Weighted Automata and Expressions over Pre-Rational Monoids

Nicolas Baudru ORCID
Aix Marseille Univ, CNRS, LIS, Marseille, France

Louis-Marie Dando ORCID
Aix Marseille Univ, CNRS, LIS, Marseille, France

Nathan Lhote ORCID
Aix Marseille Univ, CNRS, LIS, Marseille, France

Benjamin Monmege ORCID
Aix Marseille Univ, CNRS, LIS, Marseille, France

Pierre-Alain Reynier
Aix Marseille Univ, CNRS, LIS, Marseille, France

Jean-Marc Talbot
Aix Marseille Univ, CNRS, LIS, Marseille, France

Abstract

The Kleene theorem establishes a fundamental link between automata and expressions over the free monoid. Numerous generalisations of this result exist in the literature; on one hand, lifting this result to a weighted setting has been widely studied. On the other hand, beyond the free monoid, different monoids can be considered: for instance, two-way automata, and even tree-walking automata, can be described by expressions using the free inverse monoid. In the present work, we aim at combining both research directions and consider weighted extensions of automata and expressions over a class of monoids that we call pre-rational, generalising both the free inverse monoid and graded monoids. The presence of idempotent elements in these pre-rational monoids leads in the weighted setting to consider infinite sums. To handle such sums, we will have to restrict ourselves to rationally additive semirings. Our main result is thus a generalisation of the Kleene theorem for pre-rational monoids and rationally additive semirings. As a corollary, we obtain a class of expressions equivalent to weighted two-way automata, as well as one for tree-walking automata.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory

Keywords and phrases Weighted Automata and Expressions, Inverse Monoids, Two-Way Automata

Digital Object Identifier 10.4230/LIPIcs.CSL.2022.22

Related Version A full version of the paper with additional examples and omitted proofs is available at [2], `https://arxiv.org/abs/2110.12395`.

Funding Supported by the DeLTA ANR project (ANR-16-CE40-0007)

1 Introduction

Automata are a convenient tool for algorithmically processing regular languages. However, when a short and human-readable description is required, regular expressions offer a much more proper formalism. When it comes to weighted automata (and transducers as a special case), the Kleene-Schützenberger theorem [20] relates weighted languages defined by means of such automata on one side, and rational series on the other side. Unfortunately, such expressions seem to fit mainly for one-way machines. Indeed, when it comes to two-way machines, finding adequate formalisms for expressions is not easy [13, 14].

Two-way automata have been studied in the setting of the Boolean semiring in [9]. In this work, Janin and Dicky consider a fragment of the free inverse monoid called overlapping tiles. They show that runs of a two-way automaton can be described as a recognizable language of overlapping tiles, which are words enriched with a starting and an ending position. Hence, thanks to the Kleene theorem, such two-way runs can be described as regular expressions (over tiles).

A particular class of weighted automata is that of transducers, where weights are words on an output alphabet. For this setting, Alur *et al* proposed in [1] a formalism to describe word transformations given as a deterministic streaming string transducer, a model equivalent with deterministic (or unambiguous) two-way transducers [12]. This formalism is based on some operators defining basic transformations that are composed to define the target transformation. An alternative construction of these expressions starting directly from two-way unambiguous transducers has been proposed in [3]. These expressions have also been extended to run on infinite words [8]. The general case of non-deterministic two-way transducers is much more challenging [13], as these machines may admit infinitely many accepting runs on an input word. While this general case is still open (meaning that no equivalent models of expressions are known), a solution has been proposed for the particular case where both input and output alphabets are unary [6].

For a further weighted generalisation, the ability to sum values computed by different runs on the same input structure (no matter if it is a word, a tree or even a graph) is also crucial in terms of expressiveness. However, not all weighted two-way automata (or weighted one-way automata with ε-transitions) are valid: indeed, as these machines may have infinitely many runs over a single input, it may be the case that the automaton does not provide any semantics for such inputs, infinite sums being not guaranteed to converge. To overcome this issue, additional properties are required over the considered semiring: for instance, *rationally additive semirings* [11] allow one to define valid non-deterministic two-way automata [15].

Our initial motivation was to elaborate on the approach proposed by Janin and Dicky in the setting of weighted languages. As already said, the main ingredient of their approach is to consider the free inverse monoid as an input structure. However, going one step further, we consider a generalisation, namely *pre-rational monoids*. These are monoids M such that for all finite alphabets A and for all morphisms from the free monoid A^* to M, the pre-image of $m \in M$ is a rational language of A^*. This class of monoids contains, in particular, the free inverse monoid. After introducing the monoids and semirings of interest in Section 2, we present our main contributions, which hold for pre-rational monoids and rationally additive semirings:

1. We prove in Section 3 that all weighted automata are valid.
2. We introduce in Section 4 a syntax for weighted expressions and show that the semantics of these expressions is always well-defined.
3. We prove in Section 5 a Kleene-like theorem stating that weighted automata and weighted expressions define the same series.
4. We deal with the particular case of unambiguous automata and expressions in Section 6. More precisely, our conversions are shown to preserve the ambiguity, meaning that an element of the monoid "accepted" k times by a weighted automaton can be "decomposed" in k different ways by the weighted expression we obtain, and vice versa.
5. In Section 7, we apply our results on two-way word automata and tree-walking automata which can be viewed as part of the free inverse monoids (which are pre-rational) and show how expressions are quite natural to write via a variety of examples. As a corollary, we obtain a formalism of expressions equivalent to non-deterministic two-way transducers

(relying on the unambiguity result presented in the previous section).

Our results can be understood as a trade-off between the generality of the monoid and that of the semiring. Indeed, instead of rationally additive semirings, one could have considered *continuous semirings* in which all infinite sums are well-defined. On such semirings, weighted automata are valid on all input monoids [19]. However, our framework allows one to consider semirings that are not continuous, and as a consequence we have to restrict in this case the input monoid. On the other end of the spectrum, restricting oneself to graded monoids (as also done in [19]) allows one to consider any semiring, since only finite sums are then involved. However, the free inverse monoid is a typical example of non-graded monoid.

2 Monoids and semirings

We recall that a *monoid* $(M, \cdot, \varepsilon_M)$ is given by a set M and an associative product $\cdot$ with ε_M as neutral element. For our purpose, we consider special classes of monoids:

▶ **Definition 1.** *A monoid $(M, \cdot, \varepsilon_M)$ is* pre-rational *if for every finite alphabet A, for every morphism $\mu\colon A^* \to M$, and for every $m \in M$, the language $\mu^{-1}(m) \subseteq A^*$ is rational.*

Many natural examples of monoids are pre-rational: the free monoid $(A^*, \cdot, \varepsilon)$ over a finite alphabet A, the natural monoid $(\mathbb{N}, +, 0)$, and even the one completed with an infinite element $(\mathbb{N} \cup \{+\infty\}, +, 0)$. Other examples, of particular interest in this article, are *free inverse monoids* that we study in Section 7. Another non-trivial example of pre-rational monoid is $(\{L \subseteq A^* \mid \varepsilon \in L\}, \cdot, \{\varepsilon\})$, with A a finite alphabet. In contrast, a typical example of monoid that is not pre-rational is the free group generated by one element, or $(\mathbb{Z}, +, 0)$ equivalently. For instance, given the morphism $\mu\colon \{a, \bar{a}\}^* \to \mathbb{Z}$ mapping a to 1 and $\bar{a}$ to -1, then $\mu^{-1}(0) = \{w \in \{a, \bar{a}\}^* \mid \ |w|_a = |w|_{\bar{a}}\}$ which is not rational.

