From Hadamard expressions to weighted rotating automata and back

Louis-Marie Dando, Sylvain Lombardy

From Hadamard Expressions to Weighted Rotating Automata and Back

Louis-Marie Dando, Sylvain Lombardy

LaBRI UMR 5800, Université de Bordeaux, INP Bordeaux, CNRS, Bordeaux, France

Abstract

This paper deals with the conversion of expressions denoting Hadamard series into weighted rotating automata. We prove that any algorithm converting rational series into one-way weighted automata can be extended to provide an algorithm which achieves our goal. We apply this to define the derivation and the follow automata of a Hadamard expression. Our method may also be used to extend algorithms which perform the inverse conversion, but it is required to enhance the algorithm to fulfill some constraints.

Keywords: Weighted automata, Rotating automata, Hadamard product, Rational series

1. Introduction

Rotating automata are a natural extension of (one-way) automata. They have been introduced in [12]. Such an automaton can read its input several times as if this input were a cyclic word endowed with a marker to separate the last and the first letters. In the Boolean case, *i.e.* for unweighted rotating automata, accepted languages are regular languages, but they have been studied (*cf.* *e.g.* [7, 11]) since they can be much more succint than NFA, and simpler than two-way automata. In particular, the intersection of two languages recognised by such automata can be computed by first reading the input in the first automaton, and then in the second one: this can be realised with a linear number of states. These automata are sometimes introduced as restrictions of two-way automata, but we present them here as an extension of one-way automata, endowed with *rewinding* transitions that allow to come back at the beginning of the input.

Like two-way automata, rotating automata may have an infinite number of computations accepting a given finite word. This may lead to some issues in the definition of the behaviour of weighted rotating automata, since the weight of a word accepted by a weighted automaton is the sum of the weights of its accepting computations. In this paper, we focus on rotating automata with

Email addresses: louis-marie.dando@labri.fr (Louis-Marie Dando), sylvain.lombardy@labri.fr (Sylvain Lombardy)

weights in *rationally additive semirings* [5]. In this framework, the behaviour of rotating automata is always defined.

Moreover, the series realised by these automata are exactly the Hadamard series [9]. The Hadamard product of two series is the entrywise product, the Hadamard iteration (that is a pseudo-inverse of the Hadamard product, like the Kleene star is a pseudo-inverse of the Cauchy product) is the sum of the Hadamard powers. The set of Hadamard series is the closure of rational series under sum, Hadamard product and Hadamard iteration. If the semiring of coefficients is commutative, the Hadamard product of two rational series is rational, but it is no more the case if the coefficients are not commutative, and, even in the commutative case, the Hadamard iteration does not preserve the rationality of series (Example 14 in [2] shows a $\mathbb{Q}$-Hadamard series which is not rational).

This paper presents a generic framework to extend any algorithm that converts a rational expression to a (weighted) one-way automaton, into an algorithm that converts a Hadamard expression to a rotating automaton. More precisely, we show that every Hadamard expression E can be rewritten into a rational expression using a mapping ρ. The conversion of $\rho(\mathsf{E})$ into a (one-way) automaton by some algorithm σ can be interpreted as a rotating automaton (mapping r) that realises the series denoted by the original Hadamard expression E. These transformations are presented in Figure 4. The proof of our main theorem amounts to prove that this diagram – where $[\![.]\!]$ is the interpretation of expressions and $|\cdot|$ is the *behaviour* of automata – is actually commutative.

We then apply this result to two methods of automata synthesis: the *derivation automaton* and the *follow automaton*. We show that we can deduce rules from the general setting, that allow in each of these cases to directly handle Hadamard expressions.

The inverse conversion can also be extended, even if some extra conditions are required on the algorithm. A variant of the well-known State Elimination method is described; this variant is used in the core of an algorithm converting weighted rotating automata into Hadamard expressions.

2. Rational and Hadamard Formal Power Series

Let A be a finite alphabet and let A^* be the set of words over A, where ε denotes the empty word. A language over A is a subset of A^*.

A semiring is a set $\mathbb{K}$ endowed with two operations, sum and product; both operations are associative and have distinct neutral elements respectively denoted $0_{\mathbb{K}}$ and $1_{\mathbb{K}}$, the sum is commutative, the product is distributive over the sum and zero is an annihilator for product. In this paper, we consider *rationally additive semirings* as defined in [5].

Definition 1 ([5])**.** *A rationally additive semiring $\mathbb{K}$ is a semiring equipped with a partial summation defined on countable families such that:*

Ax1. *for finite families, the summation coincide with the sum of the semiring;*

Ax2. *for every element x of $\mathbb{K}$, the family of powers of x is summable and this sum is denoted x^* (the* star *of x);*

Ax3. *if $(s_i)_{i\in I}$ is a summable family, then for every x in $\mathbb{K}$, $(x.s_i)_{i\in I}$ and $(s_i.x)_{i\in I}$ are summable, and*

$$\sum_{i\in I} x.s_i = x(\sum_{i\in I} s_i), \text{ and } \sum_{i\in I} s_i.x = (\sum_{i\in I} s_i)x; \tag{1}$$

Ax4. *for every countable family $(s_i)_{i\in I}$, and every partition of I, $I = \cup_{j\in J} I_j$, if, for every j in J, $r_j = \sum_{i\in I_j} s_i$ is defined, and $r = \sum_{j\in J} r_j$ is defined, then $\sum_{i\in I} s_i$ is defined and equal to r.*

Ax5. *for every summable family $(s_i)_{i\in I}$, and every partition of I, $I = \cup_{j\in J} I_j$, if, for every j in J, $r_j = \sum_{i\in I_j} s_i$ is defined, then $r = \sum_{j\in J} r_j$ is defined and equal to $\sum_{i\in I} s_i$.*

Examples of rationally additive semirings are:

- the Boolean semiring $\mathbb{B}$;
- complete lattices;
- $\mathbb{Q}_+ \cup \{\infty\}$;
- the regular languages over a given alphabet;
- $(\mathbb{N} \cup \{\infty\}, \min, +)$;
- $([0;1], \max, .)$.

Notice that no ring is a rationally additive semiring. From [5], rationally additive semirings are Conway semirings, in particular, for every x it holds $x^* = 1_{\mathbb{K}} + x.x^*$. In a ring, if the star of $1_{\mathbb{K}}$ was defined, it would imply $0_{\mathbb{K}} = 1_{\mathbb{K}}$.

We consider the set $\mathbb{K}\langle\langle A^*\rangle\rangle$ of formal power series over A^* with coefficients in $\mathbb{K}$. A series s in $\mathbb{K}\langle\langle A^*\rangle\rangle$ is a mapping from A^* into $\mathbb{K}$; the coefficient of a word w in s is denoted $\langle s, w\rangle$, and s itself is denoted as a formal sum:

$$s = \sum_{w\in A^*} \langle s, w\rangle w. \tag{2}$$

The support of s is the language of words with a coefficient different from $0_{\mathbb{K}}$. If it is finite, s is a polynomial: $\mathbb{K}\langle A^*\rangle$ is the subset of polynomials of $\mathbb{K}\langle\langle A^*\rangle\rangle$. Different operations can be defined for series:

$$\begin{aligned}
\text{Sum: } s + t &= \sum_{w\in A^*} \left(\langle s, w\rangle + \langle t, w\rangle\right) w; \\
\text{Cauchy product: } s \cdot t &= \sum_{w\in A^*} \sum_{\substack{u,v\in A^* \\ uv=w}} \left(\langle s, u\rangle.\langle t, v\rangle\right) w; \\
\text{Hadamard product: } s \odot t &= \sum_{w\in A^*} \left(\langle s, w\rangle.\langle t, w\rangle\right) w.
\end{aligned} \tag{3}$$

The Hadamard and the Cauchy products are both associative operations which distribute over the sum. From [5], if $\mathbb{K}$ is a rationally additive semiring, so is the semiring $(\mathbb{K}\langle\langle A^*\rangle\rangle, +, \cdot)$; it is also true with $(\mathbb{K}\langle\langle A^*\rangle\rangle, +, \odot)$ since all operations are entrywise. The star of a series s in each of these semirings is respectively called the *Kleene star* denoted s^* and the *Hadamard iteration* denoted $s^{\circledast}$. Notice that, for every word w, $\langle s^{\circledast}, w\rangle = \langle s, w\rangle^*$.

The sum, the Cauchy product, and the Kleene star are called *rational operations*; the sum, the Hadamard product, and the Hadamard iterations are called *entrywise operations*.

Definition 2. *The set* $\mathbb{K}\mathrm{Rat}A^*$ *of* rational series *is the closure of the set of polynomials* $\mathbb{K}\langle A\rangle$ *by the rational operations. The set* $\mathbb{K}\mathrm{Had}A^*$ *of* Hadamard series *is the closure of* $\mathbb{K}\mathrm{Rat}A^*$ *by the entrywise operations.*

3. Weighted Rotating Automata

Rotating automata are one-way automata enhanced with rewinding transitions: when the input head reach the end of the input, if such a transition is available, it can go back to the beginning of the input. In our model, the computation can stop only if the input head is at the end of the input.

Definition 3. *Let* $\mathbb{K}$ *be a semiring and* A *an alphabet. A rotating* $\mathbb{K}$*-automaton over* A *is a tuple* (Q, E, R, I, T)*, where*
– Q *is a finite set of states;*
– $E : Q \times A \times Q \longrightarrow \mathbb{K}$ *is the transition function,*
– $R : Q \times Q \longrightarrow \mathbb{K}$ *is the rewinding function;*
– $I : Q \longrightarrow \mathbb{K}$ *is the initial function,*
– $T : Q \longrightarrow \mathbb{K}$ *is the final function.*
The set of rotating $\mathbb{K}$*-automata over* A *is denoted* $\mathsf{R}\mathbb{K}\mathsf{Aut}A$.

A state p is initial if $I(p) \neq 0$; it is final if $T(p) \neq 0$. In order to define the labels of computations and the behaviour of the automaton, we assume that there exists a special letter $\mathbf{r}$ which does not belong to A. In the sequel we denote $A_{\mathbf{r}} = A \cup \{\mathbf{r}\}$. A *transition* is a triple (p, a, q) in $Q \times A_{\mathbf{r}} \times Q$ such that either a is in A and $E(p, a, q) \neq 0$, or $a = \mathbf{r}$ and $R(p, q) \neq 0$. The letter a in $A_{\mathbf{r}}$ is the *label* of such a transition; for every transition (p, a, q), the *weight* $\mathsf{w}(p, a, q)$ of the transition is equal to $E(p, a, q)$ if a is in A, and to $R(p, q)$ if $a = \mathbf{r}$.

For every k in $\mathbb{N}$, a *path* π with length k is a triple $(r, (p_i, a_i, q_i)_{i\in[\![1;k]\!]}, s)$, such that:

- for every i in $[\![1; k]\!]$, (p_i, a_i, q_i) is a transition;
- if k is positive, $r = p_1$ and $s = q_k$, otherwise, $r = s$;
- for every i in $[\![1; k-1]\!]$, $q_i = p_{i+1}$.

The *label* of a path is the concatenation of labels of its transitions. A *computation* over a word w in A^* is a path $(r, (e_i)_{i\in[\![1;k]\!]}, s)$, where r is initial and s final, with a label in $(w\mathbf{r})^*w$.

