x$ and, (ii) if $\psi = \tau(\psi|_1, \dots, \psi|_k)$ for some $k \in \mathbb{N}$ and $\tau \in \delta^{(k)}$, then $c(\psi) = c(\tau)[c(\psi|_1), \dots, c(\psi|_k)]$.

For a set $\Psi \subseteq \Psi^q_{q_1 \dots q_n}$ of $(q_1 \dots q_n, q)$-computations we introduce the set of costs in the usual manner, i.e., $c(\Psi) = \{\, c(\psi) \mid \psi \in \Psi \,\}$ and for every $n$ semiring elements $a_1, \dots, a_n \in A$, we let $c(\Psi)(a_1, \dots, a_n) = \{\, c(\psi)(a_1, \dots, a_n) \mid \psi \in \Psi \,\}$. Finally, we write $c(M)_q$ for $c(\Psi^q_\varepsilon)$ and define the set of *accepting costs* as

$$c(M) = \bigcup_{q \in F} c(M)_q.$$

**Example 22.** *Let $M_E = (Q, \Sigma, \delta_E, F)$ be the tree automaton with $Q = \{q_0, q_1, q, r\}$, input ranked alphabet $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}\}$, final states $F = \{q_1, r\}$, and the following set $\delta_E$ of transitions.*

$$\delta_E = \{\underbrace{(\varepsilon, \alpha, q_0)}_{\tau_1}, \underbrace{(\varepsilon, \alpha, q)}_{\tau_2}, \underbrace{(q_0 q_0, \sigma, q_0)}_{\tau_3}, \underbrace{(q_0 q, \sigma, q_1)}_{\tau_4}, \underbrace{(q_0 q, \sigma, q)}_{\tau_5}, \underbrace{(q q_1, \sigma, r)}_{\tau_6}, \underbrace{(r r, \sigma, r)}_{\tau_7}\}$$

*The tree language accepted by $M_E$ in state $q_1$ is $L(M_E)_{q_1} = \{\, \sigma(s_1, s_2) \mid s_1, s_2 \in T_\Sigma \,\}$,* *and since* $\alpha \notin L(M_E)_r$*, the tree language accepted by* $M_E$ *is*

$$L(M_E) = \{\, \sigma(s_1, s_2) \mid s_1, s_2 \in T_\Sigma \,\}.$$

*Now we add the cost function* $c_E : \delta_E \longrightarrow P(\mathbb{N}, X)$ *over the semiring* Nat *specified by*

$$\begin{array}{llll} c_E(\tau_1) = 0 & c_E(\tau_2) = 2 & c_E(\tau_3) = 3x_1 + 4x_2 & c_E(\tau_4) = 3x_1x_2 \\ c_E(\tau_5) = 2x_1 + x_2 & c_E(\tau_6) = 5x_1 & c_E(\tau_7) = x_1 + x_2. & \end{array}$$

*The set of costs computed by states* $q_0$ *and* $q_1$ *is*

$$[c_E(M_E)_{q_0}]_\equiv = [c_E(M_E)_{q_1}]_\equiv = \{[0]_\equiv\}.$$

*Moreover,* $[c_E(M_E)_q]_\equiv = \{[2]_\equiv\}$ *and*

$$[c_E(M_E)_r]_\equiv = \{\, [n]_\equiv \mid n \in \mathbb{N}_+,\ (\exists k \in \mathbb{N}_+) : \ 10 \cdot k = n \,\}.$$

*Consequently,* $[c_E(M_E)]_\equiv = \{[0]_\equiv, [10]_\equiv, [20]_\equiv, [30]_\equiv, \ldots\}$*. Finally, Figure 1 graphically displays* $M_E$ *and Figure 2 displays an input tree, one possible computation tree for that input tree, the cost function applied to this computation tree, and the trace graph* $G(M_E)$*.*

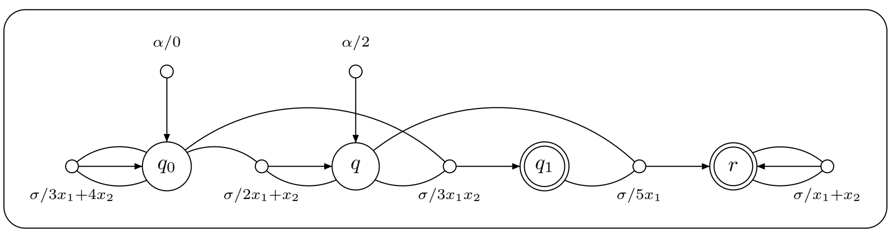


Figure 1: Example tree automaton with cost function of Example 22.

In several proofs of the present paper we use the technique of decomposing a $q$-computation $\psi$ into a $q'$-computation $\psi_2$ and a $(q', q)$-computation $\psi_1$. The following observation shows how this decomposition is reflected in the costs. A straightforward inductive proof on the length of the path in the computation $\psi_1$ from the root to the node labeled with the variable $x_1$ shows the following observation. For the rest of this section, let $M = (Q, \Sigma, \delta, F)$ be a tree automaton with cost function $c : \delta \longrightarrow P(A, X)$ over a semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$.

**Observation 23.** *Let* $q, q' \in Q$ *be states,* $\psi \in \Psi^q_\varepsilon$ *be a* $q$*-computation,* $\psi_2 \in \Psi^{q'}_\varepsilon$ *be a* $q'$*-computation, and* $\psi_1 \in \Psi^q_{q'}$ *be a* $(q', q)$*-computation such that* $\psi = \psi_1[\psi_2]$*. Then* $c(\psi) = c(\psi_1)[c(\psi_2)]$*.*

**Definition 24.** *For every subset* $E \subseteq A$ *we define the set* $Q_E \subseteq Q$ *of* $E$-states *of* $M$ *to be*

$$Q_E = \{\, q \in Q \mid (\forall \psi \in \Psi^q_\varepsilon)(\exists e \in E) : \ c(\psi) \equiv e \,\}.$$

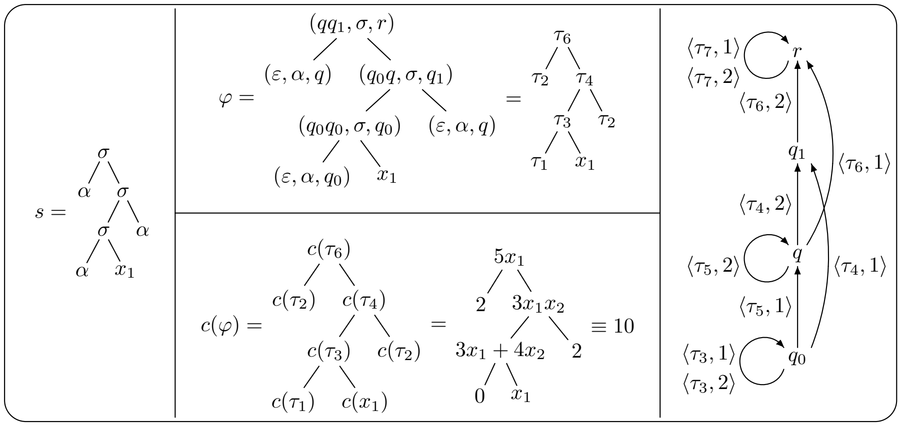

Figure 2: A sample input tree $s$, computation tree $\varphi$, and the trace graph $G(M_E)$.

Clearly, $E \subseteq E' \subseteq A$ implies $Q_E \subseteq Q_{E'} \subseteq Q_A = Q$. Moreover, $Q_\emptyset = \emptyset$ if $M$ has no useless states. If we reconsider the tree automaton $M_E = (Q, \Sigma, \delta_E, F)$ with cost function $c_E : \delta_E \longrightarrow P(\mathbb{N}, X)$ of Example 22, then $Q_{\{0\}} = \{q_0, q_1\}$, $Q_{\{2\}} = \{q\}$, and $Q_{\{10,20,30,\ldots\}} = \{r\}$, for example. Next we introduce a shorthand for a simple case analysis. For every set $S$ let

$$e_S, e'_S : (X \cup A) \times S \times \mathcal{P}(S) \longrightarrow X \cup A$$

be the mappings specified for every $z \in X \cup A$, element $s \in S$, and subset $S' \subseteq S$ by

$$e_S(z, s, S') = \begin{cases} z & \text{, if } s \in S' \\ \mathbf{0} & \text{, otherwise} \end{cases} \qquad e'_S(z, s, S') = \begin{cases} z & \text{, if } s \in S' \\ \mathbf{1} & \text{, otherwise} \end{cases}.$$

Now we give an algorithm that computes the set of $\{\mathbf{0}\}$-states.

**Lemma 25.** *Let $\mathcal{A}$ be positive. The set $Q_{\{\mathbf{0}\}}$ of all $\{\mathbf{0}\}$-states can effectively be computed.*

*Proof.* We give an algorithm which computes $Q_{\{\mathbf{0}\}}$ in Algorithm 1. Upon termination of the algorithm, i.e., $Q_n = Q_{n-1}$, it is easy to see that for every $q \in Q$, it holds that $q \in Q_n$, if and only if there is a tree $s \in L(M)_q$ and a computation $\psi \in \Psi^q_\varepsilon(s)$ such that $c(\psi) \not\equiv \mathbf{0}$. Thus $Q_{\{\mathbf{0}\}} = Q \setminus Q_n$. □

For every integer $k \in \mathbb{N}$ and $(q_1, \ldots, q_k) \in Q^k$ we let

$$\begin{aligned} \theta^Q_{q_1 \ldots q_k} &= [e_Q(x_1, q_1, Q \setminus Q_{\{\mathbf{0}\}}), \ldots, e_Q(x_k, q_k, Q \setminus Q_{\{\mathbf{0}\}})] \\ \eta^Q_{q_1 \ldots q_k} &= \theta^Q_{q_1 \ldots q_k} [e'_Q(x_1, q_1, Q \setminus Q_{\{\mathbf{1}\}}), \ldots, e'_Q(x_k, q_k, Q \setminus Q_{\{\mathbf{1}\}})]. \end{aligned}$$

be shorthands for substitutions. We can likewise compute the sets of $\{\mathbf{1}\}$-states and $\{\mathbf{0}, \mathbf{1}\}$-states. The algorithms are presented in Algorithms 2 and 3, respectively.

**Algorithm 1** An algorithm computing the set of $\{\mathbf{0}\}$-states of $M$.

**Require:** $M$ has no useless states, $\mathcal{A}$ is positive

$n := 0$, $Q_0 := \emptyset$

**repeat**

$$Q_{n+1} := Q_n \cup \left\{ q \in Q \;\middle|\; \begin{array}{l} (\exists k \in \mathbb{N})(\exists \sigma \in \Sigma^{(k)})(\exists q_1, \dots, q_k \in Q)\colon \\ \quad \tau = (q_1 \dots q_k, \sigma, q) \in \delta^{(k)}, \\ \quad c(\tau)[e_Q(x_1, q_1, Q_n), \dots, e_Q(x_k, q_k, Q_n)] \not\equiv \mathbf{0} \end{array} \right\}$$

$n := n + 1$

**until** $Q_n = Q_{n-1}$

**Ensure:** $Q_{\{\mathbf{0}\}} = Q \setminus Q_n$

**Algorithm 2** An algorithm computing the set of $\{\mathbf{1}\}$-states of $M$.

**Require:** $M$ has no useless states, $\mathcal{A}$ is positive, one-summand free, one-product free

$n := 0$, $Q_0 := \emptyset$

**repeat**

$$Q_{n+1} := Q_n \cup \left\{ q \in Q \;\middle|\; \begin{array}{l} (\exists k \in \mathbb{N})(\exists \sigma \in \Sigma^{(k)})(\exists q_1, \dots, q_k \in Q)\colon \\ \quad \tau = (q_1 \dots q_k, \sigma, q) \in \delta^{(k)}, \\ \quad c(\tau)\, \theta^{Q}_{q_1 \dots q_k} \,[e'_Q(x_1, q_1, Q_n), \dots, e'_Q(x_k, q_k, Q_n)] \not\equiv \mathbf{1} \end{array} \right\}$$

$n := n + 1$

**until** $Q_n = Q_{n-1}$

**Ensure:** $Q_{\{\mathbf{1}\}} = Q \setminus Q_n$

**Algorithm 3** An algorithm computing the set of $\{\mathbf{0}, \mathbf{1}\}$-states of $M$.

**Require:** $M$ has no useless states, $\mathcal{A}$ is positive, one-summand free, one-product free

$n := 0$, $Q_0 := \emptyset$

**repeat**

$$Q_{n+1} := Q_n \cup \left\{ q \in Q \;\middle|\; \begin{array}{l} (\exists k \in \mathbb{N})(\exists \sigma \in \Sigma^{(k)})(\exists q_1, \dots, q_k \in Q)\colon \\ \quad \tau = (q_1 \dots q_k, \sigma, q) \in \delta^{(k)}, \\ \quad c(\tau)\, \eta^{Q}_{q_1 \dots q_k} \,[e'_Q(x_1, q_1, Q_n), \dots, \\ \qquad\qquad\qquad e'_Q(x_k, q_k, Q_n)] \notin [\mathbf{0}]_\equiv \cup [\mathbf{1}]_\equiv \end{array} \right\}$$

$n := n + 1$

**until** $Q_n = Q_{n-1}$

**Ensure:** $Q_{\{\mathbf{0},\mathbf{1}\}} = Q \setminus Q_n$

If we apply Algorithm 1 to the tree automaton $M_E$ of Example 22, then the algorithm terminates with final values $n = 3$, $Q_0 = \emptyset$, $Q_1 = \{q\}$, $Q_2 = \{q, r\}$, and $Q_3 = Q_2$. Hence we computed $Q_{\{0\}} = Q \setminus Q_3 = \{q_0, q_1\}$ which we already stated in the previous paragraph. Following up we present a notion of reducedness for tree automata with cost functions (cf. the concept of $E$-parameter reduced tree automata with costs in [31]). Similar to the removal of useless states, we now remove unnecessary variables from the cost polynomials.

