

Finite transducers for divisibility monoids

Matthieu PICANTIN

LIAFA CNRS UMR 7089
Université Paris 7 Denis Diderot
F-75251 Paris France

Abstract

Divisibility monoids are a natural lattice-theoretical generalization of Mazurkiewicz trace monoids, namely monoids in which the distributivity of the involved divisibility lattices is kept as an hypothesis, but the relations between the generators are not supposed to necessarily be commutations. Here, we show that every divisibility monoid admits an explicit finite transducer which allows to compute normal forms in quadratic time. In addition, we prove that every divisibility monoid is biautomatic.

Key words: divisibility monoid, transducer, normal form, automatic monoid.

1 Introduction

The goal of this paper is to establish the following result :

Main Theorem. *Every left divisibility monoid admits an explicit finite transducer which allows to compute right normal forms in quadratic time.*

Mazurkiewicz's trace theory provides a well-investigated mathematical model for the sequential behavior of a parallel system in which the order of two independent actions is regarded as irrelevant. This is achieved by considering a free partially commutative monoid, namely, the free monoid of all words over a fixed alphabet modulo the congruence generated by equations of the form $ab = ba$ for pairs of independent actions (a, b) . Roughly speaking, a letter corresponds to an event and two letters commute when the corresponding events can occur simultaneously.

Email address: picantin@liafa.jussieu.fr (Matthieu PICANTIN).

URL: <http://www.liafa.jussieu.fr/~picantin> (Matthieu PICANTIN).

However, there are several areas in computer science where one would like to consider more general equations of the form $ab = cd$, rather than just $ab = ba$ as in trace theory. Left divisibility monoids have been introduced as a natural algebraic generalization of Mazurkiewicz’s trace monoids, namely monoids in which the distributivity of the underlying left divisibility lattices is kept as an hypothesis, but the relations between the generators are not supposed to necessarily be commutations.

The purpose of this paper is to study how to compute efficiently normal forms in left divisibility monoids. Following Thurston’s original idea about the automaticity of the braid groups (see [11]), we shall construct an explicit finite transducer—that is a finite automaton with output—allowing to compute normal forms in every left divisibility monoid. Since a standard transducer reads words *from the left to the right*, we need to define a *right* normal form, even if a *left* normal form for elements in a *left* divisibility monoid—like the normal form defined by Kuske [18] and generalizing the Cartier-Foata normal form known from the theory of Mazurkiewicz traces [6]—seems to be *a priori* the most pertinent choice.

Before describing the transduction machinery, we exhibit several nice properties of this new normal form with, in particular, a deep geometric property concerning the associated so-called Cayley graph. Our work provides a detailed and complete proof to Kuske’s claim that every left divisibility monoid is automatic [18]. Furthermore, we show that every left divisibility monoid is (both left and right) biautomatic, according to Hoffmann’s terminology [12].

The rest of this paper is organized as follows. In Section 2, we recall several definitions about automaticity for monoids. In Section 3, we gather the needed basic properties of left divisibility monoids. Section 4 introduces the right normal form. We show that the language of right normal forms has good properties, preparing the proof of how it provides a biautomatic structure to every left divisibility monoid. In Section 5, we then state the main results of this paper (Theorems 27 and 28), discuss and illustrate them.

2 Background from automaticity of monoids

In this section, we review the theory of automatic monoids—we may focus on cancellative monoids, since left divisibility monoids are defined to be cancellative—and, according to this context, we recall links between automata and transducers. We refer the reader interesting with the notions of automatic structures to [1,11] for automatic groups and to [5,12,13,15,24] for automatic monoids. A general reference about transducers is [2].

2.1 Languages, automata and transducers

We first give a brief introduction to formal language theory (particularly regular languages).

For a finite set X , let X^* denote the set of all finite words over the alphabet X , including the empty word ε . For a word u , let $|u|$ denote the length of u and let $u_{[t]}$ (resp. $u^{[t]}$) denote its length t suffix (resp. prefix) for $t < |u|$ and u itself for $t \geq |u|$.

Some computations on words and on languages can be interpreted as a work of a machine, which being in a state p and receiving as input a letter x , goes into a state q and possibly outputs a word w . Such machines are formalized by the following definitions.