The weight of a computation is defined as:

$$\mathsf{w}(r, (e_i)_{i\in[1;k]}, s) = I(r).\prod_{i=1}^{k}\mathsf{w}(e_i).T(s). \tag{4}$$

For every word w, the number of computations over w in a rotating $\mathbb{K}$-automaton is countable, potentially infinite. If C_w is the set of computations over w in $\mathcal{A}$, the weight of w in $\mathcal{A}$ is

$$\langle\mathcal{A}, w\rangle = \sum_{c\in C_w}\mathsf{w}(c), \tag{5}$$

if the sum is defined, otherwise, the weight of w is undefined.

Definition 4. *An automaton $\mathcal{A}$ is* valid *if for every word w, the weight of w in the automaton exists. In this case, the* behaviour *of $\mathcal{A}$ is the series*

$$|\mathcal{A}| = \sum_{w\in A^*}\langle\mathcal{A}, w\rangle w. \tag{6}$$

The following proposition can be deduced from the result in [9] that states that every two-way $\mathbb{K}$-automaton is valid, if $\mathbb{K}$ is a rationally additive semiring. We provide here a direct proof.

Proposition 1. *If $\mathbb{K}$ is a rationally additive semiring, every rotating $\mathbb{K}$-automaton is valid, and its behaviour is a Hadamard series.*

Proof. Let $\mathcal{A} = (Q, E, R, I, T)$ be a rotating $\mathbb{K}$-automaton, and let n be the size of Q. We identify Q with the interval $[\![1; n]\!]$.
Let w be a word in A^*, k in $\mathbb{N}$, and let π be a path in $\mathcal{A}$ with label $u = (w\mathbf{r})^k$. If k is equal to 0 or 1, we set $\mathsf{order}(\pi) = 0$; otherwise, for every j in $[\![0; k]\!]$, let s_j be the state met along π after reading $(w\mathbf{r})^j$; we set $\mathsf{order}(\pi)$ as the largest state in $(s_j)_{j\in[\![1;k-1]\!]}$ (we do not consider s_0 and s_k which are the first and the last states of π).
Let $\mathcal{P}(w, i, p, q)$ be the set of paths with label in $(w\mathbf{r})^*$ with order at most i that start in p and end in q, and let $S(w, i, p, q)$ be the sum of the weights of paths in $\mathcal{P}(w, i, p, q)$. We show by induction on i that, for every pair of states (p, q), $S(w, i, p, q)$ is defined.
If $\mathsf{order}(\pi) = 0$, the length of π is 0 or $|w| + 1$, hence $\mathcal{P}(w, 0, p, q)$ is finite for every pair (p, q). Assume that for every $j < i$, $S(w, j, p, q)$ is defined for every pair (p, q). If we consider the concatenation of paths as a (partial) product, the set $\mathcal{P}(w, i, p, q)$ can be unambiguously described as

$$\mathcal{P}(w, i, p, q) = \mathcal{P}(w, i-1, p, q) \cup \mathcal{P}(w, i-1, p, i).\mathcal{P}(w, i-1, i, i)^*.\mathcal{P}(w, i-1, i, q). \tag{7}$$

Since this description is unambiguous,

$$S(w,i,p,q) = S(w,i-1,p,q)+S(w,i-1,p,i).S(w,i-1,i,i)^*.S(w,i-1,i,q). \quad (8)$$

In a rationally additive semiring, the star is always defined, hence $S(i,p,q)$ is defined.
Let $F(w,p,q)$ be the sum of paths from p to q with label w (there is a finite amount of such paths). The weight of w in $\mathcal{A}$ is therefore defined and equal to

$$\sum_{p,q,r\in Q} I(p).S(w,n,p,q).F(w,q,r).T(r). \quad (9)$$

Let $s(i,p,q)$ be the series $\sum_{w\in A^*} S(w,i,p,q)$ and $f(p,q) = \sum_{w\in A^*} F(w,p,q)$. The series $s(0,p,q)$ and $f(p,q)$ are rational for every pair (p,q). Moreover, by Equation (8), $s(i,p,q) = s(i-1,p,q)+s(i-1,p,i)\odot s(i-1,i,i)^{\circledast}\odot s(i-1,i,q)$, for every i in $[\![1;n]\!]$. Thus, by induction, $s(n,p,q)$ is a Hadamard series. Hence, the behaviour of $\mathcal{A}$ is the Hadamard series:

$$\sum_{p,q,r\in Q} I(p).s(n,p,q)\odot f(q,r).T(r). \quad (10)$$

□

Proposition 2. *Every Hadamard series over $\mathbb{K}$ is the behaviour of a rotating $\mathbb{K}$-automaton.*

Proof. The Kleene-Schützenberger Theorem [13] states that every rational series is the behaviour of a one-way $\mathbb{K}$-automaton, that is a particular rotating $\mathbb{K}$-automaton. We prove that the set of series which are behaviours of rotating $\mathbb{K}$-automata is closed under the entrywise operations.

If X is a subset of Y and f is a function from X into $\mathbb{K}$, then f is naturally extended to Y with $f(y) = 0_{\mathbb{K}}$ for every $y \notin X$. If X and X' are two subsets of Y, and $f : X \to \mathbb{K}$ and $f' : X' \to \mathbb{K}$ are two functions, then, for every y in Y, $(f+f')(y) = f(y)+f'(y)$.

If s and t are respectively the behaviours of $\mathcal{A} = (Q,E,R,I,T)$ and $\mathcal{B} = (Q',E',R',I',T')$, then

- $s+t$ is the behaviour of the union of $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A}\cup\mathcal{B} = (Q\cup Q', E+E', R+R', I+I', T+T')$. For every word w, the set of runs over w is the disjoint union of the sets of runs over w in $\mathcal{A}$ and $\mathcal{B}$, thus the weight of w in the union is the sum of the weight of w in $\mathcal{A}$ and $\mathcal{B}$.
- $s\odot t$ is the behaviour of $\mathcal{A}\odot\mathcal{B} = (Q\cup Q', E+E', R+R'+S, I, T')$, with

  $$\forall (p,q)\in Q\times Q', S(p,q) = T(p).I'(q). \quad (11)$$

  For every word w, and every pair of runs $(p,(e_i),q)$ and $(p',(e'_j),q')$ over w in $\mathcal{A}$ and $\mathcal{A}'$, there is a run $(p,(f_k),q')$ in $\mathcal{A}\odot\mathcal{B}$, where the sequence (f_k) is the concatenation of (e_i), a rewind transition (q,p') and the sequence (e'_j); the weight of this run is the product of the weights of the runs in $\mathcal{A}$ and $\mathcal{B}$. Therefore $\mathcal{A}\odot\mathcal{B}(w) = \mathcal{A}(w).\mathcal{B}(w)$.

- $s^{\circledast}$ is the behaviour of $\mathcal{A}^{\circledast} = (Q \cup \{i\}, E + F, R + S, I + \chi_i, T + \chi_i)$, with

$$\forall a \in A, F(i,a,i) = 1_{\mathbb{K}},\ \forall (p,q) \in Q, S(p,q) = T(p).I(q),\ \chi_i(i) = 1_{\mathbb{K}}. \quad (12)$$

For every word w, and every list of runs $(p^{(j)}, (e_i^{(j)}), q^{(j)})_{j\in[1;r]}$ over w in $\mathcal{A}$, there is a run $(p^{(1)}, (f_k), q^{(r)})$ in $\mathcal{A}^{\circledast}$, where the sequence (f_k) is the concatenation of the sequences $(e_i^{(j)})$ separated by rewind transitions $(q^{(j)}, p^{(j+1)})$; the weight of this run is the product of the weights of the runs in $\mathcal{A}$. Therefore $\mathcal{A}^{\circledast}(w) = \mathcal{A}(w)^*$.

□

We now study a link between rotating and one-way automata, which are, in our framework, the subclass of rotating $\mathbb{K}$-automata that do not contain any rewinding transition. $1\mathbb{K}\mathsf{Aut}A$ is the set of one-way $\mathbb{K}$-automata over A, and every automaton in $1\mathbb{K}\mathsf{Aut}A$ is characterised by a tuple (Q, E, I, T).

Moreover, an automaton $\mathcal{A} = (Q, E, R, I, T)$ in $\mathsf{R}\mathbb{K}\mathsf{Aut}A$ can be considered as a one-way $\mathbb{K}$-automaton $\mathsf{1w}(\mathcal{A}) = (Q, E', I, T)$, over the alphabet $A_{\mathbf{r}}$, with $E' = E \cup \{(p, \mathbf{r}, q) \mapsto R(p,q) \mid (p,q) \in Q^2\}$. Conversely, if $\mathcal{A}$ is in $1\mathbb{K}\mathsf{Aut}A_{\mathbf{r}}$, $\mathsf{rot}(\mathcal{A})$ is the corresponding automaton in $\mathsf{R}\mathbb{K}\mathsf{Aut}A$, obtained by replacing every transition with label $\mathbf{r}$ by a rewinding transition.

The canonical bijection between transitions of $\mathcal{A}$ and $\mathsf{1w}(\mathcal{A})$ extends to paths, and every pair of corresponding paths has the same label and the same weight. In particular, for every word w in A^*, every computation in $\mathcal{A}$ over w corresponds to a computation in the automaton $\mathsf{1w}(\mathcal{A})$ over a word u in $(w\mathbf{r})^*w$.

To characterise the behaviour of the rotating $\mathbb{K}$-automaton $\mathcal{A}$ with respect to the behaviour of the one-way $\mathbb{K}$-automaton $\mathsf{1w}(\mathcal{A})$, we define a linear function from $\mathbb{K}\langle\langle A_{\mathbf{r}}^*\rangle\rangle$ to $\mathbb{K}\langle\langle A^*\rangle\rangle$:

$$\forall s \in \mathbb{K}\langle\langle A_{\mathbf{r}}^*\rangle\rangle, \qquad \varphi\Big(\sum_{u\in A_{\mathbf{r}}^*} \langle s,u\rangle u\Big) = \sum_{w\in A^*}\Big(\sum_{u\in(w\mathbf{r})^*w} \langle s,u\rangle\Big) w. \qquad (13)$$

Notice that φ is partially defined; for every series s, $\varphi(s)$ is defined if and only if, for every word w, the family $(\langle s,u\rangle)_{u\in(w\mathbf{r})^*w}$ is summable.

The following proposition naturally follows:

Proposition 3. *Let $\mathbb{K}$ be a rationally additive semiring. For every automaton $\mathcal{A}$ in $\mathsf{R}\mathbb{K}\mathsf{Aut}A$, $\varphi(|\mathsf{1w}(\mathcal{A})|)$ is defined and equal to $|\mathcal{A}|$.*

Proof. Let $\mathcal{A}$ be an automaton in $\mathsf{R}\mathbb{K}\mathsf{Aut}A$, and let $\mathcal{B} = \mathsf{1w}(\mathcal{A})$. By proposition 1, $\mathcal{A}$ is valid; this means that the sum of weights of runs in $\mathcal{B}$ labeled by a word in $w(rw)^*$ is defined. Moreover, for every word u in $w(rw)^*$ the sum $\langle s,u\rangle$ of the weights of the (finite) set of runs in $\mathcal{B}$ with label u is defined. Hence, by Axiom 5 of rationally additive semirings, the sum of $\langle s,u\rangle$ over all words u in $w(rw)^*$ is defined and equal to the sum of the weights of runs in $\mathcal{A}$ with label w. □

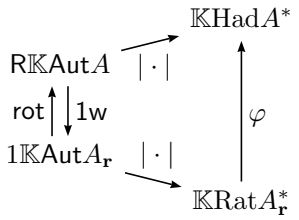


Figure 1: Proposition 3 as a commutative diagram.