**Definition 26.** *$M$ is called* reduced *[31], if the following conditions are satisfied. There exists a distinguished state $\perp \in Q$ such that*

*(i) $M$ has no useless states or the set of useless states is $\{\perp\}$,*

*(ii) $\perp$ is the only zero-state, i.e., $Q_{\{\mathbf{0}\}} = \{\perp\}$,*

*(iii) for every integer $k \in \mathbb{N}$, transition $\tau = (q_1 \dots q_k, \sigma, q) \in \delta^{(k)}$ for some states $q, q_1, \dots, q_k \in Q$, and symbol $\sigma \in \Sigma^{(k)}$, if $q \in Q_{\{\mathbf{0},\mathbf{1}\}}$ then we have $q_j = \perp$ for every index $j \in [k]$ and $c(\tau) = e_Q(\mathbf{1}, q, Q \setminus \{\perp\})$.*

*(iv) For every integer $k \in \mathbb{N}$, transition $\tau = (q_1 \dots q_k, \sigma, q) \in \delta^{(k)}$ for some states $q, q_1, \dots, q_k \in Q$, and input symbol $\sigma \in \Sigma^{(k)}$,*

- *either $c(\tau) = \mathbf{0}$ and $q_1, \dots, q_k \in Q_{\{\mathbf{0},\mathbf{1}\}}$,*
- *or $c(\tau) \in P(A_+, X_k)$ is a zero-free cost polynomial and we demand for every index $j \in [k]$ that*

$$x_j \in \mathrm{var}(c(\tau)) \qquad \Longleftrightarrow \qquad q_j \notin Q_{\{\mathbf{0},\mathbf{1}\}}.$$

Apparently the tree automaton $M_E$ with cost function of Example 22 is not reduced because $Q_{\{0\}} = \{q_0, q_1\}$ is not a singleton. Next we prove that, under certain conditions, for every tree automaton $M$ with cost function $c$, a reduced tree automaton $M'$ with cost-function $c'$ can be constructed such that $c(M) \equiv c'(M')$, i.e., $M$ and $M'$ are cost-equivalent (cf. Theorem 2.3 of [31]). Whereas the removal of useless states preserves the tree language accepted by a tree automaton, our construction of reducing a tree automaton does not preserve the tree language accepted but rather the set of accepting costs.

**Lemma 27.** *Let $\mathcal{A}$ be positive, one-summand free, and one-product free. Then a reduced tree automaton $M' = (Q', \Sigma, \delta', F')$ with cost function $c' : \delta' \longrightarrow P(A, X)$ can effectively be constructed such that $M$ and $M'$ are* cost-equivalent*, i.e., we have that $c(M) \equiv c'(M')$.*

*Proof.* We recall that the sets $Q_{\{\mathbf{0}\}}$, $Q_{\{\mathbf{1}\}}$, and $Q_{\{\mathbf{0},\mathbf{1}\}}$ can effectively be computed due to Algorithms 1, 2, and 3. Firstly we construct the tree automaton $M'' = (Q'', \Sigma, \delta'', F'')$ with cost function $c'' : \delta'' \longrightarrow P(A, X)$ as follows. We tag the states of the original automaton with $\{1, 2, 3\}$ where roughly speaking $Q \times \{1\}$ corresponds to the original states, $Q \times \{2\}$ corresponds to a copy of $Q$ preparing the cost $\mathbf{1}$, and $Q \times \{3\}$ corresponds to a copy of $Q$ preparing cost $\mathbf{0}$. Note that we will assign cost $\mathbf{1}$ to states $(q, 3)$ for $q \in Q$, however these states will never be final states, so that this will not influence the set of accepting costs. Let $Q'' = Q \times [3] \cup \{\bot\}$, where $\bot \notin Q$ is a new distinguished state and

$$F'' = \begin{cases} F \times [1] & \text{, if } F \cap Q_{\{\mathbf{0},\mathbf{1}\}} = \emptyset \\ F \times [2] & \text{, if } F \cap (Q_{\{\mathbf{0},\mathbf{1}\}} \setminus Q_{\{\mathbf{1}\}}) = \emptyset, F \cap Q_{\{\mathbf{1}\}} \neq \emptyset \\ (F \setminus Q_{\{\mathbf{0}\}}) \times [1] \cup \{\bot\} & \text{, if } F \cap (Q_{\{\mathbf{0},\mathbf{1}\}} \setminus Q_{\{\mathbf{0}\}}) = \emptyset, F \cap Q_{\{\mathbf{0}\}} \neq \emptyset \\ (F \setminus Q_{\{\mathbf{0}\}}) \times [2] \cup \{\bot\} & \text{, if } F \cap (Q_{\{\mathbf{0},\mathbf{1}\}} \setminus Q_{\{\mathbf{1}\}}) \neq \emptyset, \\ & \quad \text{and } F \cap (Q_{\{\mathbf{0},\mathbf{1}\}} \setminus Q_{\{\mathbf{0}\}}) \neq \emptyset. \end{cases}$$

Moreover, the set of transitions $\delta''$ and the cost function $c'' : \delta'' \longrightarrow P(A, X)$ are defined in the following way.

(i) For every $k \in \mathbb{N}$ and $\sigma \in \Sigma^{(k)}$ let $\tau'' = (\bot \ldots \bot, \sigma, \bot) \in (\delta'')^{(k)}$ with $c''(\tau'') = \mathbf{0}$.

(ii) For every $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$, and $q \in Q \times \{2, 3\}$ let $\tau'' = (\bot \ldots \bot, \sigma, q) \in (\delta'')^{(k)}$ with $c''(\tau'') = \mathbf{1}$.

(iii) For every $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$, $q, q_1, \ldots, q_k \in Q$, $\tau = (q_1 \ldots q_k, \sigma, q) \in \delta^{(k)}$ with $q \notin Q_{\{\mathbf{0},\mathbf{1}\}}$ we define the set of transitions as follows. For every partition $\{q_1, \ldots, q_k\} = Q_0 \cup Q_1 \cup Q_2$ into pairwise disjoint sets, i.e., for every $i, j \in [0, 2]$ we have $Q_i \cap Q_j = \emptyset$, if $i \neq j$, with the properties that

- $Q_{\{\mathbf{0}\}} \cap \{q_1, \ldots, q_k\} \subseteq Q_0$,
- $Q_{\{\mathbf{1}\}} \cap \{q_1, \ldots, q_k\} \subseteq Q_1$, and
- $\{q_1, \ldots, q_k\} \setminus Q_{\{\mathbf{0},\mathbf{1}\}} = Q_2$,

let

$$\theta = [e_Q(x_1, q_1, Q \setminus Q_0), \ldots, e_Q(x_k, q_k, Q \setminus Q_0)] \\ [e'_Q(x_1, q_1, Q \setminus Q_1), \ldots, e'_Q(x_k, q_k, Q \setminus Q_1)].$$

If $c(\tau)\,\theta \equiv \mathbf{0}$, which is decidable by Lemma 11, then

$$\tau'' = ((q_1, 3) \ldots (q_k, 3), \sigma, (q, 1)) \in \delta''^{(k)} \quad \text{and} \quad c''(\tau'') = \mathbf{0}.$$

Otherwise, by Lemma 7, construct a zero-free polynomial $p \in P(A_+, X_k)$ such that $p \equiv c(\tau)\,\theta$. Moreover, let $\tau'' = (r_1 \ldots r_k, \sigma, (q, 1)) \in \delta''^{(k)}$ and $c''(\tau'') = p$, where for every index $j \in [k]$ we have

$$r_j = \begin{cases} (q_j, 1) & \text{, if } x_j \in \text{var}(p) \\ (q_j, 2) & \text{, if } x_j \notin \text{var}(p),\ q_j \in Q_1 \\ (q_j, 3) & \text{, otherwise} \end{cases}.$$

(iv) The set of transitions $\delta''$ has no additional elements, i.e., every element is required by either (i), (ii), or (iii).

Then construct the tree automaton $M'$ with cost function $c' : \delta' \longrightarrow P(A, X)$ from $M''$ and $c''$ by removing useless states $q \in Q'' \setminus \{\bot\}$ and transitions, and let the cost function $c'$ be defined by $c'(\tau') = c''(\tau')$ for every transition $\tau' \in \delta'$.

First we show that $M'$ is reduced. Obviously, $M'$ does not have useless states $q \in Q' \setminus \{\bot\}$. Trivially $Q'_{\{\mathbf{0}\}} = \{\bot\}$ (cf. Condition (ii) of Definition 26), $Q'_{\{\mathbf{1}\}} = \{\, (q,j) \in Q' \mid j \in \{2,3\}\,\}$ and $Q'_{\{\mathbf{0},\mathbf{1}\}} = Q'_{\{\mathbf{0}\}} \cup Q'_{\{\mathbf{1}\}}$, because $\mathcal{A}$ is positive. For every integer $k \in \mathbb{N}$, input symbol $\sigma \in \Sigma^{(k)}$, $k+1$ states $q, q_1, \dots, q_k \in Q'$, and transition $\tau' = (q_1 \dots q_k, \sigma, q) \in (\delta')^{(k)}$ we have the following. If $q \in Q'_{\{\mathbf{0},\mathbf{1}\}}$, then $q_1 = \cdots = q_k = \bot$ and $c'(\tau') = e_{Q'}(\mathbf{1}, q, Q' \setminus \{\bot\})$ by Items (i), (ii), and (iv), thus fulfilling Condition (iii) of Definition 26. On the other hand, if $q \notin Q'_{\{\mathbf{0},\mathbf{1}\}}$, then for every index $j \in [k]$ the variable $x_j$ is in $\mathrm{var}(c'(\tau'))$, if and only if $q_j \notin Q_{\{\mathbf{0},\mathbf{1}\}}$ by Items (iii) and (iv). This saturates Condition (iv) of Definition 26, so all conditions of Definition 26 are met; hence $M'$ is reduced.

Next we show that $c(M) \equiv c'(M')$. We show this by proving that, for every $d \in [A]_\equiv$,

$$
\begin{aligned}
& d \in [c(M)]_\equiv \\
\iff{} & \text{there are a } q \in F \text{ and a } q\text{-computation } \psi \in \Psi^q_\varepsilon \text{ (using } M\text{)} \\
& \quad \text{such that } c(\psi) \equiv d \\
\overset{\dagger}{\iff}{} & \text{there are a } q' \in F' \text{ and a } q'\text{-computation } \psi' \in \Psi^{q'}_\varepsilon \text{ (using } M'\text{)} \\
& \quad \text{such that } c'(\psi') \equiv d \\
\iff{} & d \in [c'(M')]_\equiv .
\end{aligned}
$$

It only remains to prove the equivalence marked by †.

Part $\overset{\dagger}{\Rightarrow}$: We distinguish two cases.

Case 1: Let $q \in F \cap Q_{\{\mathbf{0},\mathbf{1}\}}$. Now, by Definition 24, $d \in [\mathbf{0}]_\equiv \cup [\mathbf{1}]_\equiv$, hence we have either $d \equiv \mathbf{1}$ or $d \equiv \mathbf{0}$. If $d \equiv \mathbf{0}$, then we have $\bot \in F'$ and, moreover, there is a $\bot$-computation $\psi' \in \Psi^\bot_\varepsilon$ using $M'$ with $c'(\psi') \equiv \mathbf{0}$. Consequently, $c'(\psi') \equiv d$ and setting $q' = \bot$ proves the statement in case $d \equiv \mathbf{0}$.

On the other hand, if $d \equiv \mathbf{1}$, then $(q,2) \in F'$ and there is a $(q,2)$-computation $\psi' \in \Psi^{(q,2)}_\varepsilon$ using $M'$ with $c'(\psi') \equiv \mathbf{1}$. Consequently, $c'(\psi') \equiv d$ and setting $q' = (q,2)$ proves the statement, if $d \equiv \mathbf{1}$.

Case 2: $q \in F \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$. By assumption there is a $q$-computation $\psi \in \Psi^q_\varepsilon$ using $M$ with $c(\psi) \equiv d$. By induction over the height of the computation $\psi$ we prove that there is a $(q,1)$-computation $\psi' \in \Psi^{(q,1)}_\varepsilon$ using $M'$ with $c'(\psi') \equiv d$.

**Induction base:** Let $\mathrm{height}(\psi) = 1$. Then there exist a nullary input symbol $\alpha \in \Sigma^{(0)}$ and a transition $\tau = (\varepsilon, \alpha, q) \in \delta^{(0)}$ such that $c(\tau) \equiv d$. According to Item (iii) also $\tau' = (\varepsilon, \alpha, (q,1)) \in (\delta')^{(0)}$ and $c'(\tau') = c(\tau)$. Hence $c'(\tau') \equiv d$ and setting $\psi' = \tau'$ proves the statement.