Definition 1 A (deterministic) automaton is a set $\mathcal{A} = (X, Q, q_-, Q_+, \tau)$, where

- (1) X is a finite set (the input alphabet),
- (2) Q is a set (the set of states),
- (3) q_- is a fixed element in Q (the initial state),
- (4) Q_+ is a fixed subset of Q (the set of accepting states), and
- (5) $\tau : X \times Q \rightarrow Q$ is a mapping (the transition function).

The map τ can be extended to $\tau : X^* \times Q \rightarrow Q$ by $\tau(\varepsilon, q) = q$ and $\tau(xu, q) = \tau(u, \tau(x, q))$ for $x \in X$, $u \in X^*$ and $q \in Q$. A word w over X is recognized by \mathcal{A} if $\tau(w, q_-)$ belongs to Q_+ .

Definition 2 A (sequential) transducer is a set $\mathcal{T} = (X, Y, Q, q_-, Q_+, \tau, \lambda)$, where

- (1) (X, Q, q_-, Q_+, τ) is an automaton,
- (2) Y is a finite set (the output alphabet), and
- (3) $\lambda : X \times Q \rightarrow Y^*$ is a mapping (the output function).

The map λ can be extended to $\lambda : X^* \times Q \rightarrow Y^*$ by $\lambda(\varepsilon, q) = \varepsilon$ and $\lambda(xu, q) = \lambda(x, q)\lambda(u, \tau(x, q))$ for $x \in X$, $u \in X^*$ and $q \in Q$.

An automaton (resp. a transducer) is *finite* if the set Q is finite. A finite automaton (resp. a finite transducer) can be represented as a labelled directed graph, known as a *Moore diagram*. The vertices of such a graph correspond to the states of the automaton (resp. the transducer), and, for every letter x of the input alphabet X , an arrow labelled by x (resp. by $x|\lambda(x, q)$) goes from the state $q \in Q$ to the state $\tau(x, q)$. An incoming unlabelled arrow represents the initial state. Accepting states are denoted by double circles.

Definition 3 A language is regular whenever it is the language of words recognized by some finite automaton.

Example 4 For given positive integers b and k , divisibility by k in base b can be decided by a finite automaton, and it turns out that converting integers from base b to base k can be made by using a finite transducer : the latter computes the remainder—the final state—and the quotient—the output—modulo b . Figure 1 displays an automaton which reads from the left to the right and decides whether a base 2 integer is divisible by 3. Figure 2 displays the associated transducer which reads from the left to the right and allows—via multiple runs—to convert an integer from base 2 to base 3.

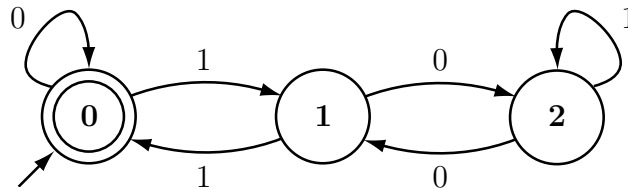


Fig. 1. Automaton deciding division by 3 in base 2.

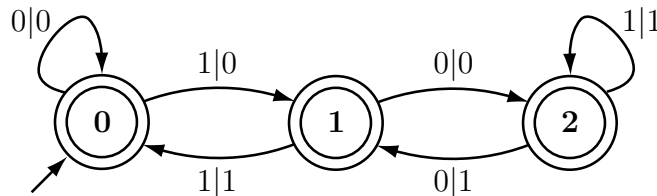


Fig. 2. Transducer allowing conversion from base 2 to base 3.

Remark 5 Most of the transducers we shall consider have only accepting states. This special feature allows to use these transducers iteratively.

2.2 Notions of automatic monoids

First defined by Thurston two decades ago, automatic groups attracted a lot of attention in geometric and combinatorial group theory and are the subject of a major book [11] (see also [1]). Roughly speaking, an automatic group is a finitely generated group for which one can check, by means of a finite automaton, whether two words over a finite generating alphabet represent the same element or not, and whether or not the elements they represent differ by multiplication by a single generator. A few years ago, the notion of automaticity was generalized for semi-groups and monoids : it is worth mentioning the work by Hoffmann in [12] (see also [5,13,15,20]).

As with automatic groups, we may consider automata reading pairs of words where we introduce a *padding symbol* to deal with the case where the lengths of the two words are not the same. One can introduce the paddings on the right or on the left.