Showing pre-rationality might sometimes be challenging, since considering arbitrary alphabets and arbitrary morphisms is not really convenient. An easier definition is however possible for monoids M that are generated by a finite family $G = \{g_1, \dots, g_n\}$ of generators. In this case, consider the canonical morphism φ from the free monoid G^* (considering generators as letters) to M, that consists in evaluating the sequence of generators in M. Then, M is pre-rational if and only if for all $m \in M$, the language $\varphi^{-1}(m) \subseteq G^*$ is rational. Pre-rationality is then easier to check, and this, without much of a restriction: the automata and expressions we will consider thereafter only use a finite set of elements of the monoid as atoms, and we can thus restrict ourselves to the finitely generated submonoid. An even simpler sufficient condition for pre-rationality is:

▶ **Lemma 2.** *If every element m of a monoid M has a finite number of* prefixes*, i.e. elements $p \in M$ such that there exists $p' \in M$ with $m = p \cdot p'$, then M is pre-rational.*

Proof. For a finite alphabet A and a morphism $\mu\colon A^* \to M$, and an element $m \in M$, with $\{m_1, \dots, m_n\}$ as finite set of prefixes, we can build a finite automaton reading letters of A and, after having read a word $w \in A^*$, storing the current element $\mu(w)$ when it is a prefix of m (going to a non-accepting sink state otherwise). This automaton can then be used to recognise $\mu^{-1}(m)$, by starting in the prefix ε_M and accepting in the prefix m. ◀

This allows us to easily show that all finitely generated graded monoids [19] (i.e. monoids M equipped with a gradation $\varphi\colon M \to \mathbb{N}$ such that $\varphi(m) = 0$ only if $m = \varepsilon_M$, and $\varphi(mn) = \varphi(m) + \varphi(n)$ for all $m, n \in M$) are pre-rational. Indeed, the gradation ensures that each element $m \in M$ can have only a finite number of prefixes [19, Chap. III, Cor. 1.2,p.384], allowing us to apply the previous lemma. However, notice that the condition in Lemma 2 is not a necessary one: $(\mathbb{N} \cup \{+\infty\}, +, 0)$ does not fulfil the condition, since $+\infty$ has infinitely many factors, while it is indeed pre-rational.

A *semiring* $(\mathbb{K}, +, \times, 0, 1)$ is an algebraic structure such that $(\mathbb{K}, \times, 1)$ is a monoid, $(\mathbb{K}, +, 0)$ is a commutative monoid, the product $\times$ distributes over the sum $+$, and 0 is absorbing for $\times$. Once again, we consider special classes of semirings, introduced in [11]:

▶ **Definition 3.** *A semiring* $(\mathbb{K}, +, \times, 0, 1)$ *is* rationally additive *if it is equipped with a partial operator defining sums of countable families, associating with some infinite families* $(\alpha_i)_{i\in I}$*, with* I *at most countable, an element* $\sum_{i\in I} \alpha_i$ *of* $\mathbb{K}$ *such that for all families* $(\alpha_i)_{i\in I}$*:*

- **Ax.1** *If* I *is finite, the value* $\sum_{i\in I} \alpha_i$ *exists and coincides with the usual sum in the semiring.*
- **Ax.2** *For each* $\alpha \in \mathbb{K}$, $\sum_{n=0}^{\infty} \alpha^n$ *exists.*
- **Ax.3** *If* $\sum_{i\in I} \alpha_i$ *exists and* $\beta \in \mathbb{K}$*, then* $\sum_{i\in I} \beta\alpha_i$ *and* $\sum_{i\in I} \alpha_i\beta$ *exist, and are respectively equal to* $\beta(\sum_{i\in I} \alpha_i)$ *and* $(\sum_{i\in I} \alpha_i)\beta$*.*
- **Ax.4** *Let* I *be the disjoint union of* $(I_j)_{j\in J}$ *with* J *at most countable. If for all* $j \in J$*,* $r_j = \sum_{i\in I_j} \alpha_i$ *exists, and if* $r = \sum_{j\in J} r_j$ *exists, then* $\sum_{i\in I} \alpha_i$ *exists and is equal to* r*.*
- **Ax.5** *Let* I *be the disjoint union of* $(I_j)_{j\in J}$ *with* J *at most countable. If* $s = \sum_{i\in I} \alpha_i$ *exists, and for all* $j \in J$*,* $r_j = \sum_{i\in I_j} \alpha_i$ *exists, then* $\sum_{j\in J} r_j$ *exists and is equal to* s*.*

Examples of rationally additive semirings are the Boolean semiring, natural semirings over positive rationals or reals $(\mathbb{Q}_+ \cup \{\infty\}, +, \times, 0, 1)$[1], the tropical (or arctic) semiring $(\mathbb{Q}\cup\{-\infty, +\infty\}, \sup, +, -\infty, 0)$, the language semiring over a finite alphabet $(2^{A^*}, \cup, \cdot, \emptyset, \{\varepsilon\})$, the sub-semiring of rational languages, or distributive lattices. Throughout this article, $\mathbb{K}$ will denote a rationally additive semiring.

Let us state a few useful properties of rationally additive semirings. The *support* of a family $(\alpha_i)_{i\in I}$ is the set $\{i \in I \mid \alpha_i \neq 0\}$ of indices of non-zero elements.

▶ **Lemma 4** **([11]).** *Let* $(\alpha_i)_{i\in I}$ *be a countable family in* $\mathbb{K}$*, of support* J*. Then,* $\sum_{i\in I} \alpha_i$ *exists if and only if* $\sum_{i\in J} \alpha_i$ *exists, and when these sums exist, they are equal.*

▶ **Lemma 5.** *Let* $(\alpha_i)_{i\in I}$ *and* $(\beta_i)_{i\in I}$ *be two countable families of* $\mathbb{K}$ *of disjoint supports, i.e. for all* $i \in I$*,* $\alpha_i = 0$ *or* $\beta_i = 0$ *(or both). If* $\sum_{i\in I} \alpha_i$ *and* $\sum_{i\in I} \beta_i$ *exist, then* $\sum_{i\in I}(\alpha_i+\beta_i)$ *exists and is equal to* $(\sum_{i\in I} \alpha_i) + (\sum_{i\in I} \beta_i)$*.*

Proof. Let J_α and J_β be the support of the families $(\alpha_i)_{i\in I}$ and $(\beta_i)_{i\in I}$, and $J_0 = J\backslash(J_\alpha \cup J_\beta)$. If $\sum_{i\in I} \alpha_i$ and $\sum_{i\in I} \beta_i$ exist, $\sum_{i\in I} \alpha_i + \sum_{i\in I} \beta_i$ exists, and by Lemma 4, is equal to $\sum_{i\in J_\alpha} \alpha_i + \sum_{i\in J_b} \beta_i$. Since the supports are disjoint, this is equal to $\sum_{i\in J_\alpha}(\alpha_i + \beta_i) + \sum_{i\in J_b}(\alpha_i + \beta_i)$. By definition of J_0, $\sum_{i\in J_0}(\alpha_i + \beta_i)$ exists and is equal to 0. Therefore, $\sum_{i\in I} \alpha_i + \sum_{i\in I} \beta_i$ is equal to $\sum_{i\in J_\alpha}(\alpha_i + \beta_i) + \sum_{i\in J_b}(\alpha_i + \beta_i) + \sum_{i\in J_0}(\alpha_i + \beta_i)$. **Ax.**4 allows us to conclude. ◀

▶ **Lemma 6.** *Let* $(\alpha_{i,j})_{(i,j)\in I\times J}$ *be a countable family of elements of* $\mathbb{K}$*, such that* $\alpha_{i,J} = \sum_{j\in J} \alpha_{i,j}$ *exists for all* $i \in I$*, and* $\alpha_{I,j} = \sum_{i\in I} \alpha_{i,j}$ *exists for all* $j \in J$*. Then,* $\sum_{i\in I} \alpha_{i,J}$ *exists if and only if* $\sum_{j\in J} \alpha_{I,j}$ *exists, and when these sums exists, they are equal.*

Proof. Immediate by **Ax.**4 and **Ax.**5. ◀

[1] All infinite sums of elements in $\mathbb{Q}_+$ do not converge towards a rational number or $+\infty$, but all *geometric sums* do. In particular, this semiring is not *continuous* (see [19, Chap. III, Sec. 5]).

3 Series and Weighted Automata

A $\mathbb{K}$-*series over* M is a mapping $s\colon M \to \mathbb{K}$ associating a weight $s(m)$ with each element m of the monoid. The set of all such series is denoted by $\mathbb{K}\langle\langle M\rangle\rangle$. Notice that the pointwise sum of two series s_1 and s_2, defined for all $m \in M$ by $(s_1 + s_2)(m) = s_1(m) + s_2(m)$, is a series. However, the Cauchy product $s_1 \cdot s_2$ mapping m to the possibly infinite sum $\sum_{m_1 m_2 = m} s_1(m_1) \times s_2(m_2)$ might not exist[2]. We define two canonical injections: $M \to \mathbb{K}\langle\langle M\rangle\rangle$ which maps m to the characteristic function of m (mapping m to 1 and the other elements from M to 0), and $\mathbb{K} \to \mathbb{K}\langle\langle M\rangle\rangle$ which maps k to the function mapping the neutral element ε_M of M to k and all other values to 0. For this reason, we often abuse notations and consider $\mathbb{K}$ and M as subsets of $\mathbb{K}\langle\langle M\rangle\rangle$.