**Induction step:** Let $\mathrm{height}(\psi) > 1$. Then there exist an integer $k \in \mathbb{N}$, an input symbol $\sigma \in \Sigma^{(k)}$, states $q_1, \dots, q_k \in Q$, a transition $\tau = (q_1 \dots q_k, \sigma, q) \in \delta^{(k)}$, and $q_j$-subcomputations $\psi_j \in \Psi^{q_j}_\varepsilon$ for every index $j \in [k]$ such that $\psi = \tau(\psi_1, \dots, \psi_k)$. We construct the partition $\{q_1, \dots, q_k\} = Q_0 \cup Q_1 \cup Q_2$ as follows. For every index $j \in [k]$

- if $q_j \in Q_{\{\mathbf{0}\}}$ then $q_j \in Q_0$.
- if $q_j \in Q_{\{\mathbf{1}\}}$ then $q_j \in Q_1$.
- if $q_j \in Q_{\{\mathbf{0},\mathbf{1}\}}$ and $c(\psi_j) \equiv \mathbf{0}$ then $q_j \in Q_0$.
- if $q_j \in Q_{\{\mathbf{0},\mathbf{1}\}}$ and $c(\psi_j) \equiv \mathbf{1}$ then $q_j \in Q_1$.
- if $q_j \notin Q_{\{\mathbf{0},\mathbf{1}\}}$ then $q_j \in Q_2$.

This partition certainly fulfills the restrictions imposed in Item (iii). Let

$$\begin{aligned}\theta = [&e_Q(x_1, q_1, Q \setminus Q_0), \ldots, e_Q(x_k, q_k, Q \setminus Q_0)] \\ &[e'_Q(x_1, q_1, Q \setminus Q_1), \ldots, e'_Q(x_k, q_k, Q \setminus Q_1)].\end{aligned}$$

It should be clear that $c(\tau)\,\theta\,[c(\psi_1), \ldots, c(\psi_k)] \equiv c(\tau)[c(\psi_1), \ldots, c(\psi_k)] = c(\psi) \equiv d$. If $c(\tau)\,\theta \equiv \mathbf{0}$, then $d \equiv \mathbf{0}$. Moreover, by Item (iii), there exists a transition $\tau' = ((q_1, 3) \ldots (q_k, 3), \sigma, (q, 1)) \in \delta'^{(k)}$ with $c'(\tau') = \mathbf{0}$. Thus there certainly exists a $(q, 1)$-computation $\psi' \in \Psi_\varepsilon^{(q,1)}$ with $c'(\psi') \equiv d$.

If $c(\tau)\,\theta \not\equiv \mathbf{0}$, then, by Lemma 7, there exists a zero-free $p \in P(A_+, X_k)$ with $p \equiv c(\tau)\,\theta$ such that, by Item (iii), there exists a transition

$$\tau' = (r_1 \ldots r_k, \sigma, (q, 1)) \in \delta'^{(k)} \quad \text{with} \quad c'(\tau') = p$$

where for every index $j \in [k]$

$$r_j = \begin{cases} (q_j, 1) & \text{, if } x_j \in \mathrm{var}(p) \\ (q_j, 2) & \text{, if } x_j \notin \mathrm{var}(p),\ q_j \in Q_1 \\ (q_j, 3) & \text{, otherwise} \end{cases}.$$

Moreover, for every index $j \in [k]$, if $r_j = (q_j, 1)$, then there also exists a $(q_j, 1)$-computation $\psi'_j \in \Psi_\varepsilon^{(q_j,1)}$ (using $M'$) with cost $c'(\psi'_j) \equiv c(\psi_j)$ by induction hypothesis. Otherwise $r_j \in Q'_{\{\mathbf{1}\}}$ and according to Items (i) and (ii), $x_j \notin \mathrm{var}(c(\tau))$ or either $c(\psi_j) \equiv \mathbf{0}$ and there is a $(q_j, 3)$-computation $\psi'_j \in \Psi_\varepsilon^{(q_j,3)}$ (using $M'$) with cost $c'(\psi'_j) \equiv \mathbf{1}$, or $c(\psi_j) \equiv \mathbf{1}$ and there exists a $(q_j, 2)$-computation $\psi'_j \in \Psi_\varepsilon^{(q_j,2)}$ (using $M'$) with cost $c'(\psi'_j) \equiv \mathbf{1}$.

We consider $p = c'(\tau') \equiv c(\tau)\,\theta$. Let $j \in [k]$. Apparently, $c(\psi_j) \equiv \mathbf{0}$, if $q_j \in Q_0$, and $c(\psi_j) \equiv \mathbf{1}$, if $q_j \in Q_1$ by Definition 24. Obviously,

$$p[c(\psi_1), \ldots, c(\psi_k)] \equiv c(\tau)[c(\psi_1), \ldots, c(\psi_k)].$$

Furthermore, if a variable $x_j$ for some index $j \in [k]$ obeys $x_j \notin \mathrm{var}(c'(\tau'))$, then the cost $c(\psi_j)$ of the corresponding subcomputation can be set to an arbitrary value. Thus, by the Replacement Theorem (cf. Theorem 4) and the condition that $c(\psi_j) \equiv c'(\psi'_j)$ for every $j \in [k]$ such that $x_j \in \mathrm{var}(c'(\tau'))$, we have

$$c(\tau)[c(\psi_1), \ldots, c(\psi_k)] \equiv c'(\tau')[c(\psi_1), \ldots, c(\psi_k)] \equiv c'(\tau')[c'(\psi'_1), \ldots, c'(\psi'_k)].$$

Finally, the induction step, and thereby, the first part are proved.

Part $\overset{\dagger}{\Leftarrow}$: The proof of this direction is similar to the previous one. □

If we apply the construction present in the previous proof to our example tree automaton $M_E = (Q, \Sigma, \delta_E, F)$ with cost function $c_E : \delta_E \longrightarrow P(\mathbb{N}, X)$ of Example 22, then we might obtain the reduced tree automaton $M'_E = (Q', \Sigma, \delta'_E, F')$ (displayed in Figure 3) with cost function $c'_E : \delta'_E \longrightarrow P(\mathbb{N}, X)$ with

$$\begin{aligned}
Q' &= \{\bot, (q_0, 3), (r, 2), (q, 1), (r, 1), (q_1, 3)\},\\
F' &= \{\bot, (r, 1), (r, 2)\}
\end{aligned}$$

and transitions

$$\begin{aligned}
\delta' = \{&(\varepsilon, \alpha, \bot), (\bot\bot, \sigma, \bot), (\varepsilon, \alpha, (q_0, 3)), (\bot\bot, \sigma, (q_0, 3)), (\varepsilon, \alpha, (r, 2)), (\bot\bot, \sigma, (r, 2)),\\
&(\varepsilon, \alpha, (q_1, 3)), (\bot\bot, \sigma, (q_1, 3)), (\varepsilon, \alpha, (q, 1)), ((q, 1)(q_1, 3), \sigma, (r, 1)),\\
&((r, 1)(r, 1), \sigma, (r, 1)), ((q_0, 3)(q, 1), \sigma, (q, 1))\}.
\end{aligned}$$

Further, the cost function $c'_E$ is specified by (note that we omitted the outermost parentheses for brevity)

$$\begin{aligned}
0 &= c'_E(\varepsilon, \alpha, \bot) = c'_E(\bot\bot, \sigma, \bot)\\
1 &= c'_E(\varepsilon, \alpha, (q_0, 3)) = c'_E(\bot\bot, \sigma, (q_0, 3)) = c'_E(\varepsilon, \alpha, (r, 2)) = c'_E(\bot\bot, \sigma, (r, 2))\\
1 &= c'_E(\varepsilon, \alpha, (q_1, 3)) = c'_E(\bot\bot, \sigma, (q_1, 3))\\
2 &= c'_E(\varepsilon, \alpha, (q, 1))\\
5 \cdot x_1 &= c'_E((q, 1)(q_1, 3), \sigma, (r, 1))\\
x_1 + x_2 &= c'_E((r, 1)(r, 1), \sigma, (r, 1))\\
x_2 &= c'_E((q_0, 3)(q, 1), \sigma, (q, 1)).
\end{aligned}$$

Next we show the main beneficial property of a reduced tree automaton with cost function. Roughly speaking, it states that the existence of a $(q', q)$-computation implies the existence of a $(q', q)$-computation $\psi$ such that $c(\psi)$ is zero-free and $x_1 \in \mathrm{var}(c(\psi))$. For this we need the following preparatory lemma, which also shows an interesting property of a reduced tree automaton.

**Lemma 28.** *Let $M$ be reduced and $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$. There exists a computation $\psi \in \Psi^q_\varepsilon$ such that $c(\psi)$ is zero-free.*

*Proof.* Since $M$ is reduced, it has no useless states. Hence there exists a computation $\psi \in \Psi^q_\varepsilon$. The property can then be proved by an easy induction on the height of $\psi$. □

**Lemma 29.** *Let $M$ be reduced. For every two states $q, q' \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$, if $\widehat{\Psi}^q_{q'} \neq \emptyset$, then there also exists a $(q', q)$-computation $\psi \in \widehat{\Psi}^q_{q'}$ such that $c(\psi)$ is zero-free and $x_1 \in \mathrm{var}(c(\psi))$.*

*Proof.* Let $\varphi \in \widehat{\Psi}^q_{q'}$ be arbitrary. We prove the statement by induction on the length $l = |w|$ where $w \in \mathrm{pos}(\varphi)$ is the unique position such that $\mathrm{lab}_\varphi(w) = x_1$.

**Induction base:** Assume that $l = 0$; thus $\varphi = x_1$ and $q' = q$ in order for $\varphi \in \widehat{\Psi}^q_{q'}$.

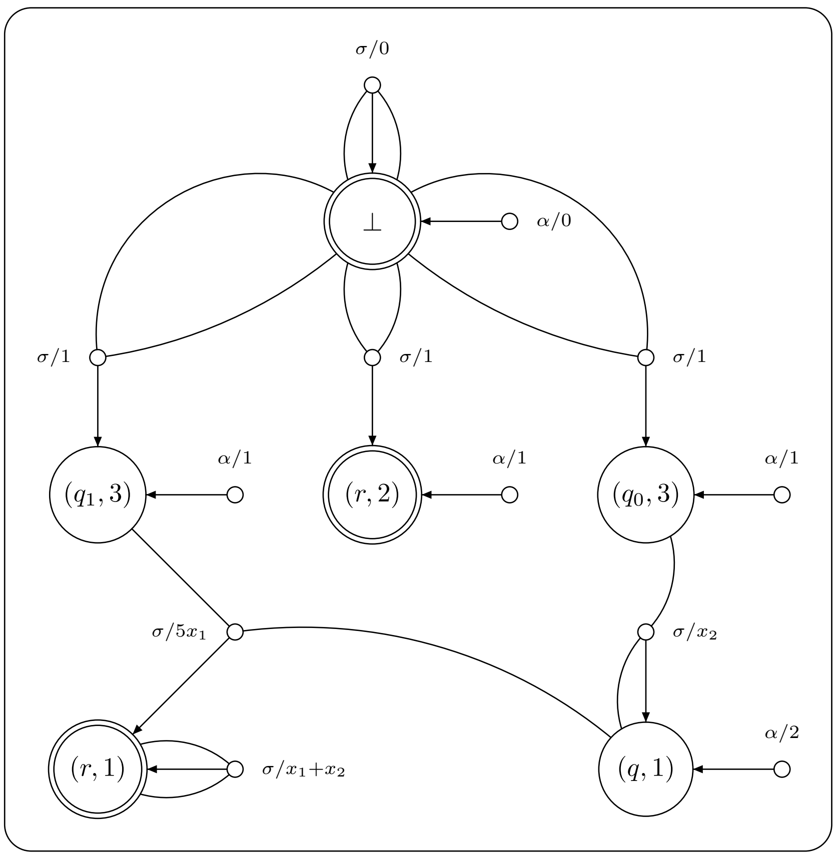


Figure 3: Example of a reduced tree automaton with cost function.

This computation $\varphi$ is appropriate, since $c(\varphi) = c(x_1) = x_1$ and certainly $x_1$ is zero-free and $x_1 \in \mathrm{var}(x_1)$. Thus set $\psi = \varphi$.

**Induction step:** Assume that $l = l'+1$ for some $l' \in \mathbb{N}$. Thus for some integer $k \in \mathbb{N}$, $k$-ary input symbol $\sigma \in \Sigma^{(k)}$, and states $q_1, \ldots, q_k \in Q$ we have $\varphi = \tau(\varphi|_1, \ldots, \varphi|_k)$, where $\tau = (q_1 \ldots q_k, \sigma, q) \in \delta^{(k)}$. Moreover, there is an index $i \in [k]$ such that $\varphi|_i \in \widehat{\Psi}^{q_i}_{q'}$ and for every $j \in [k]$ with $j \neq i$ we have $\varphi|_j \in \Psi^{q_j}_{\varepsilon}$.