Definition 6 For every alphabet X , the mappings $\overrightarrow{\cdot}_X$ and $\overleftarrow{\cdot}_X$ from $X^* \times X^*$ to Y^* with $\$ \notin X$ and $Y = (X \cup \{\$\}) \times (X \cup \{\$\}) \setminus \{(\$, \$)\}$ are defined by

$$\overrightarrow{(x_1 \cdots x_n, y_1 \cdots y_m)}_X = \begin{cases} (x_1, y_1) \cdots (x_n, y_n) & \text{for } n = m, \\ (x_1, y_1) \cdots (x_n, y_n)(\$, y_{n+1}) \cdots (\$, y_m) & \text{for } n < m, \\ (x_1, y_1) \cdots (x_m, y_m)(x_{m+1}, \$) \cdots (x_n, \$) & \text{for } n > m, \end{cases}$$

and $\overleftarrow{(x_1 \cdots x_n, y_1 \cdots y_m)}_X$ to be the mirror of $\overrightarrow{(x_n \cdots x_1, y_m \cdots y_1)}_X$, where the x_i 's and the y_j 's belong to X for $1 \leq i \leq n$ and $1 \leq j \leq m$.

Hoffmann purposed then four notions of automaticity for semigroups : roughly speaking, for π, μ in $\{\text{left}, \text{right}\}$, a semigroup is said to be π - μ -automatic if it is automatic with π the direction of *padding* and μ the direction of *multiplication*.

For a monoid M generating by a set X , there is a canonical mapping $\overline{\cdot} : X^* \rightarrow M$.

Definition 7 Assume that M is a monoid—or a semigroup—generating by a finite set X and that L is a language over X that maps onto M . Then

(X, L) is a left-left automatic structure for M

if ${}^{\$}L = \{\overleftarrow{(u, v)}_X : u, v \in L, x\bar{u} = \bar{v}\}$ is regular for $x \in X \cup \{\varepsilon\}$;

(X, L) is a right-left automatic structure for M

if ${}_xL^{\$} = \{\overrightarrow{(u, v)}_X : u, v \in L, x\bar{u} = \bar{v}\}$ is regular for $x \in X \cup \{\varepsilon\}$;

(X, L) is a left-right automatic structure for M

if ${}^{\$}L_x = \{\overleftarrow{(u, v)}_X : u, v \in L, \bar{u}x = \bar{v}\}$ is regular for $x \in X \cup \{\varepsilon\}$;

(X, L) is a right-right automatic structure for M

if $L_x^{\$} = \{\overrightarrow{(u, v)}_X : u, v \in L, \bar{u}x = \bar{v}\}$ is regular for $x \in X \cup \{\varepsilon\}$.

Those automata accepting such languages are called equality recognizer automata for $x = \varepsilon$ and multiplier automata for $x \in X$.

The notion of *automatic* as defined in [5] for semigroups is equivalent to the notion of *right-right automatic* here. These four notions of automaticity are shown to be independent for general semigroups and to collapse into a dual notion of μ -automaticity for cancellative monoids (whether automaticity implies biautomaticity is still an open question for groups). Roughly speaking, the property of automaticity for a cancellative monoid does not depend on the direction of padding or reading.

Proposition 8 Assume that M is a cancellative monoid generating by a finite set X and L is a language over X that maps onto M . Then, for μ in $\{\text{left}, \text{right}\}$, (X, L) is a left- μ automatic structure for M if and only if (X, L) is a right- μ automatic structure for M .

Remark 9 Hoffmann exhibited an example of a cancellative monoid that satisfies all four of notions of automaticity but which is not right-biautomatic, that is, which does not admit a structure being both right-left and right-right automatic.