We now introduce the notion of weighted automata we consider in this article: weights are taken from a rationally additive semiring $\mathbb{K}$ and *labels* from a pre-rational monoid M.

▶ **Definition 7.** *A* $\mathbb{K}$-automaton over M*, or simply a* weighted automaton*, is a tuple* $\mathcal{A} = (Q, I, \Delta, F)$*, with* Q *a finite set of states,* $I \subseteq Q$ *the set of initial states,* $\Delta \subseteq Q \times M \times \mathbb{K} \times Q$ *the finite set of transitions each equipped with a label in* M *and a weight in* $\mathbb{K}$*, and* $F \subseteq Q$ *the set of final states.*

We introduce two mappings $\lambda_{\mathcal{A}}$ and $\pi_{\mathcal{A}}$ that extract the label and the weight of a transition, that we can extend to morphisms from Δ^* to M and the multiplicative monoid of $\mathbb{K}$, respectively. A *run* of $\mathcal{A}$ is then a sequence w of transitions $(p_i, m_i, k_i, q_i)_{1\le i\le n}$ such that for all i, $q_i = p_{i+1}$. The *label* of a run is given by $\lambda_{\mathcal{A}}(w)$; its weight is $\pi_{\mathcal{A}}(w)$. The run is said to be *accepting* if $p_1 \in I$ and $q_n \in F$. We let $R_{\mathcal{A}} \subseteq \Delta^*$ denote the rational language of all accepting runs. The *semantics* of $\mathcal{A}$ is the series $[\![\mathcal{A}]\!]$ such that for all $m \in M$, the weight $[\![\mathcal{A}]\!](m)$ is the sum of the weights of accepting runs that are labelled by m, if the (potentially infinite) sum exists: $[\![\mathcal{A}]\!](m) = \sum_{w\in R_{\mathcal{A}}\cap\lambda_{\mathcal{A}}^{-1}(m)} \pi_{\mathcal{A}}(w)$.

The automaton $\mathcal{A}$ is called *valid* if the sum in $[\![\mathcal{A}]\!](m)$ exists for all $m \in M$. Instead of enforcing properties on the automata for them to be valid, we ensure their validity by combining the rational additivity of $\mathbb{K}$ and the pre-rationality of M. The crucial technical property considers the special case of the monoid of strings A^* over a finite alphabet A. We then lift the result using pre-rationality. For a language $L \subseteq A^*$ and a semiring $\mathbb{K}$, we denote by $\chi_L \in \mathbb{K}\langle\langle A^*\rangle\rangle$ its *characteristic series* in $\mathbb{K}$, defined for all $w \in A^*$ as $\chi_L(w) = 1$ if $w \in L$, and 0 otherwise. By Lemma 4, we have that for all series s over A^*,

$$\sum_{w\in L} s(w) \text{ is defined iff } \sum_{w\in A^*} s(w)\chi_L(w) \text{ is defined, and then these sums are equal.} \quad (1)$$

▶ **Lemma 8.** *For every finite alphabet* A*, morphism* $\pi\colon A^* \to \mathbb{K}$*, and rational language* $L \subseteq A^*$*, the sum* $\sum_{w\in L} \pi(w)$ *exists.*

Proof. The proof is by induction on rational languages, denoted by unambiguous regular expressions [5]. Indeed, all rational languages can be obtained by closing the set of finite languages by the operations of disjoint unions, unambiguous concatenations (the concatenation $L_1 \cdot L_2$ is unambiguous when each word w of $L_1 \cdot L_2$ can be uniquely decomposed as $w = w_1 \cdot w_2$ with $w_1 \in L_1$ and $w_2 \in L_2$), and unambiguous Kleene stars (the Kleene star L^* is unambiguous when each word $w \in L^*$ can be uniquely decomposed as $w = w_1 \cdots w_n$ with

[2] Here and in the following, $\sum_{m_1 m_2 = m}$ is the sum over all pairs $(m_1, m_2) \in M^2$ such that $m_1 m_2 = m$.

$n \in \mathbb{N}$ and $w_i \in L$ for all i). Please note that for convenience, the sentences "$A = B$" should be read as "B exists and is equal to A".

First, for finite languages L, the sum $\sum_{w\in L} \pi(w)$ exists, by **Ax.**1. In the case where L is the disjoint union of two languages L_1 and L_2, such that $\sum_{w\in L_1} \pi(w)$ and $\sum_{w\in L_2} \pi(w)$ exist,

$$\begin{aligned}
\sum_{w\in L_1} \pi(w) + \sum_{w\in L_2} \pi(w) &= \sum_{w\in A^*} \pi(w)\chi_{L_1}(w) + \sum_{w\in A^*} \pi(w)\chi_{L_2}(w) && \text{(by 1)}\\
&= \sum_{w\in A^*} (\pi(w)\chi_{L_1}(w) + \pi(w)\chi_{L_2}(w)) && \text{(by Lemma 5)}\\
&= \sum_{w\in A^*} \pi(w)\chi_{L_1\cup L_2}(w) && \text{(disjoint union)}\\
&= \sum_{w\in L_1\cup L_2 = L} \pi(w).
\end{aligned}$$

If L is the unambiguous concatenation of two languages L_1 and L_2 such that $\sum_{u\in L_1} \pi(u)$ and $\sum_{v\in L_2} \pi(v)$ exist, then

$$\begin{aligned}
\Big(\sum_{u\in L_1} \pi(u)\Big) \times \Big(\sum_{v\in L_2} \pi(v)\Big) &= \sum_{u\in L_1} \Big(\pi(u) \times \sum_{v\in L_2} \pi(v)\Big) && \text{(by } \mathbf{Ax.}3\text{)}\\
&= \sum_{u\in L_1} \sum_{v\in L_2} \pi(u)\pi(v) && \text{(by } \mathbf{Ax.}3\text{)}\\
&= \sum_{(u,v)\in L_1\times L_2} \pi(u)\pi(v) && \text{(by } \mathbf{Ax.}4\text{)}\\
&= \sum_{(u,v)\in L_1\times L_2} \pi(uv) && (\pi \text{ is a morphism}).
\end{aligned}$$

Moreover, by unambiguity, there exists a bijection from the pairs of $L_1 \times L_2$ to the words of the concatenation $L_1 \cdot L_2$ sending (u, v) to uv. Bijections on the support of families conserve the summability property by [11, Proposition 3], therefore $\sum_{w\in L} \pi(w)$ exists (and is equal to $\sum_{(u,v)\in L_1\times L_2} \pi(uv)$).

Finally, suppose that L is the unambiguous Kleene star L_1^*, and $\sum_{w\in L_1} \pi(w)$ exists. In particular, for all $n \in \mathbb{N}$, the iterated concatenation L_1^n is unambiguous, and thus, with a straightforward induction using the previous case, $\sum_{w\in L_1^n} \pi(w)$ exist and we have

$$\Big(\sum_{w\in L_1} \pi(w)\Big)^n = \sum_{w\in L_1^n} \pi(w).$$

By **Ax.**2, $\sum_{n=0}^{\infty} \Big(\sum_{w\in L_1} \pi(w)\Big)^n$ exists, and by (1), we have:

$$\sum_{n=0}^{\infty} \Big(\sum_{w\in L_1} \pi(w)\Big)^n = \sum_{n=0}^{\infty} \sum_{w\in L_1^n} \pi(w) = \sum_{n=0}^{\infty} \sum_{w\in A^*} \pi(w)\chi_{L_1^n}(w).$$

By unambiguity, for all $w \in A^*$, the infinite sum $\sum_{n=0}^{\infty} \pi(w)\chi_{L_1^n}(w)$ has finite support (at most 1 non-zero element) and therefore exists (by Lemma 4). By Lemma 6, we deduce that

$$\begin{aligned}
\sum_{n=0}^{\infty} \sum_{w\in A^*} \pi(w)\chi_{L_1^n}(w) = \sum_{w\in A^*} \sum_{n=0}^{\infty} \pi(w)\chi_{L_1^n}(w) &= \sum_{w\in A^*} \pi(w) \sum_{n=0}^{\infty} \chi_{L_1^n}(w) && \text{(by } \mathbf{Ax.}3\text{)}\\
&= \sum_{w\in A^*} \pi(w)\chi_{L_1^*}(w) && \text{(by unambiguity)}\\
&= \sum_{w\in L} \pi(w).
\end{aligned}$$

◀

From this result, to have a sufficient condition for validity we only need to have sums over rational languages, hence our requirement that M is pre-rational.