Firstly, we observe $q_i \notin Q_{\{\mathbf{0},\mathbf{1}\}}$. In fact, if $q_i \in Q_{\{\mathbf{0},\mathbf{1}\}}$ then by a straightforward induction using Condition (iii) of Definition 26 we gain $q' = \perp$, hence $q' \in Q_{\{\mathbf{0},\mathbf{1}\}}$, which contradicts to the assumption $q' \notin Q_{\{\mathbf{0},\mathbf{1}\}}$. Hence by Condition (iv) of Definition 26, $c(\tau)$ is zero-free and $x_i \in \mathrm{var}(c(\tau))$. Moreover, by the induction hypothesis there exists a $(q', q_i)$-computation $\psi_i \in \widehat{\Psi}^{q_i}_{q'}$ such that $c(\psi_i)$ is zero-free and $x_1 \in \mathrm{var}(c(\psi_i))$.

Finally, for every index $j \in [k]$ with $j \neq i$, there are two cases. (i) Either $x_j \in \mathrm{var}(c(\tau))$, then by Condition (iv) in Definition 26 we have $q_j \notin Q_{\{\mathbf{0},\mathbf{1}\}}$. Then, by Lemma 28, there is a zero-free computation $\psi_j \in \Psi^{q_j}_{\varepsilon}$. (ii) On the other hand $x_j \notin \mathrm{var}(c(\tau))$, then we can take any arbitrary computation, e.g, we set $\psi_j = \varphi_j$. Now we consider the $(q', q)$-computation $\psi = \tau(\psi_1, \ldots, \psi_k)$. It should be clear that $c(\psi)$ is zero-free and $x_1 \in \mathrm{var}(c(\psi))$. $\square$

## 5. Characterizing and Deciding Cost-Finiteness

In this section we investigate the cost-finiteness of reduced tree automata with cost function where the underlying semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ is finitely factorizing and monotonic. We show that such a tree automaton $M = (Q, \Sigma, \delta, F)$ having cost function $c : \delta \longrightarrow P(A, X)$ is cost-finite, i.e., $[c(M)]_{\equiv}$ is finite, if and only if for every integer $k \in \mathbb{N}_+$, states $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$ and $q_1, \ldots, q_k \in Q$, index $i \in [k]$, and transition $\tau \in \delta^{q}_{q_1 \ldots q_k}$, if $q_i \sim_M q$, then we have either

(i) $c(\tau) \equiv x_i + p$ for some polynomial $p \in P(A, X_k \setminus \{x_i\})$ and ($\mathcal{A}$ is idempotent or $p = \mathbf{0}$) or

(ii) $x_i \notin \mathrm{var}(c(\tau))$.

In Section 6 we consider instances of the aforementioned result. Generally in this section again let $M = (Q, \Sigma, \delta, F)$ be a tree automaton with cost function $c : \delta \longrightarrow P(A, X)$ over the semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$.

### *5.1. An Intermediate Characterization of Cost-Finiteness*

In this subsection—as an intermediate result—we characterize the finiteness of the set of accepting costs $[c(M)]_{\equiv}$ of a reduced tree automaton $M$ with cost function $c$ over a finitely factorizing semiring in terms of the finiteness of the sets $[c(\widehat{\Psi}^{q}_{q})]_{\equiv}$ of costs of $(q, q)$-computations for every state $q \in Q$ (cf. Corollary 35).

**Definition 30.** *The tree automaton $M$ with cost function $c$ is said to be* cost-finite *if $[c(M)]_{\equiv}$ is finite, whereas $M$ is called* cost-infinite *if it is not cost-finite.*

In the following we deeply investigate $(q, q)$-computations $\psi \in \widehat{\Psi}^q_q$, in which precisely one leaf is labeled with a variable. We would like to substitute a $q$-computation $\chi$ for this variable such that the cost of $\chi$ contributes to the cost of the computation $\psi[\chi]$, i.e., $x_1 \in \mathrm{var}(c(\psi))$. Therefore we consider reduced tree automata, which by Lemma 29 have the property that for every $(q, q)$-computation $\psi$ there exists a $(q, q)$-computation $\psi' \in \widehat{\Psi}^q_q$ such that $x_1 \in \mathrm{var}(c(\psi'))$. For the remainder of this subsection, let $M$ be reduced and $\mathcal{A}$ be finitely factorizing.

**Lemma 31.** *If $M$ is cost-finite, i.e., $[c(M)]_\equiv$ is finite, then the set $[c(M)_q]_\equiv$ is finite for every state $q \in Q$.*

*Proof.* Let $q \in Q_{\{\mathbf{0},\mathbf{1}\}}$, then the property is trivial. We prove the remaining claim by contradiction. Therefore, assume that there exists a state $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$ such that $[c(M)_q]_\equiv$ is an infinite set. Hence $c(M)_q \equiv A'$ for some infinite set $A' \subseteq A$. By the reducedness of $M$ the state $q$ is not useless, and consequently, there exists a $(q, q')$-computation $\psi \in \widehat{\Psi}^{q'}_q$ for some final state $q' \in F \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$. Finally, by Lemma 29 there also exists a $(q, q')$-computation $\psi' \in \widehat{\Psi}^{q'}_q$ such that $c(\psi')$ is zero-free and $x_1 \in \mathrm{var}(c(\psi'))$.

Let $\varphi \in \Psi^q_\varepsilon$ be a $q$-computation (which exists due to Lemma 28). We can complete $\psi'$ to a $q'$-computation $\psi'[\varphi]$ and thus according to Observation 23 obtain $c(\psi'[\varphi]) = c(\psi')[c(\varphi)]$. Recall that $q' \in F$ is a final state; hence in general we obtain

$$c(\psi'[\Psi^q_\varepsilon]) = c(\psi')[c(\Psi^q_\varepsilon)] = c(\psi')[c(M)_q] \subseteq c(M)_{q'} \subseteq c(M).$$

Since $c(M)_q \equiv A'$, we have

$$c(\psi')[c(M)_q] \equiv c(\psi')[A'] \equiv c(\psi')(A')$$

and $[c(\psi')(A')]_\equiv \subseteq [c(M)]_\equiv$ and $c(\psi')(A')$ is infinite by Lemma 9. Consequently, $[c(M)]_\equiv$ is infinite, which contradicts the assumption that $M$ is cost-finite. Consequently, $[c(M)_q]_\equiv$ is finite for every state $q \in Q$. □

In the following corollary we present another necessary condition for cost-finite tree automata with cost function over finitely factorizing semirings in terms of the set $c(\widehat{\Psi}^q_q)[a]$ of costs of $(q, q)$-computations, where $a \in c(M)_q$ is the cost of a $q$-computation. The importance of $(q, q)$-computations is illustrated in the next sentence. Apparently, every $(q, q)$-computation $\psi$ can be pumped, which gives a new $(q, q)$-computation $\psi^2$. Hence, pumping $\psi$ arbitrarily often might produce an infinite set of costs and thus might yield cost-infiniteness. Corollary 32 states this formally.

**Corollary 32.** *If $M$ is cost-finite, i.e., the set $[c(M)]_\equiv$ is finite, then the set $[c(\widehat{\Psi}^q_q)[a]\,]_\equiv$ is finite for every state $q \in Q$ and cost $a \in c(M)_q$.*

*Proof.* By $a \in c(M)_q$ there exists a $q$-computation $\chi \in \Psi^q_\varepsilon$ such that $c(\chi) = a$. We observe that every $(q, q)$-computation $\psi \in \widehat{\Psi}^q_q$ can be completed to a $q$-computation $\psi[\chi]$; hence $\widehat{\Psi}^q_q[\chi] \subseteq \Psi^q_\varepsilon$ and $c(\widehat{\Psi}^q_q[\chi]) = c(\widehat{\Psi}^q_q)[c(\chi)] = c(\widehat{\Psi}^q_q)[a]$ by Observation 23. However, $[c(\widehat{\Psi}^q_q)[a]\,]_\equiv \subseteq [c(M)_q]_\equiv$ where the set $[c(M)_q]_\equiv$ is finite by Lemma 31. Thereby also the set $[c(\widehat{\Psi}^q_q)[a]\,]_\equiv$ is finite, which proves the statement. □

Next we will prove the converse of Lemma 31, i.e., if all $(q, q)$-computations produce only finitely many costs, then $M$ is cost-finite. We perform the proof in two steps. First, in Lemma 33, we show that if all $(q, q)$-computations produce only finitely many costs, then for every two states $r$ and $q$ with $r \sim_M q$ all $(r, q)$-computations only generate a finite set of costs (for the definition of the equivalence relation $\sim_M$, see Subsection 4). Then in Lemma 34, we show that if for every two states $r$ and $q$ with $r \sim_M q$ all $(r, q)$-computations generate finitely many costs, then $M$ is cost-finite.

**Lemma 33.** *If for every state $q' \in Q$ and cost $a \in c(M)_{q'}$ the set $[c(\widehat{\Psi}^{q'}_{q'})[a]\,]_\equiv$ is finite, then also for every two states $q, r \in Q$ with $r \sim_M q$ and cost $a' \in c(M)_r$ the set $[c(\widehat{\Psi}^q_r)[a']\,]_\equiv$ is finite.*

*Proof.* First if $q \in Q_{\{\mathbf{0},\mathbf{1}\}}$ then $r \in Q_{\{\mathbf{0},\mathbf{1}\}}$ and vice versa, and the statement becomes trivial. Thus let $q, r \notin Q_{\{\mathbf{0},\mathbf{1}\}}$. Again we prove the lemma by contradiction. Therefore, let us assume the converse of the claim, i.e., $c(\widehat{\Psi}^q_r)[a'] \equiv A'$ for some infinite set $A' \subseteq A$ and cost $a' \in c(M)_r$. There exists a computation $\varphi \in \Psi^r_\varepsilon$ such that $c(\varphi) = a'$. Moreover, since $r \sim_M q$, there exists a $(q, r)$-computation, i.e., $\widehat{\Psi}^r_q \neq \emptyset$, and moreover, by Lemma 29 there exists a $(q, r)$-computation $\psi \in \widehat{\Psi}^r_q$ such that $c(\psi)$ is zero-free and $x_1 \in \mathrm{var}(c(\psi))$. Due to Observation 23, we obtain

$$c(\psi[\widehat{\Psi}^q_r[\varphi]\,]) = c(\psi[\widehat{\Psi}^q_r])[c(\varphi)] = c(\psi[\widehat{\Psi}^q_r])[a'] \subseteq c(\widehat{\Psi}^r_r)[a']$$

and

$$c(\psi[\widehat{\Psi}^q_r[\varphi]\,]) = c(\psi)[c(\widehat{\Psi}^q_r[\varphi])] = c(\psi)[c(\widehat{\Psi}^q_r)[c(\varphi)]] = c(\psi)[c(\widehat{\Psi}^q_r)[a']].$$

Besides we have $c(\psi)[c(\widehat{\Psi}^q_r)[a']\,] \equiv c(\psi)[A'] \equiv c(\psi)(A')$ where the last set is infinite due to Lemma 9. Hence $[c(\widehat{\Psi}^r_r)[a']\,]_\equiv$ is an infinite set contradicting the assumption that $[c(\widehat{\Psi}^{q'}_{q'})[a]\,]_\equiv$ is finite for every $q' \in Q$ and $a \in c(M)_{q'}$. Consequently, also $[c(\widehat{\Psi}^q_r)[a']\,]_\equiv$ is finite for every two states $r \sim_M q$ and cost $a' \in c(M)_r$. □

The following proof requires a decomposition of a computation into maximal subcomputations, which only use states of one equivalence class. However, there may appear variables in this computation, and intuitively speaking, it is at the variables where we may plug in computations using states which are not from the equivalence class. Let us therefore define for every state $q$ the class of all those computations, which only use states equivalent to $q$ at the inner nodes. Therefore, let $q, q_1, \ldots, q_k \in Q$ for some integer $k \in \mathbb{N}$. Then we define the set $\overline{\Psi}^q_{q_1 \ldots q_k}$ as follows.

$$\overline{\Psi}^q_{q_1 \ldots q_k} = \{\, \psi \in \Psi^q_{q_1 \ldots q_k} \mid (\forall w \in \mathrm{pos}(\psi))(\exists r \in [q]_{\sim_M}) : \mathrm{lab}_\psi(w) \in \delta^r \cup X_k \,\}$$

It is clear that $\overline{\Psi}^q_{q_1 \ldots q_k} = \emptyset$, if for some index $i \in [k]$ we have $q <_M q_i$. In this sense, the procedure of decomposing a computation is performed according to the partial order $\leq_M$ on the state set $Q$, which is defined in Subsection 4.