Corollary 1. *The image of φ restricted to $\mathbb{K}\mathrm{Rat}A^*_{\boldsymbol{r}}$ is the set $\mathbb{K}\mathrm{Had}A^*$.*

Proof. By Kleene-Schützenberger Theorem, every series in $\mathbb{K}\mathrm{Rat}A^*_{\mathbf{r}}$ is the behaviour of an automaton $\mathcal{A}$ in $1\mathbb{K}\mathsf{Aut}A_{\mathbf{r}}$. The image of this series by φ is the behaviour of $\mathsf{rot}(\mathcal{A})$, which is a Hadamard series.

Conversely, every Hadamard series s is the behaviour of a rotating automaton $\mathcal{B}$. The image of the (rational) behaviour of $\mathsf{1w}(\mathcal{B})$ by φ is equal to s. □

4. Hadamard Expressions

Like rational series, Hadamard series are naturally denoted by expressions. In the following grammar, where A is an alphabet, R generates $\mathbb{K}$-rational expressions in $\mathbb{K}\mathsf{RatExp}A$ and H generates $\mathbb{K}$-Hadamard expressions in $\mathbb{K}\mathsf{HadExp}A$.

$$\begin{aligned} R &\to 0 \mid 1 \mid a \in A \mid kR, k \in \mathbb{K} \mid Rk, k \in \mathbb{K} \mid R + R \mid RR \mid R^*, \\ H &\to R \mid kH, k \in \mathbb{K} \mid Hk, k \in \mathbb{K} \mid H + H \mid H \odot H \mid H \circledast H. \end{aligned} \tag{14}$$

As usual, parentheses can be added to prevent ambiguity. In the following, the classical rules of priority apply. Notice that we use in the Hadamard expressions a binary iteration operator that appears to be more suitable with the definitions below (definition of ρ in particular). It is straightforward that the expressiveness of this operator is equivalent to a unary star operator: if E denotes the series s, $s^{\circledast}$ is denoted by $\mathsf{E} \circledast (a_1 + \ldots + a_n)^*$, where $A = \{a_1, \ldots a_n\}$. Besides, the original Kleene star operator in [8] was binary.

Definition 5. *The interpretation of an expression in $\mathbb{K}\mathsf{HadExp}A$ is a series in $\mathbb{K}\mathrm{Had}A^*$ inductively defined by :*

$$\begin{gathered} [\![0]\!] = 0_{\mathbb{K}},\ [\![1]\!] = 1_{\mathbb{K}}, \qquad \forall a \in A,\ [\![a]\!] = a, \\ \forall k \in \mathbb{K},\ [\![k\mathsf{F}]\!] = k[\![\mathsf{F}]\!],\ [\![\mathsf{F}k]\!] = [\![\mathsf{F}]\!]k, \quad [\![\mathsf{F} + \mathsf{G}]\!] = [\![\mathsf{F}]\!] + [\![\mathsf{G}]\!], \\ [\![\mathsf{FG}]\!] = [\![\mathsf{F}]\!] \cdot [\![\mathsf{G}]\!], \qquad [\![\mathsf{F}^*]\!] = [\![\mathsf{F}]\!]^*, \\ [\![\mathsf{F} \odot \mathsf{G}]\!] = [\![\mathsf{F}]\!] \odot [\![\mathsf{G}]\!], \quad [\![\mathsf{F} \circledast \mathsf{G}]\!] = [\![\mathsf{F}]\!]^{\circledast} \odot [\![\mathsf{G}]\!]. \end{gathered} \tag{15}$$

If the interpretation of E in $\mathbb{K}\mathsf{HadExp}A$ is defined, for every w in A^*, we set $\langle \mathsf{E}, w \rangle = \langle [\![E]\!], w \rangle$.

In Section 3, we showed that every Hadamard series can be realised as the image by φ of the behaviour of a one-way automaton. In this section, we describe a formal inverse of φ: this is a syntactic transformation ρ which turns an expression E in $\mathbb{K}\mathsf{HadExp}A$ into an expression in $\mathbb{K}\mathsf{RatExp}A_{\mathbf{r}}$, inductively defined by:

$$\begin{gathered}\rho(\mathsf{E}) = \mathsf{E} \text{ if } \mathsf{E} \in \mathbb{K}\mathsf{RatExp}A,\\ \forall k \in \mathbb{K},\ \rho(k\mathsf{E}) = k\rho(\mathsf{E}),\ \rho(\mathsf{E}k) = \rho(\mathsf{E})k, \qquad \rho(\mathsf{F}+\mathsf{G}) = \rho(\mathsf{F}) + \rho(\mathsf{G}),\\ \rho(\mathsf{F} \odot \mathsf{G}) = \rho(\mathsf{F})\mathbf{r}\rho(\mathsf{G}), \qquad \rho(\mathsf{F} \circledast \mathsf{G}) = (\rho(\mathsf{F})\mathbf{r})^*\rho(\mathsf{G}).\end{gathered} \tag{16}$$

Proposition 4. *Let* E *be a Hadamard expression. It holds* $[\![\mathsf{E}]\!] = \varphi([\![\rho(\mathsf{E})]\!])$.

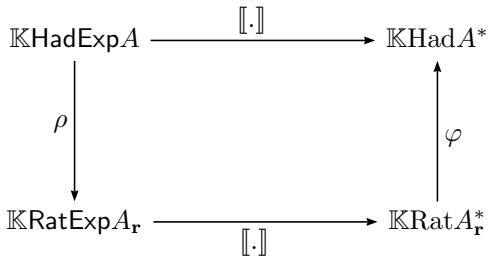


Figure 2: Proposition 4 as a commutative diagram.

Proposition 4 is based on the following lemma. The proof involves some summation exchanges which are licit in rationally additive semirings; using this result in other semirings may require a new proof.

Lemma 1. *Let* E *be an expression in* $\mathbb{K}\mathsf{HadExp}A$. *Then, for every word* w *in* A^*,

$$\langle \mathsf{E}, w\rangle = \sum_{i=0}^{\infty} \langle \rho(\mathsf{E}), (w\boldsymbol{r})^i w\rangle. \tag{17}$$

Proof. The proof is by induction on E.
If E is a rational expression, $\rho(\mathsf{E}) = \mathsf{E}$, hence,

$$\sum_{i=0}^{\infty} \langle \rho(\mathsf{E}), (w\mathbf{r})^i w\rangle = \sum_{i=0}^{\infty} \langle \mathsf{E}, (w\mathbf{r})^i w\rangle = \langle \mathsf{E}, w\rangle. \tag{18}$$

Assume that the result is true for two Hadamard expressions F and G;

- if $\mathsf{E} = \mathsf{F} + \mathsf{G}$, $\mathsf{E} = k\mathsf{F}$ or $\mathsf{E} = \mathsf{F}k$ the result holds by linearity;

- if $\mathsf{E} = \mathsf{F} \odot \mathsf{G}$, then

$$\begin{aligned}
\langle \mathsf{E}, w\rangle &= \langle \mathsf{F} \odot \mathsf{G}, w\rangle \\
&= \langle \mathsf{F}, w\rangle\langle \mathsf{G}, w\rangle = \left(\sum_{i=0}^{\infty}\langle \rho(\mathsf{F}), (w\mathbf{r})^i w\rangle\right)\left(\sum_{j=0}^{\infty}\langle \rho(\mathsf{G}), (w\mathbf{r})^j w\rangle\right) \\
&= \sum_{i,j\in\mathbb{N}^2}\langle \rho(\mathsf{F}), (w\mathbf{r})^i w\rangle\langle \rho(\mathsf{G}), (w\mathbf{r})^j w\rangle \\
&= \sum_{k=0}^{\infty}\sum_{n=0}^{k}\langle \rho(\mathsf{F}), (w\mathbf{r})^{k-n} w\rangle\langle \rho(\mathsf{G}), (w\mathbf{r})^n w\rangle \\
&= \sum_{k=0}^{\infty}\langle \rho(\mathsf{F})\mathbf{r}\rho(\mathsf{G}), (w\mathbf{r})^{k+1} w\rangle \\
&= \sum_{i=0}^{\infty}\langle \rho(\mathsf{E}), (w\mathbf{r})^i w\rangle
\end{aligned} \tag{19}$$

- if $\mathsf{E} = \mathsf{F} \circledast \mathsf{G}$, $\langle \mathsf{E}, w\rangle = \langle \mathsf{F} \circledast \mathsf{G}, w\rangle = \langle \mathsf{F}, w\rangle^*\langle G, w\rangle$.
  We consider an extension of rational and Hadamard expressions.
  For every integer k, if E is a rational expression, E^k is an extended rational expression with interpretation $[\![\mathsf{E}^k]\!] = [\![\mathsf{E}]\!]^k$.
  Likewise, if E is a Hadamard expression, $\mathsf{E}^{\odot k}$ is an extended Hadamard expression such that, for every word w, $\langle \mathsf{E}^{\odot k}, w\rangle = \langle \mathsf{E}, w\rangle^k$. By induction, it holds, for every k,

$$\langle \mathsf{E}^{\odot k}, w\rangle = \langle \mathsf{E}, w\rangle^k = \sum_{i=0}^{\infty}\langle (\rho(\mathsf{E})r)^k, (w\mathbf{r})^i\rangle\,. \tag{20}$$

  Therefore,

$$\begin{aligned}
\langle \mathsf{F}, w\rangle^* &= \sum_{k=0}^{\infty}\langle \mathsf{F}, w\rangle^k = \sum_{k=0}^{\infty}\langle \mathsf{F}^{\odot k}, w\rangle \\
&= \sum_{k=0}^{\infty}\sum_{i=0}^{\infty}\langle (\rho(\mathsf{F})\mathbf{r})^k, (wr)^i\rangle \\
&= \sum_{i=0}^{\infty}\sum_{k=0}^{\infty}\langle (\rho(\mathsf{F})\mathbf{r})^k, (wr)^i\rangle = \sum_{i=0}^{\infty}\langle (\rho(\mathsf{F})\mathbf{r})^*, (wr)^i\rangle\,.
\end{aligned} \tag{21}$$

Finally,

$$\begin{aligned}
\langle \mathsf{E}, w\rangle &= \sum_{i=0}^{\infty} \langle (\rho(\mathsf{F})\mathbf{r})^*, (wr)^i\rangle \sum_{j=0}^{\infty} \langle \rho(\mathsf{G}), w(rw)^j\rangle \\
&= \sum_{i,j\in\mathbb{N}^2} \langle (\rho(\mathsf{F})\mathbf{r})^*, (wr)^i\rangle \langle \rho(\mathsf{G}), w(rw)^j\rangle \\
&= \sum_{n=0}^{\infty}\sum_{k=0}^{n} \langle (\rho(\mathsf{F})\mathbf{r})^*, (wr)^{n-k}\rangle \langle \rho(\mathsf{G}), w(rw)^k\rangle \qquad (22)\\
&= \sum_{n=0}^{\infty} \langle (\rho(\mathsf{F})\mathbf{r})^* \rho(\mathsf{G}), w(rw)^n\rangle \\
&= \sum_{n=0}^{\infty} \langle \rho(\mathsf{E}), w(rw)^n\rangle \,.
\end{aligned}$$