▶ **Theorem 9.** *If M is a pre-rational monoid, then every $\mathbb{K}$-automaton $\mathcal{A}$ over M is valid, i.e. $[\![\mathcal{A}]\!](m)$ exists for all $m \in M$.*

Proof. Since M is pre-rational, the morphism $\lambda_\mathcal{A}$ is such that for all $m \in M$, $\lambda_\mathcal{A}^{-1}(m)$ is a rational language. Therefore, so is the language $R_\mathcal{A} \cap \lambda_\mathcal{A}^{-1}(m)$ of accepting runs that are labelled by the element m. Lemma 8 gives that $[\![\mathcal{A}]\!](m) = \sum_{w \in R_\mathcal{A} \cap \lambda_\mathcal{A}^{-1}(m)} \pi_\mathcal{A}(w)$ exists. ◀

Together with reasonable assumptions on computability for $\mathbb{K}$ and M, this also gives a procedure to evaluate the weight $[\![\mathcal{A}]\!](m)$. Notice that this is a priori non-trivial, since it involves an infinite sum. We say that M is *effectively* pre-rational if for all morphisms $\mu \colon A^* \to M$ and elements $m \in M$, one can compute a representation of the rational language $\mu^{-1}(m)$. We say that $\mathbb{K}$ is *computable* if internal operations (finite sums and products) of $\mathbb{K}$ are computable, as well as Kleene star (geometric sum). Observe that we do not require computability of arbitrary infinite sums, but only geometric ones.

▶ **Proposition 10.** *If M is effectively pre-rational and $\mathbb{K}$ is computable, then for all $\mathbb{K}$-automata $\mathcal{A}$ over M and all elements $m \in M$, one can compute $[\![\mathcal{A}]\!](m)$. This computation is achieved using a number of internal operations of $\mathbb{K}$ (i.e. sum, product and Kleene iteration) that is polynomial in the size of $\mathcal{A}$ and in the size of a deterministic automaton recognising $\lambda_\mathcal{A}^{-1}(m)$.*

Proof. By assumption of pre-rationality, the language $\lambda_\mathcal{A}^{-1}(m)$ is rational. Moreover, by effectiveness, we can let $\mathcal{D}_m$ be a deterministic automaton that recognises $\lambda_\mathcal{A}^{-1}(m)$. We denote by n_m its number of states. The $\mathbb{K}$-automaton $\mathcal{A}_m$ obtained by considering the product of $\mathcal{A}$ and $\mathcal{D}_m$ (with respect to the alphabet Δ of transitions of $\mathcal{A}$) thus restricts the runs of $\mathcal{A}$ to the ones labelled by m. In addition, as $\mathcal{D}_m$ is deterministic, the accepting runs of $\mathcal{A}$ over m are in bijection with those of $\mathcal{A}_m$. If we denote by n the number of states of $\mathcal{A}$, then $\mathcal{A}_m$ has $n \times n_m$ states. By removing all labels (replacing them by ε_M), we obtain a $\mathbb{K}$-automaton that associates with the element ε_M the weight $[\![\mathcal{A}_m]\!](\varepsilon_M) = [\![\mathcal{A}]\!](m)$. Applying classical translations from automata to regular expressions such as state-elimination algorithms yields an expression equivalent to $[\![\mathcal{A}]\!](m)$. This expression involves sum and product in $\mathbb{K}$, as well as Kleene star, which can be computed in $\mathbb{K}$. As this expression only involves element ε_M, it can be evaluated during its computation, allowing to obtain the value of $[\![\mathcal{A}]\!](m)$ using a number of internal operations of $\mathbb{K}$ that is polynomial in n and n_m. ◀

4 Weighted Expressions

We now introduce the formalism of weighted expressions to generate $\mathbb{K}$-series over a monoid M.

▶ **Definition 11.** *The set of* $\mathbb{K}$-expressions *over M, or simply* weighted expressions*, is generated by the grammar (with $k \in \mathbb{K}$ and $m \in M$):*

$$W ::= k \mid m \mid W + W \mid W \cdot W \mid W^* .$$

Expressions k and m are said to be *atomic*. We call *subexpressions* of W all the weighted expressions appearing inside W: for instance, the subexpressions of $W = (2 \cdot a + b)^*$ are 2, a, b, $2 \cdot a$, $2 \cdot a + b$, and W. To define the semantics of weighted expressions, we will use a sum operator over infinite families. As the semiring $\mathbb{K}$ is supposed to be rationally additive, some of these infinite sums exist, some others do not[3]. Then, the semantics of a weighted expression W is the series $[\![W]\!] \in \mathbb{K}\langle\langle M \rangle\rangle$ defined inductively as follows:

- $[\![k]\!]$ is the series mapping ε_M to k and other elements to 0;
- $[\![m]\!]$ is the characteristic series of m;
- $[\![U+V]\!] = [\![U]\!] + [\![V]\!]$;
- for all $m \in M$, $[\![U \cdot V]\!](m) = \sum_{m_1 m_2 = m} [\![U]\!](m_1) \times [\![V]\!](m_2)$ if the sum exists;
- for all $m \in M$, $[\![W^*]\!](m) = \sum_{n=0}^{\infty} [\![W^n]\!](m)$ if the sum exists (with W^n the expression inductively defined by 1 if $n = 0$ and $W \cdot W^{n-1}$ otherwise).

The last two cases, defining the semantics of the concatenation (or Cauchy product) of two weighted expressions, and the Kleene star of a weighted expression, are subject to the existence of the infinite sums: we say that a weighted expression is *valid* when its semantics exists for all $m \in M$ (as well as the semantics of all its subexpressions, in particular).

More usual regular expressions are recovered by considering the Boolean semiring and the monoid A^* over a finite alphabet A: in the following, such expressions are called *Kleene expressions*, and denoted by letters E, F, G, while weighted expressions are denoted by letters U, V, W. Notice that Kleene expressions are valid, as expected, since the infinite *sum* (i.e. disjunction in the Boolean semiring) is always defined in this case. Their semantics $[\![E]\!]$ is the characteristic series of the language $\mathcal{L}(E)$ classically associated with such a regular expression: alternatively, we can see $\mathcal{L}(E)$ as the support of $[\![E]\!]$ (all words $w \in A^*$ such that $[\![E]\!](w)$ is true). For any other semiring $\mathbb{K}$, we let χ_E be the characteristic function of the language of E to the semiring $\mathbb{K}$, i.e. a shortcut notation for the series $\chi_{\mathcal{L}(E)} \in \mathbb{K}\langle\langle A^* \rangle\rangle$ defined in Section 3.

We shall see that thanks to our hypothesis of $\mathbb{K}$ being rationally additive, and restricting ourselves to pre-rational monoids, all weighted expressions are valid:

▶ **Theorem 12.** *Let $\mathbb{K}$ be a rationally additive semiring, and M be a pre-rational monoid. Every $\mathbb{K}$-expression W over M is valid,* i.e. *the semantics $[\![W]\!](m)$ exists for all $m \in M$.*

Notice that this theorem relies on both its assumptions on M and $\mathbb{K}$:

- If M is not pre-rational, then the expressions may not be valid. For instance, consider M to be the free group generated by a single element a (with a^{-1} its inverse in the free group), and $\mathbb{K}$ be the semiring of rational languages over the alphabet $\{A, B\}$. Then, the expression $(a \cdot \{A\} + a^{-1} \cdot \{B\})^*$ would associate with the element ε_M of M the language of words over $\{A, B\}$ having as many A's than B's, which is not rational, and thus not a member of $\mathbb{K}$.
- If $\mathbb{K}$ is not rationally additive, then the expressions may not be valid. For instance, considering the semiring $(\mathbb{Q}, +, \times, 0, 1)$, and the (pre-rational) trivial monoid $\{\varepsilon_M\}$, the expression $W = (-1)^*$ gives as a semantics $[\![W]\!](\varepsilon_M) = \sum_{n \in \mathbb{N}} (-1)^n$ that is the archetypal diverging series in $\mathbb{Q}$.

The rest of this section is devoted to the proof of this theorem. This proof is split into two parts. We first show that the validity of a weighted expression obtained by the rewriting of "letters" in an *unambiguous* Kleene expression is equivalent to the existence of sums resembling the ones of Lemma 8. We then explain how to generate such an unambiguous Kleene expression from a weighted expression W, and apply the previous result to show the validity of W.