**Lemma 34.** *If for every two states $q, r \in Q$ with $r \sim_M q$ and cost $a \in c(M)_r$ the set $[c(\widehat{\Psi}^q_r)[a]\,]_\equiv$ is finite, then $M$ is cost-finite.*

*Proof.* Let us first prove an intermediate statement, namely that $[c(\overline{\Psi}^q_\varepsilon)]_\equiv$ is a finite set for every state $q \in Q$. We denote this statement by (†). Clearly, every $q$-computation $\psi \in \overline{\Psi}^q_\varepsilon$ is either a $q$-transition or there exists a decomposition $\psi = \varphi[\tau]$ into an $(r, q)$-computation $\varphi \in \widehat{\Psi}^q_r$ and an $r$-transition $\tau \in \delta^r_\varepsilon$ for some state $r \in [q]_{\sim_M}$. Hence

$$\overline{\Psi}^q_\varepsilon \subseteq \delta^q_\varepsilon \cup \Big( \bigcup_{r \in [q]_{\sim_M}} \widehat{\Psi}^q_r[\delta^r_\varepsilon] \Big)$$

and thus

$$[c(\overline{\Psi}^q_\varepsilon)]_\equiv \subseteq \Big[ c\Big( \delta^q_\varepsilon \cup \Big( \bigcup_{r \in [q]_{\sim_M}} \widehat{\Psi}^q_r[\delta^r_\varepsilon] \Big)\Big)\Big]_\equiv = [c(\delta^q_\varepsilon)]_\equiv \cup \Big( \bigcup_{r \in [q]_{\sim_M}} [c(\widehat{\Psi}^q_r)[c(\delta^r_\varepsilon)]\,]_\equiv \Big)$$

by Observation 23. Apparently, $c(\delta^q_\varepsilon)$ and $c(\delta^r_\varepsilon)$ are finite sets by the finiteness of $\delta$. Together with the observation $c(\widehat{\Psi}^q_r)[c(\delta^r_\varepsilon)] = \bigcup\{\, c(\widehat{\Psi}^q_r)[a'] \mid a' \in c(\delta^r_\varepsilon) \,\}$, we then conclude that the set $[c(\widehat{\Psi}^q_r)[c(\delta^r_\varepsilon)]\,]_\equiv$ is finite for every $r \sim_M q$ by assumption. Thus $\bigcup_{r \in [q]_{\sim_M}} [c(\widehat{\Psi}^q_r)[c(\delta^r_\varepsilon)]\,]_\equiv$ is finite, because $Q$ is finite, and thereby also $[c(\overline{\Psi}^q_\varepsilon)]_\equiv$ is finite. Hence we have proved (†).

In order to prove the statement of the lemma, we prove the stronger statement saying that for every state $q \in Q$ the set $[c(M)_q]_\equiv$ is finite. Clearly, this implies that $[c(M)]_\equiv$ is finite, i.e., $M$ is cost-finite. We apply well-founded induction on the ordered set $(Q, \leq_M)$ to prove that $[c(M)_q]_\equiv$ is finite for every state $q \in Q$.

Therefore, let $q \in Q$ be a state and $\psi \in \Psi^q_\varepsilon$ be a $q$-computation. Either $\psi \in \overline{\Psi}^q_\varepsilon$, i.e., for each leaf of $\psi$ there exists a state $r \in [q]_{\sim_M}$ such that the leaf is an $r$-transition, or there exists a decomposition $\psi = \varphi[\tau(\chi_1, \dots, \chi_k)]$ into a maximal $(r, q)$-computation $\varphi \in \widehat{\Psi}^q_r$ for some state $r \in [q]_{\sim_M}$, a transition $\tau \in \delta^r_{r_1 \dots r_k}$ for some positive integer $k \in \mathbb{N}_+$ and states $r_1, \dots, r_k \in Q \setminus [q]_{\sim_M}$, and $r_i$-computations $\chi_i \in \Psi^{r_i}_\varepsilon$ for every integer $i \in [k]$. In particular, for every $i \in [k]$ we note that $r_i <_M q$. Hence

$$\Psi^q_\varepsilon \subseteq \overline{\Psi}^q_\varepsilon \cup \Big( \bigcup_{r \in [q]_{\sim_M}} \widehat{\Psi}^q_r \Big[ \bigcup_{\substack{k \in [\max \mathrm{rk}_\delta(\delta)],\\ (\forall i \in [k]) \colon r_i \in Q,\, r_i <_M q}} \delta^r_{r_1 \dots r_k}[\Psi^{r_1}_\varepsilon, \dots, \Psi^{r_k}_\varepsilon] \Big] \Big)$$

and thus

$$\begin{aligned}
[c(\Psi^q_\varepsilon)]_\equiv &\subseteq \left[ c \left( \overline{\Psi}^q_\varepsilon \cup \Big( \bigcup_{r \in [q]_{\sim_M}} \widehat{\Psi}^q_r \Big[ \bigcup_{\substack{k \in [\max \mathrm{rk}_\delta(\delta)],\\ (\forall i \in [k]) \colon r_i \in Q,\, r_i <_M q}} \delta^r_{r_1 \dots r_k}[\Psi^{r_1}_\varepsilon, \dots, \Psi^{r_k}_\varepsilon] \Big] \Big) \right) \right]_\equiv \\
&= [c(\overline{\Psi}^q_\varepsilon)]_\equiv \cup \left( \bigcup_{r \in [q]_{\sim_M}} \left[ c(\widehat{\Psi}^q_r) \left[ \bigcup_{\substack{k \in [\max \mathrm{rk}_\Sigma(\Sigma)],\\ (\forall i \in [k]) \colon r_i \in Q,\, r_i <_M q}} c(\delta^r_{r_1 \dots r_k}) \Big[ c(\Psi^{r_1}_\varepsilon), \dots, c(\Psi^{r_k}_\varepsilon) \Big] \right] \right]_\equiv \right)
\end{aligned}$$

by Observation 23. So to show that $[c(\Psi^q_\varepsilon)]_\equiv$ is finite, it is sufficient to show that (i) $[c(\overline{\Psi}^q_\varepsilon)]_\equiv$ is finite, (ii) $[c(\widehat{\Psi}^q_r)[a']\,]_\equiv$ is finite for every state $r \in [q]_{\sim_M}$ and

$a' \in c(M)_r$, (iii) $[c(\delta^r_{r_1\ldots r_k})[a_1,\ldots,a_k]\,]_\equiv$ is finite for every integer $k \in \mathbb{N}$, $k+1$ states $r, r_1, \ldots, r_k \in Q$ and $a_i \in c(M)_{r_i}$ for every $i \in [k]$, and (iv) $[c(\Psi^r_\varepsilon)]_\equiv$ is finite for every state $r \in Q$ with $r <_M q$. Statement (i) is proved by Statement (†), Statement (ii) is given by the assumption, Statement (iii) is trivial because $\delta$ is finite, and Statement (iv) is given by the induction hypothesis. Consequently, the statement is proved by Principle 1. □

Let us now sum up the results, which we have proved in this subsection so far. This gives a characterization of cost-finiteness of reduced tree automata with cost function over a finitely factorizing semiring.

**Corollary 35.** *The following statements are equivalent.*

*(i) $M$ is cost-finite, i.e., $[c(M)]_\equiv$ is finite.*

*(ii) For every state $q \in Q$ and cost $a \in c(M)_q$ the set $[c(\widehat{\Psi}^q_q)[a]\,]_\equiv$ is finite.*

*Proof.* The direction (i) ⇒ (ii) was proved in Corollary 32, while the proof of (ii) ⇒ (i) follows from Lemma 33 and Lemma 34. □

### *5.2. Condition (linear) is Necessary for Cost-Finiteness*

Let us now use the intimate knowledge of monotonic semirings and thereby characterize cost-finiteness in a more sophisticated way. We will show that, roughly speaking, a given reduced tree automaton with costs is cost-finite, if and only if the cost function of every $(q,q)$-computation is semantically equivalent to either a constant or a linear polynomial of type $x_1 + a$, where $a = \mathbf{0}$ if the underlying semiring is not idempotent. Let us define the appropriate condition formally. Recall that $M = (Q, \Sigma, \delta, F)$ is a tree automaton with cost function $c : \delta \longrightarrow P(A, X)$ over the semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$.

**Definition 36.** *The Condition (linear) holds, if for every state $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$ and $(q,q)$-computation $\psi \in \widehat{\Psi}^q_q$ there exists a semiring element $a \in A$ such that either*

*(i) $c(\psi) \equiv x_1 + a$ and ($\mathcal{A}$ is idempotent or $a = \mathbf{0}$) or*

*(ii) $c(\psi) \equiv a$.*

We first show that Condition (ii) of Corollary 35, i.e., for every state $q \in Q$ and cost $a \in c(M)_q$ the set $[c(\widehat{\Psi}^q_q)[a]\,]_\equiv$ is finite, implies that Condition (linear) holds.

**Lemma 37.** *Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1}, \preceq)$ be a monotonic and finitely factorizing semiring. If $[c(\widehat{\Psi}^q_q)[a]\,]_\equiv$ is a finite set for every state $q \in Q$ and cost $a \in c(M)_q$, then Condition (linear) holds.*

*Proof.* We prove the claim by contradiction. Thus let $[c(\widehat{\Psi}^q_q)[a]\,]_\equiv$ be a finite set for every state $q \in Q$ and cost $a \in c(M)_q$, but there exist a state $r \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$ and a $(r,r)$-computation $\psi \in \widehat{\Psi}^r_r$ such that for every semiring element $a \in A$ we have $c(\psi) \not\equiv a$ and

(i) $c(\psi) \not\equiv x_1 + a$, or

(ii) $\mathcal{A}$ is not idempotent and $c(\psi) \not\equiv x_1$.

Since $M$ has no useless states, there exists an $r$-computation $\chi \in \Psi^r_\varepsilon$, and since $r$ is not a $\{\mathbf{0},\mathbf{1}\}$-state, we may select $\chi$ such that $c(\chi) \equiv a'$ for some $a' \in A \setminus \{\mathbf{0},\mathbf{1}\}$. Now we split the proof into two cases, each of which leads to a contradiction.

Case 1: First assume that $2 \leq \deg_s(c(\psi))$. Then by Item (i) in Lemma 14 and Lemma 15 we immediately have

$$\mathbf{1} \prec a' \prec c(\psi)(a') \prec c(\psi)(c(\psi)(a')) = c(\psi^2)(a') \prec c(\psi^3)(a') \prec \dots .$$

Hence $[\{\, c(\psi^n)(a') \mid n \in \mathbb{N} \,\}]_\equiv$ is an infinite set. Further

$$[\{\, c(\psi^n)[a'] \mid n \in \mathbb{N} \,\}]_\equiv \subseteq [c(\widehat{\Psi}^r_r)[a']\,]_\equiv .$$

Thus also $[c(\widehat{\Psi}^r_r)[a']\,]_\equiv$ is an infinite set, which is a contradiction to the assumption.

Case 2: Now assume that $\deg_s(c(\psi)) \leq 1$. We immediately obtain $\deg_s(c(\psi)) = 1$, else $c(\psi) \equiv a$ which is a contradiction. Further, according to (i) or (ii) the following two subcases are possible.

Subcase 2.1: Let $c(\psi) \notin \bigcup_{a\in A}[x_1 + a]_\equiv$. Then by Lemma 16 we again have

$$\mathbf{1} \prec a' \prec c(\psi)(a') \prec c(\psi)(c(\psi)(a')) = c(\psi^2)(a') \prec c(\psi^3)(a') \prec \dots .$$

Thus it can be proved in the same way as in Case 1 that $[c(\widehat{\Psi}^r_r)[a']\,]_\equiv$ is infinite, which is a contradiction.

Subcase 2.2: Let $c(\psi) \not\equiv x_1$ and $\mathcal{A}$ is not idempotent, i.e., $\mathbf{1} \prec \mathbf{1} \oplus \mathbf{1}$. We may safely assume that $c(\psi) \equiv x_1 + a$ for some $a \in A_+$, because otherwise the previous subcase already derives a contradiction. We observe that for every $n \in \mathbb{N}$ we have $c(\psi^n)(a') = a' \oplus \sum_{j\in[n]} a$. Moreover, $\{\, c(\psi^n)(a') \mid n \in \mathbb{N} \,\} \subseteq c(\widehat{\Psi}^r_r)(a')$, thus we need to show that

$$\{\, a' \oplus \sum_{j\in[n]} a \mid n \in \mathbb{N} \,\} = a' \oplus \{\, \sum_{j\in[n]} a \mid n \in \mathbb{N} \,\}$$

is infinite. According to Observation 2, it is sufficient to show that $\{\, \sum_{j\in[n]} a \mid n \in \mathbb{N} \,\}$ is infinite. Since $\mathcal{A}$ is not idempotent, but monotonic, we conclude that for every $a \in A_+$ we have $a \prec a \oplus a$ by Observation 17. Hence the set $\{\, \sum_{j\in[n]} a \mid n \in \mathbb{N} \,\}$ is infinite. $\square$

### *5.3. Two Further Sufficient Conditions for Cost-Finiteness*

Firstly we show that Condition (linear) is also sufficient for cost-finiteness provided that the underlying semiring is not idempotent.

**Lemma 38.** *Let $\mathcal{A}$ be non-idempotent. If Condition (linear) holds, then for every $r \in Q$ and $r$-computation $\varphi \in \Psi^r_\varepsilon$ there exists a $r$-computation $\varphi' \in \Psi^r_\varepsilon$ such that $c(\varphi) \equiv c(\varphi')$ and* $\mathrm{height}(\varphi') \leq 2 \cdot \mathrm{card}(Q)$.

*Proof.* Since Condition (linear) holds and $\mathcal{A}$ is not idempotent, for every state $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$ and $(q,q)$-computation $\psi \in \widehat{\Psi}^q_q$ either $c(\psi) \equiv x_1$ or $c(\psi) \equiv a$ for some $a \in A$. We prove the lemma by contradiction. Therefore let us assume that there is a state $r \in Q$, an $r$-computation $\varphi \in \Psi^r_\varepsilon$ such that for every $r$-computation $\varphi' \in \Psi^r_\varepsilon$ with $c(\varphi) \equiv c(\varphi')$, the condition $2 \cdot \mathrm{card}(Q) < \mathrm{height}(\varphi')$ holds.