Notice that the handling of infinite sums in this proof are licit thanks to the axioms of rationally additive semirings [5]. □

The proof of Proposition 4 follows:

Proof of Proposition 4. We have $[\![\rho(\mathsf{E})]\!] = \sum_{u\in A^*_{\mathbf{r}}} \langle \rho(\mathsf{E}), u\rangle u$, thus

$$\begin{aligned}
\varphi([\![\rho(\mathsf{E})]\!]) &= \varphi\Big(\sum_{u\in A^*_{\mathbf{r}}} \langle \rho(\mathsf{E}), u\rangle u\Big) = \sum_{u\in A^*_{\mathbf{r}}} \langle \rho(\mathsf{E}), u\rangle \varphi(u) \\
&= \sum_{w\in A^*} \sum_{u\in (w\mathbf{r})^* w} \langle \rho(\mathsf{E}), u\rangle w = \sum_{w\in A^*} \sum_{i=0}^{\infty} \langle \rho(\mathsf{E}), (w\mathbf{r})^i w\rangle w \qquad (23)\\
&= \sum_{w\in A^*} \langle \mathsf{E}, w\rangle w = [\![\mathsf{E}]\!] \,.
\end{aligned}$$

□

To complete the description of the function ρ, we characterise now the rational expressions which are in its image; on this image, ρ can be inverted.

Definition 6. *The set* $\mathbb{K}\mathsf{PreHadExp}A$ *of* $\mathbb{K}$*-pre-Hadamard expressions over* A *is generated by the following grammar:*

$$\mathsf{P} \to \mathsf{E} \in \mathbb{K}\mathsf{RatExp}A \mid k\mathsf{P}, k \in \mathbb{K} \mid \mathsf{P}k, k \in \mathbb{K} \mid \mathsf{P} + \mathsf{P} \mid \mathsf{P}\boldsymbol{r}\mathsf{P} \mid (\mathsf{P}\boldsymbol{r})^*\mathsf{P}. \qquad (24)$$

Proposition 5. *The image of* ρ *is* $\mathbb{K}\mathsf{PreHadExp}A$*.*

Proof. The proof is straightforward by induction: first, the image of every expression in $\mathbb{K}\mathsf{HadExp}A$ is in $\mathbb{K}\mathsf{PreHadExp}A$; conversely, every production of Grammar (24) corresponds to a right-hand side in the definition of ρ given in (16). □

5. From Hadamard Expressions to Weighted Rotating Automata

5.1. A Generic Extension of Automata Synthesis

An algorithm which turns a rational expression into a (one-way) automaton realises a mapping σ from rational expressions to classical automata which is consistent with the interpretation of expressions and the behaviour of automata.

Figure 3: A correct conversion algorithm makes this diagram commutative.

The superposition of Figures 1, 2 and 3 leads to Figure 4.

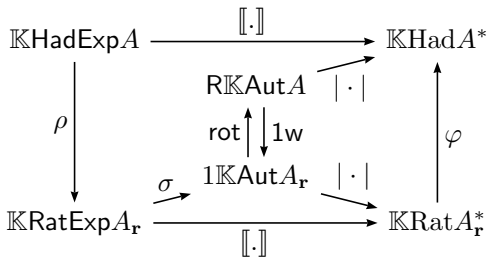


Figure 4: The transformation of Hadamard expressions into rotating automata.

This illustrates how the combination of Propositions 3 and 4 proves following theorem.

Theorem 1. *If σ is an algorithm that converts a rational expression into an equivalent one-way automaton, then $\mathsf{rot} \circ \sigma \circ \rho$ converts a Hadamard expression to an equivalent rotating automaton.*

The complexity of the transformation of a Hadamard expression into a rational expression by ρ is linear, and the complexity of rot is constant, since it is only a different interpretation of the same object. Therefore, since the complexity of the conversion σ of a rational expression to an automaton is at least linear, the complexity of $\mathsf{rot} \circ \sigma \circ \rho$ is equal to the complexity of σ.

When the number of letters in the rational expression is a parameter of the complexity of the conversion, this parameter must also count the number of Hadamard operators to get the complexity of the extension of the conversion from Hadamard expressions to rotating automata.

In the next subsections, we apply Theorem 1 to derivation, follow and Thompson-like automata. Notice that Theorem 1 allows to extend any algorithm that converts rational expressions to automata.

5.2. Derivation

The derivation of weighted rational expressions has been defined in [10]. It requires an auxiliary function Null.

Definition 7. *The function* Null *from* $\mathbb{K}\mathsf{RatExp}A$ *into* $\mathbb{K}$ *is inductively defined by*

$$\begin{gathered}\forall a \in A,\ \mathsf{Null}(a) = \mathsf{Null}(0) = 0_{\mathbb{K}}, \qquad \mathsf{Null}(1) = 1_{\mathbb{K}},\\ \mathsf{Null}(\mathsf{F}+\mathsf{G}) = \mathsf{Null}(\mathsf{F}) + \mathsf{Null}(\mathsf{G}), \qquad \mathsf{Null}(\mathsf{F}.\mathsf{G}) = \mathsf{Null}(\mathsf{F})\,\mathsf{Null}(\mathsf{G}),\\ \mathsf{Null}(k\mathsf{F}) = k\,\mathsf{Null}(\mathsf{F}), \qquad \mathsf{Null}(\mathsf{F}k) = \mathsf{Null}(\mathsf{F})\,k, \qquad \mathsf{Null}(\mathsf{F}^*) = \mathsf{Null}(\mathsf{F})^*.\end{gathered} \tag{25}$$

It is straightforward that, for every rational expression E, $\mathsf{Null}(\mathsf{E})$ is the weight of the empty word in $[\![\mathsf{E}]\!]$.

The derivative of an expression E is a linear combination of expressions, called a *polynomial* of expressions. The formal sum in polynomials of expressions (like in polynomials of *positions* in the next part) is denoted $\boxplus$ to avoid any confusion with the sum in expressions or with Hadamard operators. We use square brackets around polynomials to point out the distributivity with the operator that follows the brackets; for instance, if $\mathsf{P} = \boxplus \alpha_i \mathsf{E}_i$ is a polynomial, $[\mathsf{P}].\mathsf{F}$ is the polynomial $\boxplus \alpha_i(\mathsf{E}_i.\mathsf{F})$.

Definition 8. *The derivative of an expression* E *in* $\mathbb{K}\mathsf{RatExp}A$ *by a letter is a polynomial of expressions inductively defined for all* a *in* A *as:*

$$\begin{gathered}\frac{\partial}{\partial a}0 = \frac{\partial}{\partial a}1 = 0, \qquad \forall b \in A,\ \frac{\partial}{\partial a}b = \begin{cases}1 \text{ if } a = b,\\ 0 \text{ otherwise,}\end{cases}\\ \forall k \in \mathbb{K},\ \frac{\partial}{\partial a}k\mathsf{E} = k\,\frac{\partial}{\partial a}\mathsf{E},\ \frac{\partial}{\partial a}\mathsf{E}k = \left[\frac{\partial}{\partial a}\mathsf{E}\right]k,\ \frac{\partial}{\partial a}(\mathsf{E}+\mathsf{F}) = \frac{\partial}{\partial a}\mathsf{E} \boxplus \frac{\partial}{\partial a}\mathsf{F},\\ \frac{\partial}{\partial a}(\mathsf{E}\mathsf{F}) = \left[\frac{\partial}{\partial a}\mathsf{E}\right].\mathsf{F} \boxplus \mathsf{Null}(\mathsf{E})\frac{\partial}{\partial a}\mathsf{F},\ \frac{\partial}{\partial a}(\mathsf{E}^*) = \mathsf{Null}(\mathsf{E}^*)\left[\frac{\partial}{\partial a}\mathsf{E}\right].\mathsf{E}^*.\end{gathered} \tag{26}$$

There is only a finite number of expressions that arise in the iterated derivation of an expression E. Therefore, a weighted (one-way) automaton can be built, where each state is an expression, the initial state is E itself, there is a transition from F to G with label a and weight k if $\langle \frac{\partial}{\partial a}\mathsf{F}, \mathsf{G}\rangle = k$, and the final weight of a state F is $\langle \mathsf{F}, \varepsilon\rangle$. By [10], the behaviour of this weighted automaton is $[\![E]\!]$.

We apply now Theorem 1 to derivation.

Example. Let $\mathsf{E}_0 = ((\frac{1}{2}(a+b))^*b(a+b)^*) \circledast (ab)^*$. We set $\mathsf{F}_1 = (\frac{1}{2}(a+b))^*b(a+b)^*$, then $\mathsf{E}_1 = \rho(\mathsf{E}_0) = (\mathsf{F}_1\mathbf{r})^*(ab)^*$. The derivatives of E_1 are shown on Table 1. This leads to the derivation automaton of E_1 shown on Figure 5. This automaton can be interpreted as the rotating derivation automaton of E_0.

From the application of derivations to $\rho(\mathsf{E})$, where E is a Hadamard expression, we can define an extension of the derivation rules to directly derive Hadamard expressions. We first need to extend the Null function.

Expression	$\frac{\partial}{\partial a}$	$\frac{\partial}{\partial b}$	$\frac{\partial}{\partial \mathbf{r}}$
$\mathsf{E}_1 = (\mathsf{F}_1\mathbf{r})^*(ab)^*$	$\frac{1}{2}\mathsf{E}_2 \boxplus \mathsf{E}_4$	$\frac{1}{2}\mathsf{E}_2 \boxplus \mathsf{E}_3$	0
$\mathsf{E}_2 = \mathsf{F}_1\mathbf{r}\mathsf{E}_1$	$\frac{1}{2}\mathsf{E}_2$	$\frac{1}{2}\mathsf{E}_2 \boxplus \mathsf{E}_3$	0
$\mathsf{E}_3 = (a+b)^*\mathbf{r}\mathsf{E}_1$	E_3	E_3	E_1
$\mathsf{E}_4 = b(ab)^*$	0	E_5	0
$\mathsf{E}_5 = (ab)^*$	E_4	0	0

(27)

Table 1: The derivatives of E_1.

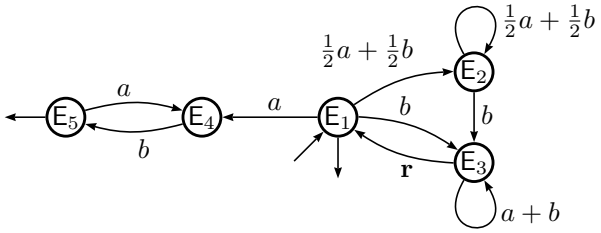


Figure 5: The one-way derivation automaton of $\mathsf{E}_1 = \rho(\mathsf{E}_0)$.