[3] In the rationally additive semiring $(\mathbb{Q}_+ \cup \{\infty\}, +, \times, 0, 1)$, the infinite sum $\sum_{i \in \mathbb{N}} 1/i!$ does not exist, since it converges to the non-rational real number e.

More formally, a Kleene expression E (over a monoid A^*) is called unambiguous if for all its subexpressions E':

- if $E' = F + G$, then $\mathcal{L}(F) \cap \mathcal{L}(G) = \emptyset$;
- if $E' = F \cdot G$, then for all $w \in A^*$, there exists at most one pair $(w_1, w_2) \in \mathcal{L}(F) \times \mathcal{L}(G)$ such that $w_1 w_2 = w$;
- if $E' = F^*$, then for all $w \in A^*$, there exists at most one natural number n, and one sequence $(w_1, w_2, \dots, w_n) \in (\mathcal{L}(F))^n$ such that $w_1 w_2 \cdots w_n = w$.

As a direct corollary, for every semiring $\mathbb{K}$,

- if $E + F$ is unambiguous, then $\chi_{E+F} = \chi_E + \chi_F$;
- if $E \cdot F$ is unambiguous, then $\chi_{E \cdot F}(w) = \sum_{uv=w} \chi_E(u)\chi_F(v)$;
- if E^* is unambiguous, then $\chi_{E^*} = \sum_{n=0}^{\infty} \chi_{E^n}$, this infinite sum having indeed a finite support and being thus meaningful in any semiring (and formally existing in a rationally additive semiring).

Given two morphisms $\lambda\colon A^* \to M$ and $\pi\colon A^* \to \mathbb{K}$, we let $E_{\lambda,\pi}$ be the weighted expression obtained from a Kleene expression E by substituting every letter a appearing in E by the expression $\lambda(a) \cdot \pi(a)$, and by replacing Booleans *true* and *false* by elements $1 \in \mathbb{K}$ and $0 \in \mathbb{K}$.

The next lemma aims at linking the validity of $E_{\lambda,\pi}$ with the existence of specific infinite sums. The same result is also fundamental in our later proofs of translations between automata and expressions in the next section.

▶ **Lemma 13.** *Let E be an unambiguous Kleene expression over a free monoid A^*, M be a monoid (not necessarily pre-rational), $\mathbb{K}$ be a rationally additive semiring, $\lambda\colon A^* \to M$ and $\pi\colon A^* \to \mathbb{K}$ be two morphisms. Then, $E_{\lambda,\pi}$ is valid if and only if for all $m \in M$ and all subexpressions F of E, the sum $\sum_{\lambda(w)=m} \pi(w)\chi_F(w)$ exists (where the sum is over all words $w \in A^*$ such that $\lambda(w) = m$). In this case, for all $m \in M$, $[\![E_{\lambda,\pi}]\!](m) = \sum_{\lambda(w)=m} \pi(w)\chi_E(w)$.*

Starting from a weighted expression W, and in order to use Lemma 13 which only applies to unambiguous Kleene expressions, we will modify W to interpret it as an unambiguous Kleene expression. We define its *indexed expression* $I(W)$ as the Kleene expression over an alphabet being a finite subset of $(\mathbb{K} \cup M) \times \mathbb{N}$, obtained by replacing each of its atomic subexpression $\ell \in \mathbb{K} \cup M$ by a letter $(\ell, i) \in (\mathbb{K} \cup M) \times \mathbb{N}$ where i is a unique index (starting from 0 for the leftmost one) associated with each atomic subexpression. For instance, with the weighted expression $W = (2 \cdot a + 3 \cdot b)^* \cdot (a + 5 \cdot b + 3)$, one associates the indexed expression $I(W) = ((2,0) \cdot (a,1) + (3,2) \cdot (b,3))^* \cdot ((a,4) + (5,5) \cdot (b,6) + (3,7))$. From the indexed expression, one can recover the initial expression, by forgetting about the index. Formally, we let λ be the morphism from $((\mathbb{K} \cup M) \times \mathbb{N})^*$ to M such that $\lambda(x,n) = x$ if $x \in M$ and ε_M otherwise, and π be the morphism from $((\mathbb{K} \cup M) \times \mathbb{N})^*$ to $\mathbb{K}$ such that $\pi(x,n) = x$ if $x \in \mathbb{K}$ and 1 otherwise. For the above example, $I(W)_{\lambda,\pi} = ((\varepsilon_M \cdot 2) \cdot (a \cdot 1) + (\varepsilon_M \cdot 3) \cdot (b \cdot 1))^* \cdot ((a \cdot 1) + (\varepsilon_M \cdot 5) \cdot (b \cdot 1) + (\varepsilon_M \cdot 3))$, which is equivalent to W. More generally, we obtain:

▶ **Lemma 14.** *For all weighted expressions W over M, $I(W)_{\lambda,\pi}$ is valid if and only if W is valid. When valid, they have the same semantics.*

We would like to conclude by combining this result with Lemma 13 and by using the pre-rationality of the monoid M, as in Theorem 9. However, $I(W)$ might not be unambiguous as expected, as shown by the example $W = (m^*)^*$, with $m \in M$, that gives rise to the (ambiguous) Kleene expression $I(W) = (((m,0))^*)^*$: indeed, the word $(m,0)(m,0)$ has several possible decompositions in the semantics of $I(W)$. To patch this last issue, we simply incorporate a dummy marker after each Kleene star as follows: from a weighted expression W, $\phi(W)$ is inductively defined by:

- if W is an atomic expression, $\phi(W) = W$;
- if $W = U + V$ then $\phi(W) = \phi(U) + \phi(V)$;
- if $W = U \cdot V$ then $\phi(W) = \phi(U) \cdot \phi(V)$;
- if $W = U^*$ then $\phi(W) = (\phi(U))^* \cdot 1$, with 1 being the neutral element of the semiring $\mathbb{K}$.

We directly obtain:

▶ **Lemma 15.** *Let W be a weighted expression. The Kleene expression $I(\phi(W))$ is unambiguous.*

We are now ready to conclude the proof of Theorem 12, moreover showing that for all weighted expressions W and $m \in M$, $[\![W]\!](m) = \sum_{\lambda(w)=m} \pi(w)\chi_{I(\phi(W))}(w)$. Indeed, operation $\phi(\cdot)$ does not change the semantics of an expression, and therefore, $\phi(W)$ is valid if and only if W is valid, in which case they share the same semantics. Using the result of Lemma 15, we can apply Lemma 14: W is valid if and only if $I(\phi(W))_{\lambda,\pi}$ is valid, in which case they are equivalent. Let $L = \mathcal{L}(I(\phi(W))) \cap \lambda^{-1}(m)$. Since M is pre-rational, L is a rational language, and $\sum_{w \in L} \pi(w)$ exists. Moreover,

$$\begin{aligned}\sum_{w\in L} \pi(w) &= \sum_{\lambda(w)=m} \pi(w)\chi_{I(\phi(W))}(w) \\ &= [\![I(\phi(W))_{\lambda,\pi}]\!](m) && \text{(by Lemma 13)} \\ &= [\![\phi(W)]\!](m) && \text{(by Lemma 14)} \\ &= [\![W]\!](m) && (W \text{ and } \phi(W) \text{ are equivalent}).\end{aligned}$$

5 A Kleene-Like Theorem

Our main result is the following Kleene-like theorem, stating the constructive equivalence between expressions and automata over a pre-rational monoid and weighted over a rationally additive semiring.

▶ **Theorem 16.** *Let $\mathbb{K}$ be a rationally additive semiring, and M be a pre-rational monoid. Let $s \in \mathbb{K}\langle\langle M\rangle\rangle$ be a series. Then s is the semantics of some $\mathbb{K}$-automaton over M if and only if it is the semantics of some $\mathbb{K}$-expression over M.*

The rest of this section is devoted to the proof of this theorem, that consists in constructive translations of automata into equivalent expressions, and vice versa.