2.3 Hoffmann's criterium

The automatic structures for groups are characterized by a geometric condition on the associated Cayley graph, known as the *fellow traveller property* (see [11, Theorem 2.3.5]). Roughly speaking, a formal language L over an alphabet X generating a group G satisfies the fellow traveller property if paths in the Cayley graph of G with respect to X , which are labelled by words in L and which eventually converge to within a distance of 1, never diverge beyond some given distance. If one seeks to apply this condition to monoids, one must decide what one means by *distance* in a monoid Cayley graph. For cancellative monoids, a convenient notion is the following :

Definition 10 Assume that M is a monoid with a finite generating alphabet X . Then the function $d_X : M \times M \rightarrow \mathbb{N} \cup \{\infty\}$ (resp. d^X) defined by

$$d_X(a, b) = \min\{|w| : w \in X^* \text{ and } (\overline{w}a = b \text{ or } a = \overline{w}b)\}$$

(resp. $d^X(a, b) = \min\{|w| : w \in X^* \text{ and } (a\overline{w} = b \text{ or } a = b\overline{w})\}$) is called the left (resp. right) directed distance function of M with respect to X .

Note that such a notion of distance does not satisfy the triangular inequality.

Definition 11 Assume that M is a monoid with a finite generating alphabet X . Then a language L over X is said to satisfy the left (resp. right) directed fellow traveller property (with respect to M) if there exists a positive integer k such that, for any two words u, v in L satisfying $d_X(u, v) \leq 1$ (resp. satisfying $d^X(u, v) \leq 1$), we have $d_X(u_{[t]}, v_{[t]}) < k$ (resp. $d^X(u_{[t]}, v_{[t]}) < k$) for every nonnegative integer t .

What we refer to Hoffmann's criterium is the following result. Let us mention that the original version [12, Proposition 8.3] is stated in term of a semigroup S such that, for every a, b, d in S satisfying $ab = ad, cb = cd$ holds for every c in S . Now, in the case of a monoid, the previous hypothesis is equivalent to left cancellativity.

Theorem 12 Assume that M is a right (resp. left) cancellative monoid with a finite generating alphabet X . Then every regular language L over X mapping onto M and satisfying the left (resp. right) directed fellow traveller property with respect to M provides a left (resp. right) automatic structure for M .

Let us mention that several different geometric conditions characterizing automatic monoids were investigated (see [24] for instance).

3 Background from left divisibility monoids

In this section, we list some basic properties of left divisibility monoids, and summarize results by Droste & Kuske about them. For all the results quoted in this section, we refer the reader to [10,17,18].

3.1 Divisors and multiples in a monoid

Assume that M is a monoid. We say that M is *conical* if 1 is the only invertible element in M . For a, b in M , we say that b is a left divisor of a —or that a is a right multiple of b —if $a = bd$ holds for some d in M . The set of the left divisors of b is denoted by $\downarrow(b)$. An element c is a right lower common multiple—or a right lcm—of a and b if it is a right multiple of both a and b , and every right common multiple of a and b is a right multiple of c . Right divisor and left multiple are defined symmetrically.

If c, c' are two right lcm's of a and b , necessarily c is a left divisor of c' , and c' is a left divisor of c . If we assume M to be conical and cancellative, we have $c = c'$. In this case, the unique right lcm of a and b is denoted by $a \vee b$. Cancellativity and conicity imply that left and right divisibility are order relations.

Let M be a monoid. An *irreducible element* of M is defined to be a non trivial element a such that $a = bc$ implies $b = 1$ or $c = 1$. The set of the irreducible elements in M can be written as $(M \setminus \{1\}) \setminus (M \setminus \{1\})^2$.

3.2 Main definitions and properties for left divisibility monoids

Let (P, \leq) be a partially ordered set. Then, for any a in P , $\downarrow(a)$ comprises all elements dominated by a , that is, $\downarrow(a) = \{b \in P; b \leq a\}$. The *width* of a partially ordered set is the maximal size of an antichain, that is, a subset such that any two distinct elements are incomparable. The partially ordered set (P, \leq) is a lattice if, for any two a, b in P , the least upper bound $\sup(a, b) = a \vee b$ and the largest lower bound $\inf(a, b) = a \wedge b$ exist. The lattice (P, \leq) is *distributive* if $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ for any a, b, c in P . This is equivalent to $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ for any a, b, c in P . For properties of finite distributive lattices, we refer the reader to [4].

Definition 13 A monoid M is called a left divisibility monoid—or simply a divisibility monoid—if M is cancellative and finitely generated by its irreducible elements, if any two elements admit a left gcd and if every element a dominates a finite distributive lattice $\downarrow(a)$.