Let $\varphi' \in \Psi^r_\varepsilon$ be such that $c(\varphi) \equiv c(\varphi')$, $2 \cdot \mathrm{card}(Q) < \mathrm{height}(\varphi')$ and the cardinality of the set $W_{\varphi'} = \{\, w \in \mathrm{pos}(\varphi') \mid 2 \cdot \mathrm{card}(Q) \leq |w| \,\}$ is minimal. Clearly, $W_{\varphi'}$ is finite and cannot be empty. Let us take a $w \in W_{\varphi'}$. There exist positions $w_1, w_2 \in \mathrm{pos}(\varphi')$ with $w_1 < w_2 < w$ (with respect to the prefix order) and $|w_2| \leq \mathrm{card}(Q)$ such that for some state $q \in Q$ we have $\mathrm{lab}_{\varphi'}(w_1) \in \delta^q$ and $\mathrm{lab}_{\varphi'}(w_2) \in \delta^q$. Furthermore, there exists a $(q, r)$-computation $\zeta_1 \in \widehat{\Psi}^r_q$, a $(q, q)$-computation $\zeta_2 \in \widehat{\Psi}^q_q$, and a $q$-computation $\zeta_3 \in \Psi^q_\varepsilon$ such that $\varphi' = \zeta_1[\zeta_2[\zeta_3]\,]$ and $\varphi'|_{w_1} = \zeta_2[\zeta_3]$ and $\varphi'|_{w_2} = \zeta_3$. Note that $\mathrm{height}(\zeta_3) > \mathrm{card}(Q)$.

Case 1: Let $q \in Q_{\{\mathbf{0},\mathbf{1}\}}$. Then either $c(\zeta_2[\zeta_3]) \equiv c(\zeta_3)$, in which case we set $\zeta = \zeta_1[\zeta_3]$ and observe $c(\varphi') \equiv c(\zeta)$ while $W_\zeta \subset W_{\varphi'}$. Thus this case is contradictory. Now assume $c(\zeta_2[\zeta_3]) \not\equiv c(\zeta_3)$. Now there exists a computation $\zeta \in \Psi^q_\varepsilon$ with $\mathrm{height}(\zeta) \leq \mathrm{card}(Q)$ and either $c(\zeta) \equiv \mathbf{0}$ or $c(\zeta) \equiv \mathbf{1}$. Consequently, $c(\zeta) \equiv c(\zeta_2[\zeta_3])$ or $c(\zeta) \equiv c(\zeta_3)$. Thus, both $W_{\zeta_1[\zeta]} \subset W_{\varphi'}$ and $W_{\zeta_1[\zeta_2[\zeta]\,]} \subset W_{\varphi'}$ and either $c(\varphi') \equiv c(\zeta_1[\zeta])$ or $c(\varphi') \equiv c(\zeta_1[\zeta_2[\zeta]\,])$. Again we derived a contradiction.

Case 2: Let $q \notin Q_{\{\mathbf{0},\mathbf{1}\}}$ and $c(\zeta_2) \equiv x_1$. Then

$$c(\zeta_1[\zeta_2[\zeta_3]\,]) = c(\zeta_1)[c(\zeta_2)[c(\zeta_3)]\,] \equiv c(\zeta_1)[x_1[c(\zeta_3)]\,] = c(\zeta_1)[c(\zeta_3)] = c(\zeta_1[\zeta_3]).$$

Thus $\zeta = \zeta_1[\zeta_3]$, which is an $r$-computation, has cost $c(\zeta) \equiv c(\varphi')$ and $W_\zeta \subset W_{\varphi'}$, which constitutes a contradiction.

Case 3: Let $c(\zeta_2) \equiv a$ for some semiring element $a \in A$. Certainly, there exists a $q$-computation $\zeta \in \Psi^q_\varepsilon$ such that $\mathrm{height}(\zeta) \leq \mathrm{card}(Q)$. Then $c(\zeta_1[\zeta_2[\zeta]\,]) \equiv c(\varphi')$ and $W_{\zeta_1[\zeta_2[\zeta]\,]} \subset W_{\varphi'}$. Hence all cases are contradictory, which yields that $W_{\varphi'} = \emptyset$ successfully proving the statement. □

As an immediate corollary of the above lemma, we obtain our first sufficient condition for cost-finiteness.

**Corollary 39.** *Let $\mathcal{A}$ be non-idempotent. If Condition (linear) holds, then $[c(M)]_\equiv$ is finite.*

Next we give a sufficient condition of cost-finiteness in the case that the underlying semiring is finitely factorizing, monotonic, and idempotent. For this, we define another condition called (linear-trans).

**Definition 40.** *We say that Condition (linear-trans) holds, if for every $k \in \mathbb{N}_+$, state $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$, states $q_1, \ldots, q_k \in Q$, index $i \in [k]$, and transition $\tau \in \delta^q_{q_1 \cdots q_k}$, if $q_i \sim_M q$, then we have either*

*(i)* $c(\tau) \equiv x_i + p$ *for some polynomial* $p \in P(A, X_k \setminus \{x_i\})$ *and* ($\mathcal{A}$ *is idempotent or* $p = \mathbf{0}$) *or*

*(ii)* $x_i \notin \mathrm{var}(c(\tau))$.

Note that, on the contrary to Condition (linear), (linear-trans) refers to finitely many costs. This property makes Condition (linear-trans) suitable for deciding cost-finiteness under certain additional conditions. First however, let us prove that (linear-trans) is sufficient to guarantee cost-finiteness provided that the underlying semiring is finitely factorizing, monotonic, and idempotent.

**Lemma 41.** *Let $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1}, \preceq)$ be a finitely factorizing, monotonic, and idempotent semiring. If $M$ obeys Condition (linear-trans), then $M$ is cost-finite.*

*Proof.* First we let $l \in \mathbb{N}$ be the integer $l = \max\{\, \mathrm{size}(c(\tau)) \mid \tau \in \delta \,\}$ and we let $C' \subseteq A$ be the set

$$C' = \{\mathbf{0}, \mathbf{1}\} \cup \{\, a \in A \mid \tau \in \delta, 1 \leq |c(\tau)|_a \,\}.$$

Obviously, the set $C'$ is finite. Finally we define for every state $q \in Q$ the set $C_q \subseteq A$ by well-founded induction on $(Q, \leq_M)$ (cf. Principle 1) as follows.

$$C_q = \Big\langle \{\, a_1 \odot \ldots \odot a_n \mid n \in [l], a_1, \ldots, a_n \in C' \cup \Big( \bigcup_{r \in Q, r <_M q} C_r \Big) \,\} \Big\rangle_\oplus$$

Note that $\langle A' \rangle_\oplus$ is the closure of the set $A' \subseteq A$ under $\oplus$, i.e., the smallest submonoid of $(A, \oplus, \mathbf{0})$ containing $A'$ in its carrier.

Next we prove that $C_q$ is finite for every state $q \in Q$. We prove the statement by well-founded induction on $(Q, \leq_M)$, hence assume that $C_r$ is finite for every state $r \in Q$ with $r <_M q$. Then clearly $S = C' \cup \bigcup_{r \in Q, r <_M q} C_r$ is finite and since there are only finitely many words over $S$ of length at most $l$, the set

$$S' = \{\, a_1 \odot \ldots \odot a_n \mid n \in [l], a_1, \ldots, a_n \in S \,\}$$

is also finite. However, the closure of a finite set $S'$ under $\oplus$, which is idempotent, is again a finite set, because there are only finitely many subsets of $S'$. Consequently, $C_q$ is finite.

Now we will prove that $[c(M)_q]_\equiv \subseteq [C_q]_\equiv$ for every state $q \in Q$, which immediately yields that $[c(M)]_\equiv = \bigcup_{q \in F} [c(M)_q]_\equiv$ is finite and hence $M$ is cost-finite. We again prove this property by well-founded induction along $(Q, \leq_M)$, namely we prove that for every state $q \in Q$ and $q$-computation $\psi \in \Psi^q_\varepsilon$ we have $[c(\psi)]_\equiv \in [C_q]_\equiv$. Therefore we decompose the computation $\psi = \psi'[\psi_1, \ldots, \psi_n]$ into a $(q_1 \ldots q_n, q)$-computation $\psi' \in \overline{\Psi}^q_{q_1 \ldots q_n}$ and $q_i$-computations $\psi_i \in \Psi^{q_i}_\varepsilon$ for some integer $n \in \mathbb{N}$, $n$ states $q_1, \ldots, q_n \in Q$ with $q_i <_M q$ for every $i \in [n]$. Thus by induction hypothesis we have $[c(\psi_i)]_\equiv \in [C_{q_i}]_\equiv$. Note that $c(\psi) = c(\psi')[c(\psi_1), \ldots, c(\psi_n)]$ by Observation 23.

Since $\psi'$ has a tree structure, we perform structural induction on $\psi'$ in order to prove the statement.

**Induction base:** Let $\psi' = \tau \in \delta^{(0)}$. Then $c(\psi')[c(\psi_1), \ldots, c(\psi_n)] = c(\tau)$, because $c(\psi') = c(\tau)$. Further, we immediately note that $c(\tau) \in P(C', \emptyset)$ and thereby $[c(\tau)]_\equiv \in [C_q]_\equiv$. On the other hand, let $\psi' = x_i$ for some index $i \in [n]$. Then $c(\psi')[c(\psi_1), \ldots, c(\psi_n)] = c(\psi_i)$, and consequently, $[c(\psi')]_\equiv \in [C_q]_\equiv$, because $[c(\psi_i)]_\equiv \in [C_{q_i}]_\equiv$ and $q_i <_M q$.

**Induction step:** Let $\psi' = \tau(\zeta_1, \ldots, \zeta_k)$ for some $k \in \mathbb{N}_+$, states $r_1, \ldots, r_k \in Q$, a transition $\tau \in \delta^q_{r_1 \ldots r_k}$, and $(q_1 \ldots q_n, r_i)$-computations $\zeta_i \in \overline{\Psi}^{r_i}_{q_1 \ldots q_n}$ for every index $i \in [k]$. Furthermore, let $\zeta'_i = \zeta_i[\psi_1, \ldots, \psi_n]$, thus $\psi = \tau(\zeta'_1, \ldots, \zeta'_k)$. By the induction hypothesis (of the inner structural induction) we have $[c(\zeta'_i)]_\equiv \in [C_{q_i}]_\equiv$. Furthermore we observe that

$$\begin{aligned}
c(\psi')[c(\psi_1), \ldots, c(\psi_n)] &= c(\tau(\zeta_1, \ldots, \zeta_k))[c(\psi_1), \ldots, c(\psi_n)] \\
&= c(\tau)[c(\zeta_1), \ldots, c(\zeta_k)]\,[c(\psi_1), \ldots, c(\psi_n)] \\
&= c(\tau)[c(\zeta_1)[c(\psi_1), \ldots, c(\psi_n)], \ldots, c(\zeta_k)[c(\psi_1), \ldots, c(\psi_n)]\,] \\
&= c(\tau)[c(\zeta'_1), \ldots, c(\zeta'_k)].
\end{aligned}$$

By Condition (linear-trans) we have

$$c(\tau) = \sum_{i \in I} x_i + p$$

for some $p \in P(A, X_k \setminus \{\, x_i \mid i \in I \,\})$ and subset $I \subseteq \{\, i \in [k] \mid q \sim_M r_i \,\}$. Consequently, $[p[c(\zeta'_1), \ldots, c(\zeta'_k)]\,]_\equiv \in [C_q]_\equiv$ by the definition of $C_q$ and also

$$(\sum_{i \in I} x_i + p)[c(\zeta'_1), \ldots, c(\zeta'_k)] = c(\psi')[c(\psi_1), \ldots, c(\psi_n)] = c(\psi) \in [C_q]_\equiv,$$

because $c(\zeta'_i) \in [C_{q_i}]_\equiv$ and $[C_{q_i}]_\equiv \subseteq [C_q]_\equiv$, and $C_q$ is closed under addition. □

### *5.4. A Characterization of Cost-Finiteness in Terms of Linear Costs*

In this subsection we give another, more impressive characterization of cost-finiteness in terms of the Conditions (linear) and (linear-trans). As a first step, we prove that these two conditions are equivalent for finitely factorizing and monotonic semirings. We start with a preparatory lemma.

**Lemma 42.** *Let $M$ be reduced and $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1}, \preceq)$ be a finitely factorizing semiring. If Condition (linear) holds, then for every two states $q, r \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$ with $q \sim_M r$ and every $(r, q)$-computation $\psi \in \widehat{\Psi}^q_r$ we have for some semiring element $a \in A$ either*

- *$c(\psi) \equiv x_1 + a$ and ($\mathcal{A}$ is idempotent or $a = \mathbf{0}$), or*
- *$c(\psi) \equiv a$.*

*Proof.* Since $q \sim_M r$, there exists a $(q, r)$-computation $\varphi \in \widehat{\Psi}^r_q$ and since $M$ is reduced, we may assume without loss of generality that $x_1 \in \mathrm{var}(c(\varphi))$ (cf. Lemma 29). Consequently, $\psi[\varphi] \in \widehat{\Psi}^q_q$ constitutes a $(q, q)$-computation. According to Condition (linear), we distinguish two cases for $c(\psi[\varphi]) = c(\psi)[c(\varphi)]$.