From Automata to Expressions. The idea is to build a $\mathbb{K}$-expression from an unambiguous expression generating the accepting runs of the automaton. Let $\mathcal{A} = (Q, \Delta, I, F)$ be a $\mathbb{K}$-automaton over M. By applying the result of [5], we build an unambiguous Kleene expression E over Δ^* recognising the language $R_\mathcal{A}$ of the accepting runs of $\mathcal{A}$. By Lemma 13, that we can apply on E since $E_{\lambda_\mathcal{A},\pi_\mathcal{A}}$ is valid (by Theorem 12), we have

$$[\![E_{\lambda_\mathcal{A},\pi_\mathcal{A}}]\!](m) = \sum_{\lambda_\mathcal{A}(w)=m} \pi_\mathcal{A}(w)\chi_E(w) = \sum_{w\in R_\mathcal{A} \mid \lambda_\mathcal{A}(w)=m} \pi_\mathcal{A}(w) = [\![\mathcal{A}]\!](m).$$

the second equality coming from (1), since $\mathcal{L}(E) = R_\mathcal{A}$.

From Expressions to Automata. We have shown in the previous section how, from a $\mathbb{K}$-expression E over M, we can construct an unambiguous Kleene expression $I(\phi(E))$ and two morphisms λ and π from[4] $(\mathbb{K} \cup M) \times \mathbb{N}$ to respectively M and $\mathbb{K}$, such that $I(\phi(E))_{\lambda,\pi}$ is equivalent to E, and by Theorem 12, $[\![E]\!](m) = \sum_{\lambda(w)=m} \pi(w)\chi_{I(\phi(E))}(w)$. We let $\{0,\ldots,n\}$ be the set of indices used in $I(\phi(E))$.

By [5], we can build (for instance, by considering the position automaton) from $I(\phi(E))$ an equivalent unambiguous Boolean automaton $\mathcal{A} = (Q, \Delta, I, F)$ with $\Delta \subseteq Q \times \big((\mathbb{K} \cup M) \times \mathbb{N}\big) \times Q$ its set of transitions labelled by indexed atomic elements appearing in E. Here, unambiguous means as usual that every accepted word in $\mathcal{A}$ is associated with a unique accepting run.

From $\mathcal{A}$, we build a $\mathbb{K}$-automaton $\mathcal{B} = (Q \times \{0,\ldots,n\}, \Delta', I \times \{0\}, F \times \{0,\ldots,n\})$ over M with transitions defined as follows: for all transitions $(p,(m,i),q) \in \Delta$, with $m \in M$, we add the transition $((p,j), m, 1, (q,i)) \in \Delta'$, and for all transitions $(p,(k,i),q) \in \Delta$, with $k \in \mathbb{K}$, we add the transition $((p,j), \varepsilon_M, k, (q,i)) \in \Delta'$. The transfer of indices from letters to states enables us to obtain a bijection f from accepted words of $\mathcal{A}$ to accepting runs of $\mathcal{B}$. Moreover, this bijection preserves the labels and weights, meaning that for all $u = (x_0, i_0) \cdots (x_m, i_m)$ accepted by $\mathcal{A}$, we have $\lambda(u) = \lambda_{\mathcal{B}}(f(u))$, and $\pi(u) = \pi_{\mathcal{B}}(f(u))$. Therefore, by applying the change of variable $w = f(u)$, we obtain

$$[\![\mathcal{B}]\!](m) = \sum_{w \in R_{\mathcal{B}} \cap \lambda_{\mathcal{B}}^{-1}(m)} \pi_{\mathcal{B}}(w) = \sum_{u \in \mathcal{L}(I(\phi(E))) \cap \lambda^{-1}(m)} \pi(u) = \sum_{\lambda(u)=m} \pi(u)\chi_{I(\phi(E))}(u) = [\![E]\!](m).$$

6 Dealing with Ambiguity

We have already encountered ambiguity in the context of the Boolean semiring and free monoids. We now study this notion for weighted expressions and automata. To do so, we use the rationally additive semiring $(\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}, +, \times, 0, 1)$ where all infinite sums exist: in particular, the sum over a family containing an infinite number of non-zero values is ∞, and otherwise the sum is equal to the finite sum over the support of the family. We call this semiring the *counting semiring*.

▶ **Definition 17.** *Given a $\mathbb{K}$-expression W over the monoid M, the* ambiguity $\mathrm{amb}(W,m)$ of W at m *is a value in $\mathbb{N}_\infty$ defined inductively over W as follows:*

- *for $W = n \in M$, $\mathrm{amb}(n,m) = 1$ if $n = m$, and 0 otherwise;*
- *for $W = k \in \mathbb{K}$, $\mathrm{amb}(k,m) = 1$ if $m = \varepsilon_M$, and 0 otherwise;*
- *for $W = U + V$, $\mathrm{amb}(U+V,m) = \mathrm{amb}(U,m) + \mathrm{amb}(V,m)$;*
- *for $W = U \cdot V$, $\mathrm{amb}(U \cdot V,m) = \sum_{m_1 m_2 = m} \mathrm{amb}(U,m_1) \times \mathrm{amb}(V,m_2)$;*
- *for $W = U^*$, $\mathrm{amb}(U^*,m) = \sum_{n \in \mathbb{N}} \mathrm{amb}(U^n,m)$.*

An expression is called unambiguous *if its ambiguity at every point is at most 1.*

For instance, the expression $W = 2 \cdot a + 3 \cdot a \cdot a$ over the free monoid $\{a\}^*$ is unambiguous, while W^* has ambiguity 2 at the word $aaa = a \cdot aa = aa \cdot a$.

The attentive reader may have noticed that the ambiguity of W is exactly the semantics of W where every atomic weight of $\mathbb{K}$ is replaced with the unit 1 of $\mathbb{N}_\infty$. Given two rationally additive semirings $\mathbb{K}_1$ and $\mathbb{K}_2$, $\mathbb{K}_1 \times \mathbb{K}_2$ is also a rationally additive semiring with the natural component-wise operations. In particular, given a $\mathbb{K}$-expression W, we can define a $\mathbb{K} \times \mathbb{N}_\infty$-expression W' by replacing every weight $k \in \mathbb{K}$ appearing in W by $(k,1) \in \mathbb{K} \times \mathbb{N}_\infty$. Then, the ambiguity of W at m is the second component of the weight $[\![W']\!](m)$.

[4] As before, in fact, we work with a finite subset of this set.

▶ **Definition 18.** *Given a $\mathbb{K}$-automaton $\mathcal{A}$ over the monoid M, the* ambiguity of $\mathcal{A}$ at m *is a value in $\mathbb{N}_\infty$ defined as the number (potentially ∞) of runs with label m. An automaton is called* unambiguous *if its ambiguity at every point is at most* 1.

Just as for expressions, the ambiguity of an automaton may be viewed as the semantics of the automaton where the weights of transitions are replaced by the unit of $\mathbb{N}_\infty$. Given $\mathcal{A}$ over $\mathbb{K}$, we can define $\mathcal{A}'$ by replacing all weights $k \in \mathbb{K}$ of transitions by $(k, 1) \in \mathbb{K} \times \mathbb{N}_\infty$. Then the ambiguity of $\mathcal{A}$ at m is exactly the second component of $[\![\mathcal{A}']\!](m)$. Now we claim:

▶ **Theorem 19.** *Let $\mathbb{K}$ be a rationally additive semiring, M be a pre-rational monoid, $s \in \mathbb{K}\langle\langle M\rangle\rangle$, and $k \in \mathbb{N}$. Then, s is the semantics of a $\mathbb{K}$-automaton over M of ambiguity k if and only if it is the semantics of a $\mathbb{K}$-expression over M of ambiguity k.*

Proof. The procedures of section 5 to go from expressions to automata and back, over a pre-rational monoid M, preserve ambiguity. Indeed, the two constructions used to prove Theorem 16 do not introduce new weights. Thus, starting from a $\mathbb{K}$-expression W, one considers the $\mathbb{K} \times \mathbb{N}_\infty$-expression W' defined above. Transforming W' into an automaton preserves the semantics, and all the transitions have a second component equal to 1. Thus, the second component of the semantics, which is preserved, is exactly the ambiguity of the automaton. Forgetting about the second component, we get the result. Note that converting W to W' is not actually a necessary step to build the automaton, it is simply a mental crutch to make the argument simpler. Symmetrically when going from automata to expressions, the transformation does not introduce new weights and thus preserves ambiguity. ◀

7 Free Inverse Monoids and Applications to Walking Automata

We conclude this article by demonstrating why our model is able to encompass and reason about the usual models of two-way automata and tree-walking automata. To do so, we consider the free inverse monoid, as it was observed by Pécuchet [18] to be linked with this model. Dicky and Janin even gave in [9, Theorem 3.21] the equivalence in the boolean case between two-way automata and regular expressions, using this monoid.