Definition 9. *For every Hadamard expression* E*, if* E *is a rational expression,* $\mathsf{Null}(\mathsf{E})$ *follows Definition 7, otherwise, it is inductively defined as:*

$$\begin{gathered}\forall k \in \mathbb{K},\ \mathsf{Null}(k\mathsf{E}) = k\,\mathsf{Null}(\mathsf{E}), \quad \mathsf{Null}(\mathsf{E}k) = \mathsf{Null}(\mathsf{E})\,k, \quad \mathsf{Null}(\mathsf{E} \odot \mathsf{F}) = 0,\\ \mathsf{Null}(\mathsf{E} + \mathsf{F}) = \mathsf{Null}(\mathsf{E}) + \mathsf{Null}(\mathsf{F}), \qquad \mathsf{Null}(\mathsf{E} \circledast \mathsf{F}) = \mathsf{Null}(\mathsf{F}).\end{gathered} \tag{28}$$

Remark. Note that $\mathsf{Null}(\mathsf{E}) = \langle \mathsf{E}, \varepsilon \rangle$ does not hold anymore. This discrepancy comes from the fact that $\mathsf{Null}(\mathsf{E})$ is forged to be equal to $\langle \rho(\mathsf{E}), \varepsilon \rangle$, which is different from $\langle \mathsf{E}, \varepsilon \rangle$.

Definition 10. *The derivation over Hadamard expressions is extended as follows; for every letter* a*, if* E *is a rational expression, the derivation follows Definition 8, otherwise, the derivation is inductively defined as:*

$$\begin{gathered}\forall k \in \mathbb{K},\ \frac{\partial}{\partial a} k\mathsf{E} = k\,\frac{\partial}{\partial a}\mathsf{E},\ \ \frac{\partial}{\partial a}\mathsf{E}k = \left[\frac{\partial}{\partial a}\mathsf{E}\right] k,\ \ \frac{\partial}{\partial a}(\mathsf{E} + \mathsf{F}) = \frac{\partial}{\partial a}\mathsf{E} \boxplus \frac{\partial}{\partial a}\mathsf{F},\\ \frac{\partial}{\partial a}(\mathsf{E} \odot \mathsf{F}) = \left[\frac{\partial}{\partial a}\mathsf{E}\right] \odot \mathsf{F}, \qquad \frac{\partial}{\partial a}(\mathsf{E} \circledast \mathsf{F}) = \left[\frac{\partial}{\partial a}\mathsf{E}\right] \odot (\mathsf{E} \circledast \mathsf{F}) \boxplus \frac{\partial}{\partial a}\mathsf{F}.\end{gathered} \tag{29}$$

A derivation with respect to the Hadamard product is also defined:

$$\begin{gathered}\frac{\partial}{\partial \odot}\mathsf{E} = 0 \text{ if } \mathsf{E} \in \mathbb{K}\mathsf{RatExp}, \qquad \frac{\partial}{\partial \odot}(k\mathsf{E}) = k\frac{\partial}{\partial \odot}\mathsf{E},\ \ \frac{\partial}{\partial \odot}(\mathsf{E}k) = \left[\frac{\partial}{\partial \odot}\mathsf{E}\right] k,\\ \frac{\partial}{\partial \odot}(\mathsf{E} + \mathsf{F}) = \frac{\partial}{\partial \odot}\mathsf{E} \boxplus \frac{\partial}{\partial \odot}\mathsf{F}, \qquad \frac{\partial}{\partial \odot}(\mathsf{E} \odot \mathsf{F}) = \mathsf{Null}(\mathsf{E})\,\mathsf{F} \boxplus \left[\frac{\partial}{\partial \odot}\mathsf{E}\right] \odot \mathsf{F},\\ \frac{\partial}{\partial \odot}(\mathsf{E} \circledast \mathsf{F}) = \left[\frac{\partial}{\partial \odot}\mathsf{E}\right] \odot (\mathsf{E} \circledast \mathsf{F}) \boxplus \mathsf{Null}(\mathsf{E})\,(\mathsf{E} \circledast \mathsf{F}) \boxplus \frac{\partial}{\partial \odot}\mathsf{F}.\end{gathered} \tag{30}$$

The correctness of this definition comes from Theorem 1 and the following proposition.

Proposition 6. *For every Hadamard expression* E*,*

$$\begin{gathered}\mathsf{Null}(\mathsf{E}) = \mathsf{Null}(\rho(\mathsf{E})),\\ \forall a \in A,\ \frac{\partial}{\partial a}\mathsf{E} = \rho^{-1}\left(\frac{\partial}{\partial a}\rho(\mathsf{E})\right), \qquad \textit{and } \frac{\partial}{\partial \odot}\mathsf{E} = \rho^{-1}\left(\frac{\partial}{\partial \boldsymbol{r}}\rho(\mathsf{E})\right),\end{gathered} \tag{31}$$

where ρ^{-1} *is extended to polynomials by linearity.*

Proof. The proof is by induction on E. If E is in $\mathbb{K}\mathsf{RatExp}$, ρ and ρ^{-1} are the identity, and $\frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{E}) = 0$, hence the result holds. Assume that the result is true for F and G. It holds by linearity for $\mathsf{E} = k\mathsf{F}$, $\mathsf{E} = \mathsf{F}k$ and $\mathsf{E} = \mathsf{F} + G$.

a) If $\mathsf{E} = \mathsf{F} \odot \mathsf{G}$, $\mathsf{Null}(\rho(\mathsf{E})) = \mathsf{Null}(\rho(F))\mathsf{Null}(\mathbf{r})\mathsf{Null}(\mathsf{G}) = 0 = \mathsf{Null}(\mathsf{E})$, and:

$$\begin{aligned}\frac{\partial}{\partial a}(\mathsf{F} \odot \mathsf{G}) &= \left[\frac{\partial}{\partial a}\mathsf{F}\right] \odot \mathsf{G} = \left[\rho^{-1}\big(\frac{\partial}{\partial a}\rho(\mathsf{F})\big)\right] \odot \mathsf{G}\\ &= \rho^{-1}\left(\left[\frac{\partial}{\partial a}\rho(\mathsf{F})\right]\mathbf{r}\rho(\mathsf{G})\right) = \rho^{-1}\left(\frac{\partial}{\partial a}\big(\rho(\mathsf{F})\mathbf{r}\rho(\mathsf{G})\big)\right)\\ &= \rho^{-1}\left(\frac{\partial}{\partial a}\rho(\mathsf{E})\right)\end{aligned} \tag{32}$$

$$\begin{aligned}\frac{\partial}{\partial \odot}(\mathsf{F} \odot \mathsf{G}) &= \mathsf{Null}(\mathsf{F})\,\mathsf{G} \boxplus \left[\frac{\partial}{\partial \odot}\mathsf{F}\right] \odot \mathsf{G}\\ &= \mathsf{Null}(\rho(\mathsf{F}))\,\mathsf{G} \boxplus \left[\rho^{-1}\big(\frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{F})\big)\right] \odot \mathsf{G}\\ &= \rho^{-1}\left(\mathsf{Null}(\rho(\mathsf{F}))\rho(\mathsf{G}) \boxplus \left[\frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{F})\right]\mathbf{r}\rho(\mathsf{G})\right)\\ &= \rho^{-1}\left(\left[\Big[\frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{F})\Big]\mathbf{r} \boxplus \mathsf{Null}(\rho(\mathsf{F}))\frac{\partial}{\partial \mathbf{r}}\mathbf{r}\right]\rho(\mathsf{G})\right)\\ &= \rho^{-1}\left(\left[\frac{\partial}{\partial \mathbf{r}}\big(\rho(\mathsf{F})\mathbf{r}\big)\right]\rho(\mathsf{G})\right) = \rho^{-1}\left(\frac{\partial}{\partial \mathbf{r}}\big(\rho(\mathsf{F})\mathbf{r}\rho(\mathsf{G})\big)\right)\\ &= \rho^{-1}\left(\frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{E})\right).\end{aligned} \tag{33}$$

b) If $\mathsf{E} = \mathsf{F} \circledast \mathsf{G}$,
$\mathsf{Null}(\rho(\mathsf{E})) = (\mathsf{Null}(\rho(F))\mathsf{Null}(\mathbf{r}))^*\mathsf{Null}(G) = \mathsf{Null}(\mathsf{G}) = \mathsf{Null}(\mathsf{E})$, and:

$$
\begin{aligned}
\frac{\partial}{\partial a}(\mathsf{F} \circledast \mathsf{G}) &= \left[\frac{\partial}{\partial a}\mathsf{F}\right] \odot (\mathsf{F} \circledast \mathsf{G}) \boxplus \frac{\partial}{\partial a}\mathsf{G} \\
&= \rho^{-1}\left(\left[\frac{\partial}{\partial a}\rho(\mathsf{F})\right]\mathbf{r}(\rho(\mathsf{F})\mathbf{r})^*\rho(\mathsf{G})\right) \boxplus \frac{\partial}{\partial a}\mathsf{G} \\
&= \rho^{-1}\left((\mathsf{Null}(\rho(\mathsf{F})\mathbf{r}))^*\left[\frac{\partial}{\partial a}\big(\rho(\mathsf{F})\mathbf{r}\big)\right](\rho(\mathsf{F})\mathbf{r})^*\rho(\mathsf{G})\right) \boxplus \frac{\partial}{\partial a}\mathsf{G} \\
&= \rho^{-1}\left(\left[\frac{\partial}{\partial a}(\rho(\mathsf{F})\mathbf{r})^*\right]\rho(\mathsf{G}) \boxplus \frac{\partial}{\partial a}\rho(\mathsf{G})\right) \\
&= \rho^{-1}\left(\left[\frac{\partial}{\partial a}(\rho(\mathsf{F})\mathbf{r})^*\right]\rho(\mathsf{G}) \boxplus \mathsf{Null}((\rho(\mathsf{F})\mathbf{r})^*)\frac{\partial}{\partial a}\rho(\mathsf{G})\right) \\
&= \rho^{-1}\left(\frac{\partial}{\partial a}\big((\rho(\mathsf{F})\mathbf{r})^*\rho(\mathsf{G})\big)\right) \\
&= \rho^{-1}\left(\frac{\partial}{\partial a}\rho(\mathsf{E})\right).
\end{aligned}
\tag{34}
$$

$$
\begin{aligned}
\frac{\partial}{\partial \odot}(\mathsf{F} \circledast \mathsf{G}) &= \left[\frac{\partial}{\partial \odot}\mathsf{F}\right] \odot \mathsf{E} \boxplus \mathsf{Null}(\mathsf{F})\,\mathsf{E} \boxplus \frac{\partial}{\partial \odot}(\mathsf{G}) \\
&= \rho^{-1}\left(\left[\frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{F})\right]\mathbf{r}\rho(\mathsf{E})\right) \boxplus \rho^{-1}\left(\mathsf{Null}(\rho(\mathsf{F}))\frac{\partial}{\partial \mathbf{r}}\mathbf{r}\rho(\mathsf{E})\right) \boxplus \frac{\partial}{\partial \odot}(\mathsf{G}) \\
&= \rho^{-1}\left(\left[\left[\frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{F})\right]\mathbf{r} \boxplus \mathsf{Null}(\rho(\mathsf{F}))\frac{\partial}{\partial \mathbf{r}}\mathbf{r}\right]\rho(\mathsf{E})\right) \boxplus \frac{\partial}{\partial \odot}(\mathsf{G}) \\
&= \rho^{-1}\left(\left[\frac{\partial}{\partial \mathbf{r}}\big(\rho(\mathsf{F})\mathbf{r}\big)\right]\rho(\mathsf{E})\right) \boxplus \frac{\partial}{\partial \odot}(\mathsf{G}) \\
&= \rho^{-1}\left(\big(\mathsf{Null}(\rho(\mathsf{F})\mathbf{r})\big)^*\left[\frac{\partial}{\partial \mathbf{r}}\big(\rho(\mathsf{F})\mathbf{r}\big)\right](\rho(\mathsf{F})\mathbf{r})^*\rho(\mathsf{G}) \boxplus \frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{G})\right) \\
&= \rho^{-1}\left(\left[\frac{\partial}{\partial \mathbf{r}}\big(\rho(\mathsf{F})\mathbf{r}\big)^*\right]\rho(\mathsf{G}) \boxplus \mathsf{Null}\big((\rho(\mathsf{F})\mathbf{r})^*\big)\frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{G})\right) \\
&= \rho^{-1}\left(\frac{\partial}{\partial \mathbf{r}}\big((\rho(\mathsf{F})\mathbf{r})^*\rho(\mathsf{G})\big)\right) \\
&= \rho^{-1}\left(\frac{\partial}{\partial \mathbf{r}}\rho(\mathsf{E})\right).
\end{aligned}
\tag{35}
$$