Case 1: Let $c(\psi)[c(\varphi)] \equiv x_1 + a'$ for some semiring element $a' \in A$ and ($\mathcal{A}$ is idempotent or $a' = \mathbf{0}$). Then by Lemma 20 we immediately have $c(\psi) \equiv x_1 + a$ for some semiring element $a \in A$ proving the statement. Moreover, $a = \mathbf{0}$ whenever $\mathcal{A}$ is not idempotent.

Case 2: Let $c(\psi)[c(\varphi)] \equiv a'$ for some semiring element $a' \in A$. Since $x_1 \in \mathrm{var}(c(\varphi))$, we must have $c(\psi) \equiv a'$. □

Now we can prove the equivalence of the Conditions (linear) and (linear-trans) for finitely factorizing and monotonic semirings.

**Lemma 43.** *Let $M$ be reduced and $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1}, \preceq)$ be a finitely factorizing and monotonic semiring. Then Condition (linear) and Condition (linear-trans) are equivalent.*

*Proof.* First we prove that Condition (linear-trans) implies Condition (linear). Therefore let $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$ be a state and $\psi \in \widehat{\Psi}^q_q$ be a $(q,q)$-computation. Furthermore, let $w \in \mathrm{pos}(\psi)$ be the unique position such that $\mathrm{lab}_\psi(w) = x_1$ and for every $j \in [0, |w|-1]$ let $w_j$ be the strict prefix of $w$ which has length $j$ and $n_j \in \mathbb{N}_+$ be the integer such that $w_{j+1} = w_j \cdot n_j$. We distinguish two cases according to Condition (linear-trans). Either for every $j \in [0, |w|-1]$ we have $c(\mathrm{lab}_\psi(w_j)) \equiv x_{n_j} + p_j$ for some polynomial $p_j \in P(A, X \setminus \{x_{n_j}\})$, then $c(\psi) \equiv x_1 + a$ for some semiring element $a \in A$ thus fulfilling Condition (linear). Note that $a = \mathbf{0}$, if $\mathcal{A}$ is not idempotent.

On the other hand, there may exist an integer $j \in [0, |w|-1]$ such that $x_{n_j} \notin \mathrm{var}(c(\mathrm{lab}_\psi(w_j)))$. In this case, we have $x_1 \notin \mathrm{var}(c(\psi))$, hence $c(\psi) \equiv a$ for some semiring element $a \in A$, also fulfilling Condition (linear). According to Condition (linear-trans) this case-distinction is complete, so we have proved one direction.

For the proof of the other direction let $\tau \in \delta^q_{q_1 \dots q_k}$ be a transition for some integer $k \in \mathbb{N}_+$, states $q_1, \dots, q_k \in Q$ and $q \in Q \setminus Q_{\{\mathbf{0},\mathbf{1}\}}$, and let $i \in [k]$ be such that $q_i \sim_M q$. Since $M$ has no useless states there exists a $q_j$-computation $\psi_j \in \Psi^{q_j}_\varepsilon$ for every index $j \in [k]$. Further, by reducedness we may assume without loss of generality that $\mathbf{1} \prec c(\psi_j)$ for every index $j \in [k]$ with $x_j \in \mathrm{var}(c(\tau))$. Consequently, $\psi = \tau(\psi_1, \dots, \psi_{i-1}, x_1, \psi_{i+1}, \dots, \psi_k)$ is a $(q_i, q)$-computation $\psi \in \widehat{\Psi}^q_{q_i}$ and

$$c(\psi) = c(\tau)[c(\psi_1), \dots, c(\psi_{i-1}), x_1, c(\psi_{i+1}), \dots, c(\psi_k)].$$

Due to $q_i \sim_M q$, we have either $c(\psi) \equiv x_1 + a$ or $c(\psi) \equiv a$ for some semiring element $a \in A$ by Lemma 42. In the former case $a = \mathbf{0}$, if $\mathcal{A}$ is not idempotent.

In case $c(\psi) \equiv a$, by Corollary 10, we have $x_1 \notin \mathrm{var}(c(\psi))$ and by several applications of Observation 8 also $x_i \notin \mathrm{var}(c(\tau))$, hence Item (ii) of Condition (linear-trans) is fulfilled.

In case $c(\psi) \equiv x_1 + a$, by Corollary 10, we have $x_1 \in \mathrm{var}(c(\psi))$ and again by repeated applications of Observation 8 we can conclude that $x_i \in \mathrm{var}(c(\tau))$. Now either $\mathcal{A}$ is not idempotent and then by $c(\psi) \equiv x_1$ also $c(\tau) \equiv x_i$. Thus Item (i) of Condition (linear-trans) is fulfilled. Otherwise $\mathcal{A}$ is idempotent and by Lemma 21 we obtain $c(\tau) \equiv x_i + p$ for some polynomial $p \in P(A, X_k \setminus \{x_i\})$ also fulfilling Item (i) in Condition (linear-trans). □

Now we are able to give our main characterization theorem for reduced tree automata with costs over finitely factorizing and monotonic semirings in terms of linear costs.

**Theorem 44.** *Let $M = (Q, \Sigma, \delta, F)$ be a reduced tree automaton with cost function $c : \delta \longrightarrow P(A, X)$ over a monotonic and finitely factorizing semiring. The following statements are equivalent.*

*(i)* *$M$ is cost-finite.*

*(ii)* *For every state $q \in Q$ and cost $a \in c(M)_q$ the set $[c(\widehat{\Psi}^q_q)(a)]_\equiv$ is finite.*

*(iii)* *Condition (linear) holds.*

*(iv)* *Condition (linear-trans) holds.*

*Proof.* The equivalence of Items (i) and (ii) is proved in Corollary 35. Moreover, by Lemma 43, Item (iii) is equivalent to Item (iv). Lemma 37 yields that Item (ii) implies Item (iii). If $\mathcal{A}$ is idempotent, then Item (iv) implies Item (i) by Lemma 41, otherwise Item (iii) implies Item (i) by Corollary 39. □

*5.5. Decidability of Cost-Finiteness*

In this subsection we establish the decidability of cost-finiteness of tree automata with costs over finitely factorizing and monotonic semirings. Since cost-finiteness and Condition (linear-trans) are equivalent for such semirings by Theorem 44, it is sufficient to show that this latter property is decidable.

**Lemma 45.** *Let $M$ be reduced and $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1}, \preceq)$ be a finitely factorizing and monotonic semiring. Then it is decidable whether Condition (linear-trans) is satisfied.*

*Proof.* We have to check, for finitely many transitions $\tau \in \delta$ and finitely many variables $x \in X_{\max \mathrm{rk}_\Sigma(\Sigma)}$, whether we have either $x \notin \mathrm{var}(c(\tau))$ or $c(\tau) \equiv x + p$ with $p \in P(A, X \setminus \{x\})$, where in case $\mathcal{A}$ is not idempotent we additionally have $p = \mathbf{0}$. The condition $x \notin \mathrm{var}(c(\tau))$ is certainly decidable.

If $\mathcal{A}$ is not idempotent, then by Lemma 18 it is decidable if $c(\tau) \equiv x$. Finally, in case $\mathcal{A}$ is idempotent it is decidable if $c(\tau) \equiv x + p$ as follows. There are integers $n \in \mathbb{N}$ and $i \in [n]$ such that $c(\tau) \in P(A, X_n)$ and $x = x_i$. Now, we have to decide if $p \equiv x_i + p'$, where $p' \in P(A, X_n \setminus \{x_i\})$. By Lemma 21, it is enough to decide whether for some $a_1, \ldots, a_n \in A \setminus \{\mathbf{0}, \mathbf{1}\}$, we have $p_1[a_1, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_n] \equiv x_i + a$ where $a \in A$. However, this is decidable by Lemma 19. □

Now we are able to state and prove our main decidability result. Recall that, if the underlying semiring is positive, one-summand free, and one-product free, then for every tree automaton with cost function, a cost-equivalent reduced tree automaton can effectively be constructed. Hence we could also require the underlying semiring to have these properties, but indeed monotonic semirings have all the aforementioned properties (cf. Lemma 14). Thus in the following main decidability theorem we can (without loss of generality) drop the assumption of $M$ being reduced.

**Theorem 46.** *For every tree automaton $M = (Q, \Sigma, \delta, F)$ with cost function $c : \delta \longrightarrow P(A, X)$ over a monotonic and finitely factorizing semiring it is decidable whether $M$ is cost-finite.*

*Proof.* By Lemma 27, a reduced tree automaton $M' = (Q', \Sigma, \delta', F')$ with cost function $c' : \delta' \longrightarrow P(A, X)$ can effectively be constructed such that $c(M) \equiv c'(M')$. Hence $M$ is cost-finite if and only if $M'$ is cost-finite. On the other hand, by Theorem 44, $M'$ is cost-finite if and only if Condition (linear-trans) holds for $M'$. Finally, by Lemma 45, it is decidable whether $M'$ satisfies Condition (linear-trans). □

## 6. From Cost-Finiteness to Boundedness

In this section we relate cost-finiteness and boundedness (with respect to the natural order) of tree automata with costs over naturally ordered, finitely factorizing, and monotonic semirings. Intuitively speaking, a tree automaton with cost function is bounded, if there exists an upper bound for the set of accepting costs. Henceforth, let $M = (Q, \Sigma, \delta, F)$ be a tree automaton with cost function $c : \delta \longrightarrow P(A, X)$ over a semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$. The exact definition for boundedness is then as follows.

**Definition 47.** *Let $\preceq \subseteq A^2$ is a partial order on $A$. The automaton $M$ is said to be* bounded *(with respect to $\preceq$), if there exists an element $b \in A$ such that for every $a \in A$ with $a \equiv p$ for some $p \in c(M)$ we have $a \preceq b$.*

We will consider tree automata with costs over naturally ordered semirings and will consider if they are bounded with respect to the natural order (which is certainly a partial order). For this, the following observation is very useful.

**Lemma 48.** *Let $\mathcal{A}$ be naturally ordered (via $\sqsubseteq$) and $(A, \oplus, \mathbf{0})$ be finitely factorizing. For every $C \subseteq A$, the following two statements are equivalent.*

*(i) $C$ is finite.*

*(ii) $C$ is bounded with respect to $\sqsubseteq$.*

*Proof.* Firstly, we show (i) $\Rightarrow$ (ii). Therefore, let $C$ be finite, then $a = \bigoplus_{c \in C} c$ is defined and for every $c \in C$ we immediately observe $c \sqsubseteq a$, because there exists an $a' = \bigoplus_{c' \in C \setminus \{c\}} c'$ such that $c \oplus a' = a$. To prove the converse, we assume that $C$ is bounded with respect to $\sqsubseteq$. Hence there exists an element $a \in A$ such that $c \sqsubseteq a$ for every $c \in C$. We consider the set $A' = \{\, a' \in A \mid a' \sqsubseteq a \,\}$. Apparently, $C \subseteq A'$. Assume that $A'$ is infinite, then $D^{\oplus}(a)$ is also infinite, because for every $a' \in A'$ there exists a semiring element $a'' \in A$ such that $a = a' \oplus a''$ by $a' \sqsubseteq a$. This would yield that the monoid $(A, \oplus, \mathbf{0})$ is not finitely factorizing. This contradicts the assumption, hence $A'$ and thereby also $C$ is finite. □

Now the following lemma immediately follows from the definitions and the above observation.

**Lemma 49.** *Let $\mathcal{A}$ be naturally ordered (via $\sqsubseteq$) and $(A, \oplus, \mathbf{0})$ be finitely factorizing. Then $M$ is cost-finite if and only if $M$ bounded with respect to $\sqsubseteq$.*

*Proof.* We prove the claim by a chain of equivalences.

$$\begin{aligned}
& M \text{ is cost-finite} \\
\iff & [c(M)]_{\equiv} \text{ is finite (Definition 30)} \\
\iff & \{\, a \in A \mid (\exists q \in F)(\exists p \in c(\Psi^q_\varepsilon)) : \; p \equiv a \,\} \text{ is finite} \\
\iff & \{\, a \in A \mid (\exists q \in F)(\exists p \in c(\Psi^q_\varepsilon)) : \; p \equiv a \,\} \text{ is bounded with respect to } \sqsubseteq \\
& \text{(see Lemma 48)} \\
\iff & M \text{ is bounded with respect to } \sqsubseteq \text{ (Definition 47)}
\end{aligned}$$

□

Now we can combine Theorem 46 and Lemma 49 for finitely factorizing, monotonic, and naturally ordered semirings. Note that the partial order $\preceq$ and natural order $\sqsubseteq$, such that $\mathcal{A}$ is monotonic with respect to $\preceq$, may well be the different.

**Theorem 50.** *Let $M$ be a tree automaton with cost function over a finitely factorizing and monotonic semiring $\mathcal{A}$, which is naturally ordered (via $\sqsubseteq$). Then it is decidable whether $M$ is bounded with respect to $\sqsubseteq$.*

*Proof.* By Lemma 49, $M$ is bounded with respect to $\sqsubseteq$ if and only if it is cost-finite. Moreover, by Theorem 46, it is decidable whether $M$ is cost-finite. □

To conclude this section, we apply the above theorem to some concrete instances of semirings. Thereby, we reobtain two decidability results of [31] (see Theorems 3.2 and 3.4 of [31]) in the first two corollaries and demonstrate that our results can be applied to other important semirings.