Let A be a finite alphabet, and $\overline{A} = \{\overline{a} \mid a \in A\}$ be a copy of A. We define the function $\dagger\colon (A \cup \overline{A})^* \to (A \cup \overline{A})^*$ inductively as: $\varepsilon^\dagger = \varepsilon$, $(ua)^\dagger = \overline{a}u^\dagger$, and $(u\overline{a})^\dagger = au^\dagger$.

▶ **Definition 20.** *The* free inverse monoid $\mathcal{I}(A)$ *generated by a finite alphabet A is the quotient of $(A \cup \overline{A})^*$ by the following equivalence relations:*

- *"$x^\dagger$ and x are pseudo-inverses": for all $x \in (A \cup \overline{A})^*$, $xx^\dagger x = x$, and $x^\dagger x x^\dagger = x^\dagger$;*
- *"idempotent elements commute": for all $x, y \in (A \cup \overline{A})^*$: $xx^\dagger yy^\dagger = yy^\dagger xx^\dagger$.*

Notice that $xx^\dagger$ are indeed idempotent elements of the free inverse monoid, since $(xx^\dagger)(xx^\dagger) = (xx^\dagger x)x^\dagger = xx^\dagger$.

The elements of this monoid are conveniently represented via tree-like structures, the Munn bi-rooted trees [17]. They are directed graphs, whose underlying undirected graph is a tree, and two special nodes are marked, the *initial* and the *final* one. Examples of elements of the monoid with their Munn tree representation are given in Figure 1. Note that if you see $a \in A$ as the traversal of an edge labelled by a, and $\overline{a}$ its traversal in reverse, an element of $(A \cup \overline{A})^*$ describes a complete walk over the graph of the corresponding element of $\mathcal{I}(A)$.

With this tree representation in mind, we see that every element of $\mathcal{I}(A)$ has finitely many prefixes, since such a prefix is a subtree of x, with the same initial node. Thanks to Lemma 2, we obtain

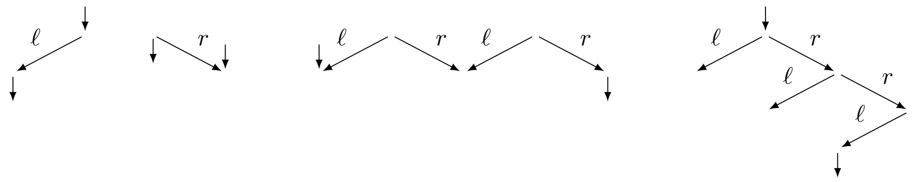

Figure 1 Munn bi-rooted trees of the elements of $\mathcal{I}(\{\ell, r\})$: ℓ, $\bar{r}$, $\bar{\ell}r\bar{\ell}r$, and $(\ell\bar{\ell}r)^2\ell$.

▶ **Proposition 21.** *The free inverse monoid is pre-rational.*

We can thus apply our results on this pre-rational monoid, for instance by considering expressions. In the Boolean semiring, for example, the expression $(\ell \cdot \bar{\ell} \cdot r)^* \cdot \ell$ describes the language of Munn bi-rooted trees that are "right-combs" (see the rightmost tree of Figure 1), when considering ℓ to be left children, and r right ones. The initial node is at the top while the final one is the farthest away from it. We can add weights to this expression: in the tropical semiring $(\mathbb{Z} \cup \{-\infty, +\infty\}, \sup, +, -\infty, 0)$, the unambiguous expression $(\ell \cdot \bar{\ell} \cdot r \cdot 1)^* \cdot \ell$ associates with a comb the length of its rightmost branch. More generally, the expression $W = \left[\sum_{a \in A} \left(a \cdot 1 + \bar{a} \cdot (-1)\right)\right]^*$ computes the (signed) length of the path linking the initial and final nodes in any Munn bi-rooted tree over alphabet A: each tree is associated with the difference between the number of positive letters of A and the number of negative letters of $\bar{A}$ of the unique acyclic path linking the initial node to the final node. On the trees of Figure 1, these lengths are respectively 1, -1, 0, 3. They represent the difference of "levels" in-between the initial and final nodes. Each tree is associated with many decompositions in the semantics of the expression W, but all of them have the same weight (and the chosen semiring has an idempotent sum operation).

Two-way Automata. Over an alphabet A, we can consider the free inverse monoid $\mathcal{I}(A \uplus \{\vdash, \dashv\})$, with two fresh symbols $\vdash$ and $\dashv$ that will help us distinguish the leftmost and rightmost letters of the word. To model two-wayness, only certain elements of $\mathcal{I}(A \uplus \{\vdash, \dashv\})$ are of interest, namely elements of $\vdash A^* \dashv$, that have linear Munn bi-rooted trees with the initial node at the leftmost position, and the final node at the rightmost one. The Munn bi-rooted tree representation of such an element is given in Figure 2.

We thus consider weighted automata and expressions over $\mathcal{I}(A)$ with weights in $\mathbb{K}$, a rationally additive semiring, and restrict our attention to words of $\vdash A^* \dashv$. From an automata perspective, this is a way to define the usual model of two-way automata, a forward movement of a two-way automaton being simulated by reading of a letter in A while a backward movement is simulated by reading a letter in $\bar{A}$. Indeed, our model of weighted automata over $\mathcal{I}(A)$ can also be simulated by the usual two-way weighted automata, since non-atomic elements of the monoid can be split into atomic elements. Therefore, in this specific context, Theorem 16 gives a new way to express the semantics of two-way weighted automata (over a rationally additive semiring) by using expressions.

Consider for example the function that maps a word $\vdash w \dashv$ with $w = w_0 \cdots w_{n-1} \in \{a, b\}^*$

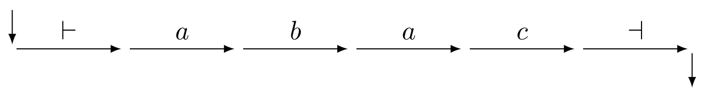

Figure 2 Munn bi-rooted tree of the "word" $\vdash abac\dashv$.

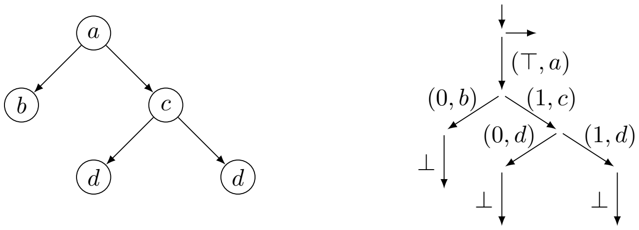

Figure 3 A binary tree, and its encoding in $\mathcal{I}(A')$.

to the set of words $\{(w_{n-1}\cdots w_0)^k \mid k \in \mathbb{N}\}$. Considering the semiring of regular languages, a weighted expression describing this function is

$$\left(\vdash \cdot (a+b)^* \cdot \dashv \cdot \overline{\dashv} \cdot (\overline{a} \cdot \{a\} + \overline{b} \cdot \{b\})^* \cdot \overline{\vdash}\right)^* \cdot \vdash \cdot (a+b)^* \cdot \dashv.$$

Notice the last pass over the word that allows one to finish the reading on the rightmost position, i.e. the final node.

Consider the alphabet $A = \{\mathbf{0}, \mathbf{1}\}$. For a word $w \in A^*$, let $w_{|2}$ denote the rational number between 0 and 1 that is written as $0.w$ in binary. Then, consider the following weighted expression with weights in $(\mathbb{Q}_+ \cup \{+\infty\}, +, \times, 0, 1)$:

$$W = \vdash \cdot \left(\mathbf{0} \cdot \frac{1}{2} + \mathbf{1} \cdot \frac{1}{2}\right)^* \cdot \mathbf{1} \cdot \frac{1}{2} \cdot (\mathbf{0} + \mathbf{1})^* \cdot \dashv.$$

It associates with a word $\vdash w \dashv$ the value $w_{|2}$, since it non-deterministically chooses a position i labelled by $\mathbf{1}$ in w and computes the value $1/2^i$. By considering the expression

$$(W \cdot \overline{\dashv} \cdot (\overline{\mathbf{0}} + \overline{\mathbf{1}})^* \cdot \overline{\vdash})^* \cdot W.$$

that consists in repeating the computation of W any number of times (at least once), with a reset of the word in-between, we associate with a word $\vdash w \dashv$ the value $\sum_{n=1}^{\infty} w_{|2}^n = w_{|2}/(1 - w_{|2})$.