□

5.3. Follow Automata

The definition of the Follow automaton for rational expressions is described in [6]. In every rational expression E, we consider the list of occurrences of letters; each of these occurrences is called a *position*, and we denote $\mathsf{pos}(\mathsf{E})$ the set of positions of the expression E. We consider (formal) linear combinations

of positions, and we denote the set of linear combinations of positions of E with $\mathbb{K}\langle\mathsf{pos}(\mathsf{E})\rangle$.[1]

The Follow automaton requires the definition of four functions: $\mathsf{Null}(\mathsf{E})$ in $\mathbb{K}$ is already defined in Proposition 7; $\mathsf{First}(\mathsf{E})$, $\mathsf{Last}(\mathsf{E})$, and $\mathsf{Follow}(\mathsf{E}, p)$ (where p is a position) are inductively defined by:

$$\begin{gathered}\mathsf{First}(0) = \mathsf{First}(1) = 0_{\mathbb{K}}, \qquad \mathsf{First}(a) = \mathsf{position\ of}(a),\\ \mathsf{First}(\mathsf{F}+\mathsf{G}) = \mathsf{First}(\mathsf{F}) \boxplus \mathsf{First}(\mathsf{G}), \quad \mathsf{First}(\mathsf{FG}) = \mathsf{First}(\mathsf{F}) \boxplus \mathsf{Null}(\mathsf{F})\mathsf{First}(\mathsf{G}),\\ \mathsf{First}(k\mathsf{F}) = k\,\mathsf{First}(\mathsf{F}),\ \mathsf{First}(\mathsf{F}k) = \mathsf{First}(\mathsf{F}), \quad \mathsf{First}(\mathsf{F}^*) = (\mathsf{Null}(\mathsf{F}))^*\mathsf{First}(\mathsf{F})\end{gathered} \tag{36}$$

$$\begin{gathered}\mathsf{Last}(0) = \mathsf{Last}(1) = 0_{\mathbb{K}}, \qquad \mathsf{Last}(a) = \mathsf{position\ of}(a),\\ \mathsf{Last}(\mathsf{F}+\mathsf{G}) = \mathsf{Last}(\mathsf{F}) \boxplus \mathsf{Last}(\mathsf{G}), \quad \mathsf{Last}(\mathsf{FG}) = \mathsf{Last}(\mathsf{G}) \boxplus \mathsf{Last}(\mathsf{F})\mathsf{Null}(\mathsf{G}),\\ \mathsf{Last}(k\mathsf{F}) = \mathsf{Last}(\mathsf{F}),\ \mathsf{Last}(\mathsf{F}k) = \mathsf{Last}(\mathsf{F})\,k, \quad \mathsf{Last}(\mathsf{F}^*) = \mathsf{Last}(\mathsf{F})(\mathsf{Null}(\mathsf{F}))^*\end{gathered} \tag{37}$$

$$\begin{gathered}\mathsf{Follow}(0,p) = \mathsf{Follow}(1,p) = \mathsf{Follow}(a,p) = 0_{\mathbb{K}},\\ \mathsf{Follow}(k\mathsf{F},p) = \mathsf{Follow}(\mathsf{F}k,p) = \mathsf{Follow}(\mathsf{F},p),\\ \mathsf{Follow}(\mathsf{F}+\mathsf{G},p) = \mathsf{Follow}(\mathsf{F},p) \boxplus \mathsf{Follow}(\mathsf{G},p),\\ \mathsf{Follow}(\mathsf{FG},p) = \mathsf{Follow}(\mathsf{F},p) \boxplus \mathsf{Follow}(\mathsf{G},p) \boxplus \langle\mathsf{Last}(\mathsf{F}),p\rangle\mathsf{First}(\mathsf{G}),\\ \mathsf{Follow}(\mathsf{F}^*,p) = \mathsf{Follow}(\mathsf{F},p) \boxplus \langle\mathsf{Last}(\mathsf{F}),p\rangle(\mathsf{Null}(\mathsf{F}))^*\mathsf{First}(\mathsf{F}),\end{gathered} \tag{38}$$

where a is a letter and k is in $\mathbb{K}$. It is convenient to extend $\mathsf{pos}(\mathsf{E})$ with an *initial position* i_0 and to extend Follow by $\mathsf{Follow}(\mathsf{E}, i_0) = \mathsf{First}(\mathsf{E})$; moreover, by convention, $\langle\mathsf{Last}(\mathsf{E}), i_0\rangle = \mathsf{Null}(\mathsf{E})$.

The weighted *Position automaton* [3] can then be defined, where the set of states is the set of positions, the initial state is the initial position, there is a transition from p to q with label a and weight k if there is a letter a in position q and $\langle\mathsf{Follow}(\mathsf{E}, p), q\rangle = k$; the final weight of state p is $\langle\mathsf{Last}(\mathsf{E}), p\rangle$.

The *Follow automaton* is a *quotient* of the Position automaton: if Follow coincides on two positions p and q, and $\langle\mathsf{Last}(\mathsf{E}), p\rangle = \langle\mathsf{Last}(\mathsf{E}), q\rangle$, then the corresponding states can be merged.

Example. We consider the expression E_1 defined in Example 5.2, a rational expression with 8 positions. For convenience, we add indices to identify positions: $((\frac{1}{2}(a_1 + b_2))^* b_3 (a_4 + b_5)^* \mathbf{r}_6)^* (a_7 b_8)^*$. The Follow and Last functions shown on Table 2 induces an equivalence on positions: $\{\{i_0, 6\}, \{1, 2\}, \{3, 4, 5\}, \{7\}, \{8\}\}$; the Follow automaton of E_1 is drawn on Figure 6; seen as a rotating automaton, this automaton realises $[\![\mathsf{E}_0]\!]$.

Like for derivatives, the functions First, Last and Follow can be extended to Hadamard expressions in order to get a direct construction. Notice that, for every Hadamard expression E, an occurrence of letter $\mathbf{r}$ appears in $\rho(\mathsf{E})$ for each Hadamard operator which appears in E. This leads to extend the positions to

[1] Notice that $\mathbb{K}\langle\mathsf{pos}(\mathsf{E})\rangle$ is not a semiring, since $\mathsf{pos}(\mathsf{E})$ is not a monoid; nevertheless, we use the same notations as series for denoting the coefficient of such a linear combination.

position	$\mathsf{Follow}(\mathsf{E}_1, \mathrm{pos})$	$\langle \mathsf{Last}(\mathsf{E}_1), \mathrm{pos} \rangle$
i_0	$\frac{1}{2}\,1 \boxplus \frac{1}{2}\,2 \boxplus 3 \boxplus 7$	1
1 (a)	$\frac{1}{2}\,1 \boxplus \frac{1}{2}2 \boxplus 3$	0
2 (b)	$\frac{1}{2}\,1 \boxplus \frac{1}{2}2 \boxplus 3$	0
3 (b)	$4 \boxplus 5 \boxplus 6$	0
4 (a)	$4 \boxplus 5 \boxplus 6$	0
5 (b)	$4 \boxplus 5 \boxplus 6$	0
6 $(\mathbf{r})$	$\frac{1}{2}\,1 \boxplus \frac{1}{2}\,2 \boxplus 3 \boxplus 7$	1
7 (a)	8	0
8 (b)	7	1

(39)

Table 2: The Follow and Last functions for expression E_1.

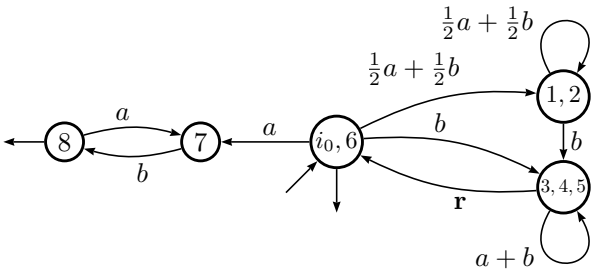


Figure 6: The Follow automaton of $\mathsf{E}_1 = \rho(\mathsf{E}_0)$.

the occurrences of Hadamard operators. The extension of function Null is done in Definition 9.

Definition 11. *For every Hadamard expression* E*, the inductive definition of* First*,* Last *and* Follow *given in Equations* (36)*,* (37) *and* (38) *is extended to Hadamard operators by*

$$\begin{aligned} \mathsf{First}(\mathsf{F} \odot_i \mathsf{G}) &= \mathsf{First}(\mathsf{F}) \boxplus \mathsf{Null}(\mathsf{F})\, i, \\ \mathsf{First}(\mathsf{F} \circledast_i \mathsf{G}) &= \mathsf{First}(\mathsf{F}) \boxplus \mathsf{Null}(\mathsf{F})\, i \boxplus \mathsf{First}(\mathsf{G}), \end{aligned} \tag{40}$$

$$\begin{aligned} \mathsf{Last}(\mathsf{F} \odot_i \mathsf{G}) &= \mathsf{Last}(\mathsf{G}) \boxplus \mathsf{Null}(\mathsf{G})\, i, \\ \mathsf{Last}(\mathsf{F} \circledast_i \mathsf{G}) &= \mathsf{Last}(\mathsf{G}) \boxplus \mathsf{Null}(\mathsf{G})\, i, \end{aligned} \tag{41}$$

$$\begin{aligned} \mathsf{Follow}(\mathsf{F} \odot_i \mathsf{G}, p) &= \mathsf{Follow}(\mathsf{F}, p) \boxplus \mathsf{Follow}(\mathsf{G}, p) \\ &\qquad \boxplus \langle \mathsf{Last}(\mathsf{F}), p \rangle\, i \boxplus \langle i, p \rangle \mathsf{First}(\mathsf{G}), \\ \mathsf{Follow}(\mathsf{F} \circledast_i \mathsf{G}, p) &= \mathsf{Follow}(\mathsf{F}, p) \boxplus \mathsf{Follow}(\mathsf{G}, p) \boxplus \langle \mathsf{Last}(\mathsf{F}), p \rangle\, i \\ &\qquad \boxplus \langle i, p \rangle \mathsf{First}(\mathsf{F} \circledast_i \mathsf{G}), \end{aligned} \tag{42}$$

where $\langle i, p \rangle$ *is equal to* 1 *if* $i = p$*, and to* 0 *otherwise.*

Since a position is assigned to each Hadamard operator of a Hadamard expression E, and each Hadamard operator has a corresponding occurrence of $\mathbf{r}$

in $\rho(\mathsf{E})$, there is a natural bijection between positions of E and positions of $\rho(\mathsf{E})$. The soundness of Definition 11 comes hence from the following proposition.