**Corollary 51.** *It is decidable whether a tree automaton $M$ with cost function over the semiring* $\mathrm{Nat} = (\mathbb{N}, +, \cdot, 0, 1)$ *is bounded with respect to $\leq$.*

*Proof.* Apparently, Nat is finitely factorizing and naturally ordered (via $\leq$). Moreover, Nat is also monotonic with respect to $\leq$. Thus Theorem 50 yields the stated. □

**Corollary 52.** *Let $M$ be a tree automaton with cost function over the arctic semiring* $\mathrm{Arct} = (\mathbb{N} \cup \{-\infty\}, \max, +, (-\infty), 0)$. *It is decidable whether $M$ is bounded with respect to $\leq$.*

*Proof.* As in the previous corollary, Arct is finitely factorizing and naturally ordered as well as monotonic with respect to $\leq$. Thus Theorem 50 again proves the statement. □

**Corollary 53.** *It is decidable whether a tree automaton $M$ with cost function over the naturally ordered* lcm*-semiring* $\mathrm{Lcm} = (\mathbb{N}, \mathrm{lcm}, \cdot, 0, 1)$ *is bounded with respect to the natural order $\sqsubseteq$.*

*Proof.* Clearly, Lcm is naturally ordered and finitely factorizing. In fact, it is even monotonic with respect to $\sqsubseteq$. This allows us to apply Theorem 50, hence we can decide whether $M$ is bounded with respect to the natural order $\sqsubseteq$. □

We lift the order $\leq$ on $\mathbb{N}$ to an order on matrices as follows. Let $n \in \mathbb{N}$ and $M, M' \in \mathbb{N}_+^{n\times n} \cup \{\underline{0}, \underline{1}\}$. Then

$$M \leq M' \quad \Longleftrightarrow \quad (\forall i, j \in [n]) : M_{ij} \leq M'_{ij} \ .$$

**Corollary 54.** *For every $n \in \mathbb{N}_+$, it is decidable whether a tree automaton $M$ with cost function over the square matrix semiring* $\mathrm{Mat}_n(\mathbb{N}_+) = (\mathbb{N}_+^{n\times n} \cup \{\underline{0}, \underline{1}\}, +, \cdot, \underline{0}, \underline{1})$ *is bounded with respect to $\leq$.*

*Proof.* It is easily seen that $\mathrm{Mat}_n(\mathbb{N}_+)$ is naturally ordered by $\leq$ and finitely factorizing as well as monotonic with respect to $\leq$. Hence Theorem 50 applies and we can decide whether $M$ is bounded with respect to $\leq$. □

Next we apply Theorem 50 in a slightly different manner because the partial order for monotonicity and the natural order will not coincide.

**Corollary 55.** *It is decidable whether a tree automaton $M$ with cost function over the finite-language semiring* $\mathrm{FLang}(\Sigma) = (\mathcal{P}_\mathrm{f}(\Sigma^*), \cup, \circ, \emptyset, \{\varepsilon\})$ *is bounded with respect to $\subseteq$.*

*Proof.* Clearly, $\mathrm{FLang}(\Sigma)$ is finitely factorizing and naturally ordered by $\subseteq$. On the other hand, it is not monotonic with respect to $\subseteq$. However, $\mathrm{FLang}(\Sigma)$ is monotonic, because it is monotonic with respect to the partial order $\preceq\ \subseteq \mathcal{P}_\mathrm{f}(\Sigma^*)^2$ defined by $L_1 \preceq L_2$ if and only if there exists an *injective* mapping $f : L_1 \longrightarrow L_2$, i.e., $w_1 \neq w_2$ implies $f(w_1) \neq f(w_2)$ for every $w_1, w_2 \in L_1$, such that for every $w_1 \in L_1$ we have that $w_1$ is a subword of $f(w_1)$. We leave the proof of monotonicity (with respect to $\preceq$) to the reader. Hence we can decide cost-finiteness of $M$ with respect to $\subseteq$ with the help of Theorem 50. □

**Corollary 56.** *It is decidable whether a tree automaton $M$ with cost function over the finite subsets semiring* $\mathrm{FSet}(\mathbb{N}) = (\mathcal{P}_\mathrm{f}(\mathbb{N}), \cup, +, \emptyset, \{0\})$ *of [31] is bounded with respect to $\subseteq$.*

*Proof.* Again $\mathrm{FSet}(\mathbb{N})$ is naturally ordered by $\subseteq$ and finitely factorizing, but not monotonic with respect to $\subseteq$. However we can again construct a partial order such that $\mathrm{FSet}(\mathbb{N})$ is monotonic. Consider the partial order $\preceq\ \subseteq \mathcal{P}_\mathrm{f}(\mathbb{N})^2$, which is defined by $N_1 \preceq N_2$, if and only if there exists an injective mapping $f : N_1 \longrightarrow N_2$ such that for every $n \in N_1$ we have that $n \leq f(n)$. Again we leave the detailed proof that $\mathrm{FSet}(\mathbb{N})$ is monotonic with respect to $\preceq$ to the reader and conclude that now Theorem 50 applies. Hence we can decide whether $M$ is bounded with respect to $\subseteq$. □

Let us mention that the above result is not the same as Theorem 3.19 of [31], in which the decidability of the finiteness of the set of cardinalities of the accepting sets (costs) is proved. We decide whether the set of all accepting sets is finite. If the set of accepting sets is finite, then clearly the set of cardinalities of the accepting sets is finite. However, the converse does not hold, e.g., the set of all singletons is a counter-example.

## 7. Conclusion

We have considered tree automata with costs over semirings. We have defined a big class of semirings, namely the class of finitely factorizing and monotonic semirings, and shown that the cost-finiteness of tree automata with costs over this semiring class is decidable. The key to this decidability results was the construction of the cost-equivalent reduced tree automaton $M'$ with costs for a given tree automaton $M$ with costs. We have shown that cost-finiteness and boundedness are equivalent for tree automata with costs over finitely factorizing and naturally ordered semirings. Hence it is also decidable whether a tree automaton with costs over a finitely factorizing, monotonic, and naturally ordered semiring is bounded with respect to the natural order. Here the partial order $\preceq$ of monotonicity and the partial order $\sqsubseteq$ of natural orderedness might be different. With this we have generalized the results of [31] concerning the decidability of the boundedness of tree automata with costs over the classical semiring Nat of natural numbers and the $(\max, +)$-semiring Arct of natural numbers.

In the whole paper we assumed that the semiring $\mathcal{A} = (A, \oplus, \odot, \mathbf{0}, \mathbf{1})$ is computable, however with a little more effort one can show that our statements (except Lemma 11(ii) and Lemma 19(ii)) are also valid provided that $\mathbf{1} \oplus \mathbf{1} = \mathbf{1}$ is decidable. Actually, in the paper we use Lemma 11(ii) only to decide whether $p \equiv \mathbf{0}$ or $p \equiv \mathbf{1}$; those conditions are decidable in arbitrary positive, one-summand free, and one-product free semirings (e.g., monotonic semirings), in which $\mathbf{1} \oplus \mathbf{1} = \mathbf{1}$ (i.e., idempotency) is decidable.

Furthermore, we believe that our method of proof can also be applied to monotonic and cancellative semirings (cancellative with respect to addition as well as multiplication), because they also enjoy the main property of finitely factorizing semirings (cf. Observation 2), namely for every $\otimes \in \{\oplus, \odot\}$ the result $A_1 \otimes A_2$ for non-empty set $A_1, A_2 \subseteq A$ is infinite provided that $A_1$ or $A_2$ is infinite (and $A_1 \neq \{\mathbf{0}\} \neq A_2$). A statement similar to Observation 3 clearly cannot be derived for cancellative semirings, but monotonic semirings are necessarily positive, such that the main statements concerning the decidability of finiteness should still be derivable for monotonic and cancellative semirings. However, the proved connection between the finiteness and boundedness problem (cf. Section 6) cannot be shown in this setting. Altogehter, this would nevertheless yield finiteness theorems for further interesting semirings such as, e.g., the real number semiring $\mathbb{R} = (\{0\} \cup \{\, n \in \mathbb{R} \mid 1 \leq n \,\}, +, \cdot, 0, 1)$ which is monotonic and cancellative.

## References


[1] J. Berstel, C. Reutenauer, Recognizable formal power series on trees. *Theoret. Comput. Sci.* **18** (1982) 2, 115–148.

[2] S. Bozapalidis, Equational elements in additive algebras. *Theory Comput. Syst.* **32** (1999) 1, 1–33.

[3] J. Doner, Decidability of the weak second-order theory of two successors. *Notices Amer. Math. Soc.* **12** (1965), Abstract No. 65T 648, 819.

[4] J. Doner, Tree acceptors and some of their applications. *J. Comput. System Sci.* **4** (1970), 406–451.

[5] J. Engelfriet, Bottom-up and top-down tree transformations - a comparison. *Math. Systems Theory* **9** (1975) 3, 198–231.

[6] J. Engelfriet, E. Schmidt, IO and OI.I. *J. Comput. System Sci.* **15** (1977) 3, 328–353.

[7] J. Engelfriet, E. Schmidt, IO and OI.II. *J. Comput. System Sci.* **16** (1978) 1, 67–99.

[8] M. Fitting, *First-Order Logic and Automated Theorem Proving*. Second edition, Graduate Texts in Computer Science, Springer, 1996.

[9] F. Gécseg, M. Steinby, *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.

[10] F. Gécseg, M. Steinby, Tree languages. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages*. 3. chapter 1, Springer-Verlag, 1997, 1–68.

[11] J. Golan, *Semirings and their Applications*. Kluwer Academic Publishers, Dordrecht, 1999.

[12] P. R. Halmos, *Naive Set Theory*. Springer, 1998.

[13] K. Hashiguchi, Limitedness theorem on finite automata with distance functions. *J. Comput. System Sci.* **24** (1982), 233–244.

[14] K. Hashiguchi, Regular languages of star height one. *Information and Control* **53** (1982) 3, 199–210.

[15] K. Hashiguchi, Algorithms for determining relative star height and star height. *Information and Computation* **78** (1988) 2, 124–169.

[16] K. Hashiguchi, Improved limitedness theorems on finite automata with distance functions. *Theoret. Comput. Sci.* **72** (1990), 27–38.

[17] K. Hashiguchi, New upper bounds to the limitedness of distance automata. *Theoretical Computer Science* **233** (2000) 1–2, 19–32.

[18] K. Hashiguchi, Erratum to: "New upper bounds to the limitedness of distance automata". *Theoretical Computer Science* **290** (2003) 3, 2183–2184.

[19] U. Hebisch, H. Weinert, *Semirings - Algebraic Theory and Applications in Computer Science*. World Scientific, Singapore, 1998.

[20] N. Jacobsen, *Basic Algebra I*. Second edition, W. H. Freeman and Company, New York, 1985.

[21] N. Jacobsen, *Basic Algebra II*. Second edition, W. H. Freeman and Company, New York, 1989.

[22] D. Kirsten, Desert automata and the finite substitution problem. In: V. Diekert, M. Habib (eds.), *21st Annual Symposium on Theoretical Aspects of Computer Science, STACS 2004, Montpellier, France, March 25–27, 2004, Proceedings*. Lecture Notes in Computer Science 2996, Springer, 2004, 305–316.

[23] D. Krob, The equality problem for rational series with multiplicities in the tropical semiring in undecidable. *International Journal of Algebra and Computation* **4** (1994) 3, 405–425.

[24] W. Kuich, Semirings and formal power series: Their relevance to formal languages and automata. In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages, Vol. 1*. chapter 9, Springer-Verlag, 1997, 609–677.

[25] W. Kuich, Formal power series over trees. In: S. Bozapalidis (ed.), *3rd International Conference on Developments in Language Theory, DLT 1997, Thessaloniki, Greece, Proceedings*. Aristotle University of Thessaloniki, 1998, 61–101.

[26] H. Leung, Limitedness theorem on finite automata with distance functions: An algebraic proof. *Theoretical Computer Science* **81** (1991) 1, 137–145.

[27] H. Leung, V. Podolskiy, The limitedness problem on distance automata: Hashiguchi's method revisited. *Theoretical Computer Science* **310** (2004) 1–3, 147–158.

[28] M. Magidor, G. Moran, *Finite Automata over Finite Trees*. Technical Report 30, Hebrew University, Jerusalem, 1969.

[29] M. D. Potter, *Sets: An Introduction*. first edition, Oxford University, New York, 1990.

[30] M. Schützenberger, On the definition of a family of automata. *Inf. and Control* **4** (1961), 245–270.

[31] H. Seidl, Finite tree automata with cost functions. *Theoret. Comput. Sci.* **126** (1994) 1, 113–142.

[32] J. Thatcher, Generalized$^2$ sequential machine maps. *J. Comput. System Sci.* **4** (1970) 4, 339–367.

[33] J. W. Thatcher, J. B. Wright, Generalized finite automata. *Notices of the American Mathematical Society* **12** (1965), Abstract No. 65T – 649, 820.

[34] A. Weber, Distance automata having large finite distance or finite ambiguity. *Mathematical Systems Theory* **26** (1993) 2, 169–185.

[35] A. Weber, Finite valued distance automata. *Theoretical Computer Science* **134** (1994) 1, 225–251.

[36] W. Wechler, *Universal Algebra for Computer Scientists*. Monogr. Theoret. Comput. Sci. EATCS Ser. 25, first edition, Springer-Verlag, Heidelberg/Berlin, 1992.

(Received: July 14, 2004; revised: January 13, 2005)