Note that the finiteness requirement on lattices is in fact not necessary since it follows from the other stipulations. Note also that cancellativity and the lattice condition imply conicity. The left gcd of two elements a, b will be denote by $a \wedge b$. The length $|a|$ of an element a is defined to be the height of the lattice $\downarrow(a)$.

Example 14 *Every (finitely generated) trace monoid is a divisibility monoid. Both the monoids $\langle x, y, z : xy = yz \rangle$ and $\langle x, y, z : x^2 = yz, yx = z^2 \rangle$ are not trace but left divisibility monoids. The monoid $\langle x, y, z : x^2 = yz, xy = z^2 \rangle$ is neither a left nor a right divisibility monoid—a monoid being called a right divisibility monoid if its antiautomorphic image is a left divisibility monoid.*

An easy but crucial fact about left divisibility monoids is the following.

Lemma 15 *Assume that M is a left divisibility monoid. Then finitely many elements in M admitting at least a right common multiple admit a unique right lcm.*

The following result states that there exists a decidable class of presentations that gives rise precisely to all left divisibility monoids.

Theorem 16 *Assume that M is a monoid finitely generated by the set Σ of its irreducible elements. Then M is a left divisibility monoid if and only if*

- (i) $\downarrow(xyz)$ is a distributive lattice,
- (ii) $xyz = xy'z'$ or $yzx = y'z'x$ implies $yz = y'z'$,
- (iii) $xy = x'y'$, $xz = x'z'$ and $y \neq z$ imply $x = x'$,

for any x, y, z, x', y', z' in Σ , and if

- (iv) we have $M \cong \Sigma^*/\sim$, with \sim the congruence on Σ^* generated by the pairs (xy, zt) for x, y, z, t in Σ and $xy = zt$.

Kuske studied a left normal form generalizing the Cartier-Foata normal form known from the theory of Mazurkiewicz traces. This left normal form can be computed by an infinite transducer and Kuske claims in [17] that the latter would allow to prove that every left divisibility monoid is automatic. His main result is that the transducer is finite if and only if the monoid is width-bounded, if and only if the monoid is a regular monoid [23]. We shall come back to Kuske's *infinite* transducer in Remark 33.

4 A right normal form

Our goal being to construct finite transducers allowing to compute normal forms and standard transducers reading words from the left to the right, we shall define a *right* normal form. The right normal form of an element will be defined as a unique decomposition into a product of so-called hypercubes, where the rightmost hypercube is the maximal one, and so on.

Our aim is to show that the right normal form we consider is associated with a biautomatic structure using Hoffmann's criterium.

4.1 Definition of a right normal form

A natural *left* normal form for *left* divisibility monoids is defined in [18] and then called *Foata normal form*. The latter does not seem to be always the best fitted to standard transducers, which reads from the left to the right. One could work with *right* divisibility monoids, but this would devalue the property for left-right reading of being standard. A convenient choice is to construct a *right* normal form for *left* divisibility monoids. Although less natural *a priori*, this choice will be shown to provide equivalent features.

Definition 17 *Assume that M is a left divisibility monoid. An element h in M is called a hypercube if there exist irreducibles x_1, \dots, x_n satisfying $h = x_1 \vee \dots \vee x_n$. By convention, the trivial element 1 is a hypercube.*

Since every finite distributive lattice whose upper bound is the join of its atoms is a hypercube (see [4] or for instance [26, page 107]), a hypercube in a left divisibility monoid is an element h whose lattice $\downarrow(h)$ is a hypercube. In particular, since every interval of a hypercube lattice is a hypercube lattice, every divisor of a hypercube in a left divisibility monoid is a hypercube.

Lemma 18 *Assume that M is a left divisibility monoid. Then every element in M is right-divided by a unique maximal hypercube.*

Proof. Let d be an element in M and $\{x_1, \dots, x_p\}$ be the set of those irreducible elements that divide d on the right. Since d is a common left multiple of the x_i 's, there exists at least one minimal common left multiple of the x_i 's. Such an element is therefore a hypercube, namely a p -cube. Assume now that b and c are two distinct p -cubes dividing d on the right. Then the elements b', c' in M satisfying $d = b'b = c'c$ do not admit a unique right lcm in the lattice $\downarrow(d)$, contradicting Lemma 15. \square

Definition 19 Assume that M is left divisibility monoid. The right normal form of a non-trivial element a in M is the unique decomposition into non-trivial hypercubes $\mathbf{N}(a) = h_p \cdots h_1$ such that $a = h_p \cdots h_1$ holds in M and h_i is the unique maximal hypercube right-dividing $h_p \cdots h_i$ for $1 \leq i \leq p$. Moreover, we set $\mathbf{N}(1) = 1$.