Proposition 7. *For every Hadamard expression* E*,*

$$\begin{gathered}\mathsf{First}(\mathsf{E}) = \mathsf{First}(\rho(\mathsf{E})), \quad \mathsf{Last}(\mathsf{E}) = \mathsf{Last}(\rho(\mathsf{E})), \\ \forall p \in \mathsf{pos}(\mathsf{E}),\ \mathsf{Follow}(\mathsf{E}, p) = \mathsf{Follow}(\rho(\mathsf{E}), p).\end{gathered} \tag{43}$$

5.4. Thompson-like Automata

Like for one-way automata, the model of rotating automata can be extended to support ε-transitions. We consider here a variant of the Thompson automaton which is more suitable for weighted automata since it prevents circuits of ε-transition. This variant is inspired by the ZPC-structure described in [4]. For every rational expression E, the automaton $\mathcal{T}(\mathsf{E})$ has a single initial state i (with weight 1) and a final state t (with weight 1), distinct from state i; on top of state t, the initial state i may also be final. There is no path with label ε from i to t; hence, the weight of ε in the behaviour of E is given by the final weight c of i. Like in the Thompson automaton,

- there is no incoming transition on state i;
- there is no outgoing transition from state t;
- if two distinct transitions come from (*resp.* go to) the same state, they are ε transitions.

The inductive construction of $\mathcal{T}(\mathsf{E})$ is described on Figure 7. Using Theorem 1, this construction can be extended to Hadamard expressions. Automata of Figure 8 are obtained from the direct application of Theorem 1 by contracting non branching paths of ε-transitions. This construction proves that, like in the case of rational expressions, for every Hadamard expression, an equivalent rotating automaton with ε-transitions can be built, with a linear number of states and transitions.

6. From Weighted Rotating Automata to Hadamard Expressions

We apply the method used in the previous section to get an algorithm which converts rotating automata to Hadamard expressions. An algorithm that turns a one-way automaton into a rational expression is a function τ from $1\mathbb{K}\mathsf{Aut}A$ to $\mathbb{K}\mathsf{RatExp}A$ which is consistent with the interpretation of expressions and the behaviour of automata. Hence, if an inverse of ρ existed, a commutative diagram similar to Figure 4 could be drawn, and a result similar to Theorem 1 would be proved.

Unfortunately, the inverse of ρ is only defined on a strict subset of rational expressions, the pre-Hadamard expressions of Definition 6. To apply our method, we must therefore ensure that the algorithm τ outputs pre-Hadamard expressions.

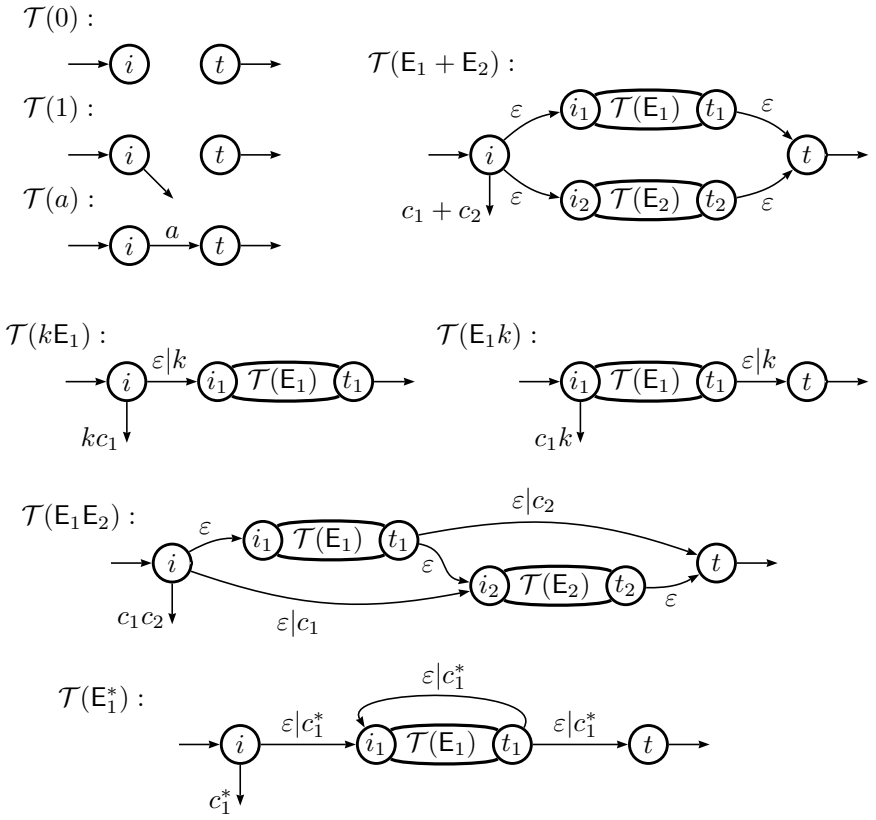


Figure 7: A Thompson-like automaton for rational series

Theorem 2. *If τ is an algorithm that converts an automaton in $1\mathbb{K}\mathsf{Aut}A_{\boldsymbol{r}}$ to an equivalent pre-Hadamard expression, then $\rho^{-1} \circ \tau \circ 1\mathsf{w}$ converts an automaton in $\mathsf{R}\mathbb{K}\mathsf{Aut}A$ to an equivalent Hadamard expression.*

6.1. State Elimination on an $\boldsymbol{r}$-local Automaton

We show in this part that the State Elimination method introduced in [1] can be applied on some particular automata over $A_{\mathbf{r}}$ in such a way that it outputs pre-Hadamard expressions. These automata are **r**-local automata.

Definition 12. *An automaton (Q, E, I, T) in $1\mathbb{K}\mathsf{Aut}A_{\boldsymbol{r}}$ is* **r**-local *if there is a partition $\{Q_1, Q_2\}$ of Q with no initial state in Q_2, and such that the label of a transition is $\boldsymbol{r}$ if and only if this transition ends in Q_2.*

We briefly recall the principle of the State Elimination method applied to an automaton $\mathcal{A} = (Q, E, I, T)$ in $\mathbb{K}\mathsf{Aut}A$.

First, $\mathcal{A}$ is converted into a directed graph with no multiple arcs, where arcs are labeled by rational expressions:

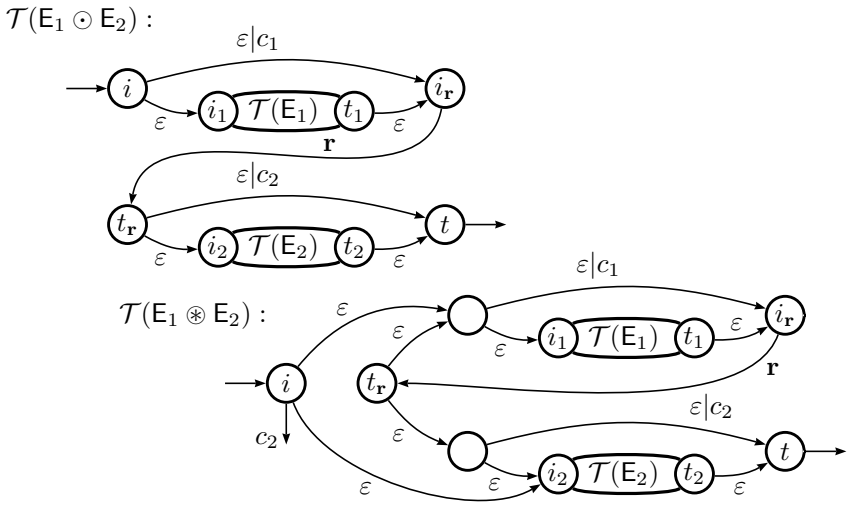
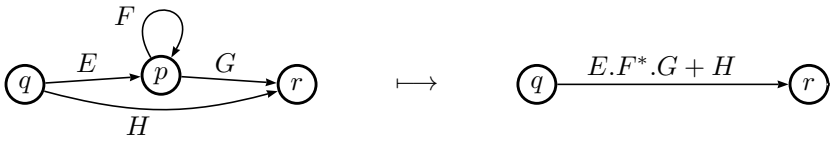


Figure 8: A Thompson-like automaton for Hadamard series

- every state is converted into a vertex, and two vertices, named i_0 and t_0 are added;
- for every transition (p, a, q) with weight $k \neq 0_{\mathbb{K}}$ in $\mathcal{A}$, there is an arc (p, q) with label ka; if there are several transitions between the same pair (p, q) of states, the label of the arc (p, q) is the sum of the corresponding expressions;
- if state p is initial with weight k, there is an arc (i_0, p) with label k;
- if state p is final with weight k, there is an arc (p, t_0) with label k.

Then, the elimination method runs as follows. At each step, a vertex p different from i_0 and t_0 is considered. For every predecessor q of p, for every successor r of p, the arc (q, r) is updated as follows:



Then, the vertex p is deleted. If the vertex p has no loop, there is no factor F^* in the resulting expression.

At the end, only vertices i_0 and t_0 remain, and the label of the arc (i_0, t_0) denotes the behaviour of $\mathcal{A}$.

Notice that the result heavily depends on the ordering on vertices during the elimination.

We present a variant of this algorithm for **r**-local automata in order to obtain a pre-Hadamard expression. First, the vertices corresponding to states

with incoming transitions with a label different from $\mathbf{r}$ are deleted before the other ones. Second, to get well-formed pre-Hadamard expressions, the following rewriting rules are applied:

$$\begin{aligned} \mathsf{E}\mathbf{r} + \mathsf{F}\mathbf{r} &\to (\mathsf{E}+\mathsf{F})\mathbf{r} \\ \mathsf{E}(k\mathbf{r}) &\to (\mathsf{E}k)\mathbf{r} \\ \mathsf{E}(\mathsf{F}\mathbf{r}) &\to (\mathsf{E}\mathsf{F})\mathbf{r}. \end{aligned} \tag{44}$$

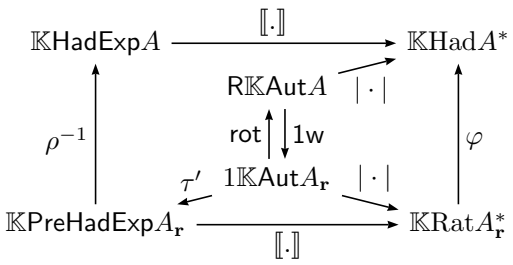


Figure 9: The transformation of rotating automata into Hadamard expressions.