Tree-Walking Automata. Another model captured by our approach is the one of tree-walking automata. These are automata whose head moves on the nodes of a rooted tree of a bounded arity m. As for words before, we can encode such trees labelled with a finite alphabet A by elements of $\mathcal{I}(A')$ with an extended alphabet $A' = (\{0, \ldots, m-1\} \cup \{\top\}) \times A \cup \{\bot\}$. In elements of $\mathcal{I}(A')$, nodes contain no information, only edges do. The idea is thus to simulate the root of a tree labelled with a by a single node labelled with $(\top, a)$; the i-th child of a node, labelled with $a \in A$, will be simulated with a node of label (i, a); finally, under each leaf of the tree, we add a node labelled with $\bot$. The root of the tree will be both the initial and the final node of the encoding, simulating a tradition of tree-walking automata to start and end in the root of the tree (without loss of generality).

As an example, consider the binary tree on the left of Figure 3. It is modelled by the following element of $\mathcal{I}(A')$, obtained from the Munn bi-rooted tree represented on the right by a depth-first search: $(\top, a)(0, b)\bot\overline{\bot}\,\overline{(0, b)}\,(1, c)(0, d)\bot\overline{\bot}\,\overline{(0, d)}\,(1, d)\bot\overline{\bot}\,\overline{(1, d)}\,\overline{(1, c)}\,\overline{(\top, a)}$.

When restricting the semantics of weighted automata and expressions to elements of $\mathcal{I}(A')$ that are encoding of trees, Theorem 16 gives an interesting model of weighted expressions equivalent to weighted tree-walking automata over rationally additive semirings.

The depth-first search of a tree is describable by an unambiguous weighted expression (and thus also an unambiguous weighted automaton): letting (i, A) denote $\sum_{a \in A}(i, a)$, and restricting ourselves to trees with nodes of arity 0 or 2 to simplify the writing, we let

$$W_0 = (0, A)^* \cdot \bot \ , \quad W_1 = \overline{\bot} \cdot \overline{(1, A)}^* \ , \quad \text{and} \quad W_{\text{succ}} = W_1 \cdot \overline{(0, A)} \cdot (1, A) \cdot W_0 .$$

The weighted expression W_0 finds the leftmost leaf; W_1 returns to the root from the rightmost leaf; and W_{succ} goes from a leaf to the next one in the depth-first search. Then, the depth-first search is implemented by the weighted expression $(\top, A) \cdot W_0 \cdot W_{\text{succ}}^* \cdot W_1 \cdot \overline{(\top, A)}$.

By Theorem 19, there exists an equivalent non ambiguous automaton, that thus visits the whole tree. Since it is possible to *reset* the tree, going back to the root, in a non ambiguous fashion, we can remove the requirement for the automata and the expressions to visit the whole tree while starting and ending at the root. This allows for more freedom in the models.

Taking advantage of this relaxation, it is possible to count the maximal number of occurrences of a letter a in branches of the tree, starting at the root of the tree, non-deterministically going down the chosen branch, and ending at the bottom: using the rationally additive semiring $(\mathbb{N} \cup \{-\infty, +\infty\}, \sup, +, -\infty, 0)$,

$$\big((\top, a) \cdot 1 + (\top, A \setminus \{a\})\big) \cdot \big((0, a) \cdot 1 + (0, A \setminus \{a\}) + (1, a) \cdot 1 + (1, A \setminus \{a\})\big)^* \cdot \bot .$$

8 Conclusion

We have given an application of our result to tree-walking automata. A natural extension consists in investigating other kinds of structure like Mazurkiewicz traces or grids.

Our approach *is* able to capture tree-walking automata, however it is intrinsically more of a tree-*generating* automaton model. Over trees it does not make a huge difference but it does if we try to extend this approach to more general graph-walking automata models. A natural way to define weighted automata over graphs is to take the sum of the weights of all paths over a given graph (in a sense already explored in [16], but limiting itself to non-looping runs). This means that a given path in the automaton can be a run in different graphs, which is not compatible with our approach of generating monoid elements.

One possible research direction would be to consider so-called SD-expressions introduced by Schützenberger (see [10]). These expressions were shown to coincide with star-free expressions with the advantage of not using the complement (instead restricting the languages over which the Kleene star can be applied, namely to prefix codes with bounded synchronisation delay) which means it can be applied to the quantitative setting. Indeed, in [7], the authors extended the result to transducers and showed that these expressions correspond to aperiodic transducers. These expressions are naturally adapted to the unambiguous setting (maybe this restriction can be overcome) but it would be interesting to study their expressive power in the context of pre-rational monoids.

A final direction would be to use logics instead of expressions, to describe in a less operational way the behaviour of weighted automata over monoids. Promising results have already been obtained in specific contexts, like non-looping automata walking (with pebbles) on words, trees or graphs [4], but a cohesive point of view via monoids is still lacking.

References

1. Rajeev Alur, Adam Freilich, and Mukund Raghothaman. Regular combinators for string transformations. In *CSL-LICS '14*, pages 9:1–9:10. ACM, 2014.
2. Nicolas Baudru, Louis-Marie Dando, Nathan Lhote, Benjamin Monmege, Pierre-Alain Reynier, and Jean-Marc Talbot. Weighted automata and expressions over pre-rational monoids, October 2021. `arXiv:2110.12395`.

3 Nicolas Baudru and Pierre-Alain Reynier. From two-way transducers to regular function expressions. *International Journal of Foundations of Computer Science*, 31(6):843–873, 2020.

4 Benedikt Bollig, Paul Gastin, Benjamin Monmege, and Marc Zeitoun. Logical characterization of weighted pebble walking automata. In Thomas A. Henzinger and Dale Miller, editors, *Proceedings of the joint meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, Vienna, Austria, July 2014. ACM. doi:10.1145/2603088.2603118.

5 R. Book, S. Even, Sheila A. Greibach, and G. Ott. Ambiguity in graphs and expressions. *IEEE Transactions on Computers*, 20(2):149–153, 1971.

6 Christian Choffrut and Bruno Guillon. An algebraic characterization of unary two-way transducers. In *MFCS 2014*, volume 8634 of *LNCS*, pages 196–207. Springer, 2014.

7 Luc Dartois, Paul Gastin, and Shankara Narayanan Krishna. SD-regular transducer expressions for aperiodic transformations. *CoRR*, arXiv:2101.07130, 2021. arXiv:2101.07130.

8 Vrunda Dave, Paul Gastin, and Shankara Narayanan Krishna. Regular transducer expressions for regular transformations. In *LICS 2018*, pages 315–324. ACM, 2018.

9 Anne Dicky and David Janin. Two-way automata and regular languages of overlapping tiles. *Fundamenta Informaticae*, 142:1–33, 2015.

10 Volker Diekert and Tobias Walter. Characterizing classes of regular languages using prefix codes of bounded synchronization delay. In *ICALP 2016*, volume 55 of *LIPIcs*, pages 129:1–129:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

11 Zoltán Ésik and Werner Kuich. Rationally additive semirings. *Journal of Universal Computer Science*, 8(2):173–183, 2002.

12 Emmanuel Filiot and Pierre-Alain Reynier. Transducers, logic and algebra for functions of finite words. *SIGLOG News*, 3(3):4–19, 2016.

13 Bruno Guillon. *Two-wayness: Automata and Transducers*. PhD thesis, Université Paris-Diderot and Universita di Milano, 2016.

14 Sylvain Lombardy. Two-way representations and weighted automata. *RAIRO - Theor. Inf. and Appl.*, 50(4):331–350, 2016. URL: http://dx.doi.org/10.1051/ita/2016026, doi:10.1051/ita/2016026.

15 Sylvain Lombardy and Jacques Sakarovitch. The validity of weighted automata. *Internat. J. Algebra Comput.*, 23:863–913, 2013.

16 Benjamin Monmege. *Specification and Verification of Quantitative Properties: Expressions, Logics, and Automata*. Phd, ENS Cachan, France, 2013.

17 Walter D Munn. Free inverse semigroups. *Proceedings of the London Mathematical Society*, 3(3):385–404, 1974.

18 Jean-Pierre Pécuchet. Automates boustrophedon, semi-groupe de birget et monoide inversif libre. *RAIRO Theor. Inf. and Appl.*, 19(1):71–100, 1985.

19 Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

20 Marcel-Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4:245–270, 1961.