Proposition 8. *The variant of the elimination method applied to an $\mathbf{r}$-local automaton $\mathcal{A}$ yields a pre-Hadamard expression $\mathsf{E}(\mathcal{A})$ such that $[\![\mathsf{E}(\mathcal{A})]\!] = |\mathcal{A}|$.*

Proof. Let $\mathcal{A} = (Q_1 \cup Q_2, E, I, T)$ be an $\mathbf{r}$-local automaton in $1\mathbb{K}\mathsf{Aut}A_{\mathbf{r}}$. Let R denote the set of vertices that remain at each step of the elimination. The first stage is the removing of vertices in Q_1; the following properties are invariant during this stage :

$$\begin{aligned} \forall p \in R,\, \forall q_1 \in R \setminus Q_2,\ p \xrightarrow{\mathsf{E}} q_1 &\implies \mathsf{E} \in \mathbb{K}\mathsf{RatExp}A, \\ \forall q_2 \in Q_2 \cap R,\, p \xrightarrow{\mathsf{E}} q_2 &\implies \mathsf{E} = \mathsf{F}\mathbf{r} \text{ with } \mathsf{F} \in \mathbb{K}\mathsf{RatExp}A. \end{aligned} \tag{45}$$

The second stage is the removing of vertices in Q_2, and the following properties are invariant:

$$\begin{aligned} \forall p \in R,\ \forall q_2 \in Q_2 \cap R,\ p \xrightarrow{\mathsf{E}} q_2 &\implies \mathsf{E} = \mathsf{F}\mathbf{r} \text{ with } \mathsf{F} \in \mathbb{K}\mathsf{PreHadExp}A, \\ p \xrightarrow{\mathsf{E}} t_0 &\implies \mathsf{E} \in \mathbb{K}\mathsf{PreHadExp}A. \end{aligned} \tag{46}$$

Finally, at the end, the label of the arc (i_0, t_0) is in $\mathbb{K}\mathsf{PreHadExp}A$. □

6.2. State Elimination Variant in $1\mathbb{K}\mathsf{Aut}A_r$.

If an automaton in $1\mathbb{K}\mathsf{Aut}A_{\mathbf{r}}$ is not $\mathbf{r}$-local, a preprocessing must be applied. It consists in splitting states which violate the $\mathbf{r}$-local property. The operation is a *covering*; this ensures that it preserves the behaviour.

Definition 13. *Let $\mathcal{A} = (Q, E, I, T)$ and $\mathcal{B} = (S, F, J, U)$ be two automata in $1\mathbb{K}\mathsf{Aut}A$. $\mathcal{A}$ is a* covering *of $\mathcal{B}$ if there exists a surjection ψ from Q to S such that:*

- $\forall p \in Q,\ T(p) = U(\psi(p))$;
- *ψ induces a bijection from initial states of $\mathcal{A}$ onto initial states of $\mathcal{B}$, such that for every initial state p of $\mathcal{A}$, $I(p) = J(\psi(p))$;*
- *for every transition (r, a, s) in $\mathcal{B}$, and every state p in $\psi^{-1}(r)$, there exists one and only one state q in $\psi^{-1}(s)$ such that (p, a, q) is a transition of $\mathcal{A}$; moreover $E(p, a, q) = F(r, a, s)$.*

This definition implies that there is a bisimulation between automaton $\mathcal{A}$ and $\mathcal{B}$: there is one and only one way to lift up every initial state of $\mathcal{B}$ in $\mathcal{A}$ consistently with ψ, and for every path of $\mathcal{B}$ lifted up in $\mathcal{A}$, and every transition extending this path, there is one and only one way to lift this transition in $\mathcal{A}$. Therefore, there is a canonical one-to-one mapping of computations of $\mathcal{B}$ into computations of $\mathcal{A}$ that preserves both the labels and the weights.

Lemma 2. *If $\mathcal{A}$ is a covering of $\mathcal{B}$, they have the same behaviour: $|\mathcal{A}| = |\mathcal{B}|$.*

Remark. There is a one-to-one mapping between computations of $\mathcal{A}$ and $\mathsf{rot}(\mathcal{A})$, hence Lemma 2 implies that coverings of rotating automata also preserve the behaviour.

Lemma 3. *Every automaton in $1\mathbb{K}\mathsf{Aut}A_{\boldsymbol{r}}$ admits an $\boldsymbol{r}$-local covering.*

Proof. Let $\mathcal{B} = (S, F, J, U)$ be in $1\mathbb{K}\mathsf{Aut}A_{\mathbf{r}}$. We define the automaton $\mathcal{A} = (Q_1 \cup Q_2, E, I, T)$ in $1\mathbb{K}\mathsf{Aut}A_{\mathbf{r}}$. Q_1 and Q_2 are two distinct copies of S; for every state p in S, the correponding state in Q_1 (*resp.* Q_2) is denoted p_1 (*resp.* p_2). For every p, q in S, for every i in $\{1, 2\}$,

$$\begin{gathered} I(p_1) = J(p), \qquad I(p_2) = 0, \qquad T(p_i) = U(p), \\ \forall a \in A,\ E(p_i, a, q_1) = F(p, a, q), \qquad E(p_i, a, q_2) = 0, \\ E(p_i, \mathbf{r}, q_1) = 0, \qquad E(p_i, \mathbf{r}, q_2) = F(p, \mathbf{r}, q). \end{gathered}$$

$\mathcal{A}$ is a covering of $\mathcal{B}$ where p_1 and p_2 are mapped onto p. It is straightforward that $\mathcal{A}$ is $\mathbf{r}$-local. □

Finally, we get an algorithm τ that fulfills the hypothesis of Theorem 2, from which an algorithm that converts rotating automata to Hadamard expressions is deduced.

Proposition 9. *Let $\mathcal{A}$ be in $\mathsf{R}\mathbb{K}\mathsf{Aut}A$ and let F be the pre-Hadamard expression resulting from the elimination method on a $\boldsymbol{r}$-local covering of $\mathsf{1w}(\mathcal{A})$. Then, $\rho^{-1}(\mathsf{F})$ is a Hadamard expression such that*

$$[\![\rho^{-1}(\mathsf{F})]\!] = |\mathcal{A}|. \tag{47}$$

Remark. It is possible to design an algorithm h that turns every expression in $\mathbb{K}\mathsf{RatExp}A_{\mathbf{r}}$ to a pre-Hadamard expression. Then for every algorithm τ which converts weighted one-way automata into rational expressions, the algorithm $\rho^{-1} \circ h \circ \tau \circ \mathsf{1w}$ converts weighted rotating automata to Hadamard expressions. Nevertheless, it seems that it is more efficient to modify the algorithm τ such that it directly outputs pre-Hadamard expressions.

Example. The application of the usual State Elimination method on the automaton $\mathcal{A}_2$ drawn on Figure 10 results in the following expressions,

$$\left(ba^*(\frac{1}{2}\mathbf{r}+a)\right)^*, \tag{48}$$

which is not the image by ρ of any Hadamard expression. The automaton $\mathcal{A}'_2$ on Figure 11 is a $\mathbf{r}$-local covering of $\mathcal{A}_2$. We apply the variant of the state elimination method; we first eliminate states q_1 and p_1 which have no incoming transition with label $\mathbf{r}$; this leads to the graph of Figure 12. For convenience, to shorten the expressions, we use the identity $\varepsilon + \mathsf{E}\mathsf{E}^* \equiv \mathsf{E}^*$. Finally, after eliminating state p_2, the following rational expression is obtained:

$$(ba^*a)^* + (ba^*a)^*ba^*\frac{1}{2}\mathbf{r}\left((ba^*a)^*ba^*\frac{1}{2}\mathbf{r}\right)^*(ba^*a)^*. \tag{49}$$

This expression is indeed a pre-Hadamard expression; it is the image by ρ of

$$(ba^*a)^* + (ba^*a)^*ba^*\frac{1}{2} \odot \left((ba^*a)^*ba^*\frac{1}{2}\right) \circledast (ba^*a)^*. \tag{50}$$

Notice that it can easily be proved that $\mathsf{F} + \mathsf{E} \odot \mathsf{E} \circledast \mathsf{F}$ and $\mathsf{E} \circledast \mathsf{F}$ are equivalent for every pair of Hadamard expressions, hence, the behaviour of $\mathcal{A}_2$ is described by the Hadamard expression $\left((ba^*a)^*ba^*\frac{1}{2}\right) \circledast (ba^*a)^*$.

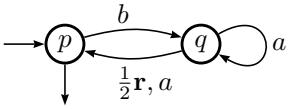


Figure 10: The rotating automaton $\mathcal{A}_2$.

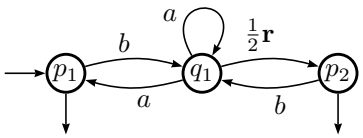


Figure 11: The $\mathbf{r}$-local rotating automaton $\mathcal{A}'_2$.

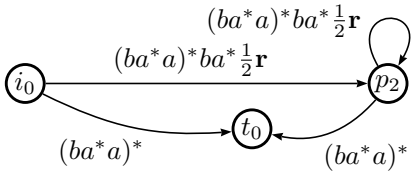


Figure 12: After the first stage in the variant of the state elimination applied to $\mathcal{A}'_2$.

7. Conclusion

The results stated in this paper are generic and can be applied to any semiring, commutative or not. The same methods can be applied in semirings which are not rationally additive semirings, but the fact that the interpretation of expressions and the behaviour of automata are defined, as well as the validity of the transformations, depend on specific properties of the expressions or automata.

[1] J. A. Brzozowski and E. J. McCluskey. Signal flow graph techniques for sequential circuit state diagrams. *IEEE Trans. Electron. Comput.*, EC-12(2):67–76, 1963.

[2] V. Carnino and S. Lombardy. On determinism and unambiguity of weighted two-way automata. *Int. J. Found. Comput. Sci.*, 26(8):1127–1146, 2015.

[3] P. Caron and M. Flouret. Glushkov construction for series: The non commutative case. *Int. J. Comput. Math.*, 80(4):457–472, 2003.

[4] J. Champarnaud, É. Laugerotte, F. Ouardi, and D. Ziadi. From regular weighted expressions to finite automata. In *Proc. of CIAA 2003*, volume 2759 of *LNCS*, pages 49–60, 2003.

[5] Z. Ésik and W. Kuich. Rationally additive semirings. *J. UCS*, 8(2):173–183, 2002.

[6] L. Ilie and S. Yu. Follow automata. *Inf. Comput.*, 186(1):140–162, 2003.

[7] C. Kapoutsis, R. Královič, and T. Mömke. Size complexity of rotating and sweeping automata. *J. Comput. System Sci.*, 78(2):537–558, 2012.

[8] S. C. Kleene. Representation of events in nerve nets and finite automata. *Automata Studies*, 1956.

[9] S. Lombardy. Two-way representations and weighted automata. *RAIRO: Theoret. Informatics Appl.*, 50(4):331–350, 2016.

[10] S. Lombardy and J. Sakarovitch. Derivatives of rational expressions with multiplicity. *Theoret. Comput. Sci.*, 332(13):141 – 177, 2005.

[11] G. Pighizzini. Two-way finite automata: Old and recent results. *Fundam. Inform.*, 126(2-3):225–246, 2013.

[12] W. J. Sakoda and M. Sipser. Nondeterminism and the size of two way finite automata. In *Proc. of STOC'78*, pages 275–286. ACM, 1978.

[13] M.-P. Schützenberger. On the definition of a family of automata. *Inform. Control*, 4:245–270, 1961.