The remainder of this section is devoted to prove several properties of this right normal form. The latter will allow to finally establish that the language of these right normal forms provides a biautomatic structure to every left divisibility monoid.

4.2 Regularity of the language of right normal forms

The first of the two key points is that the normality of a word is characterized by a local condition, what is captured by the following lemma.

Lemma 20 Assume that M is a left divisibility monoid and \mathcal{H} is the set of its hypercubes. Let $\mathbf{h} : M \rightarrow \mathcal{H}$ map an element a to the maximal hypercube right-dividing a . Then $\mathbf{h}(ab) = \mathbf{h}(\mathbf{h}(a)b)$ holds for any two elements a, b in M .

Proof. We use an induction on the length $|b|$ of b . For $|b| = 0$, the formula follows from $\mathbf{h}^2 = \mathbf{h}$. Assume $|b| = 1$. Then b is an irreducible element, say $b = x$. Let y_1, \dots, y_p be the distinct irreducible elements right-dividing a , so right-dividing $\mathbf{h}(a)$ by definition. The distributivity condition implies that, for every j , there exists at most one irreducible element z_{i_j} satisfying $y_j x = t_{i_j} z_{i_j}$ for some irreducible $t_{i_j} \neq y_j$. Therefore $\mathbf{h}(ax)$ and $\mathbf{h}(\mathbf{h}(a)x)$ are the $(q+1)$ -cube right-divided by x, y_1, \dots, y_q with $q \leq p$. We obtain $\mathbf{h}(ax) = \mathbf{h}(\mathbf{h}(a)x)$ for every element a and every irreducible x in M .

Assume now $|b| > 1$. Then there exist an irreducible x and an element b' in M satisfying $b = xb'$, and, by induction hypothesis, we obtain

$$\mathbf{h}(ab) = \mathbf{h}(axb') = \mathbf{h}(\mathbf{h}(ax)b') = \mathbf{h}(\mathbf{h}(\mathbf{h}(a)x)b') = \mathbf{h}(\mathbf{h}(a)xb') = \mathbf{h}(\mathbf{h}(a)b),$$

which concludes the induction. \square

Proposition 21 Assume that M is a left divisibility monoid and \mathcal{H} is the set of its hypercubes. Let h_1, \dots, h_p belong to \mathcal{H} . Then $h_p \cdots h_1$ is a right normal form if and only if so is $h_{i+1} \cdots h_i$ for $1 \leq i < p$.

Proof. Using the formula $\mathbf{h}(ab) = \mathbf{h}(\mathbf{h}(a)b)$ from Lemma 20, we find $\mathbf{h}(h_p \cdots h_i) = \mathbf{h}(h_{i+1}h_i)$ for $1 \leq i < p$. \square

Corollary 22 Assume that M is a left divisibility monoid. Then the language of its right normal forms is regular.

Proof. Our language is over the finite alphabet \mathcal{H} of hypercubes in M . It suffices to take the automaton with \mathcal{H} as set of states and with a transition from a to b whenever $a \cdot b$ is normal, that is, whenever $\mathbf{h}(ab) = b$ holds. \square

Remark 23 *Contrary to the case of trace monoids (see [16, Lemma 3.2]) or braid monoids (see [3, Proposition 4.9]), the graph of hypercubes of a general left divisibility monoid—defined to be the oriented graph with \mathcal{H} as set of vertices and with an edge from a to b whenever $\mathbf{h}(ab) = b$ holds—need not necessarily be strongly connected, even if a condition of irreducibility (see [22]) is required. For instance, in the divisibility monoid $\langle x, y, z : x^2 = yz, y^2 = zx, z^2 = xy \rangle$, there is no right normal form like $z^2 \cdot \dots \cdot x$. The graph of its hypercubes is displayed on Figure 3 (we have omitted the vertices of the two central hypercubes which anyway are not involved in the strong connectivity). Such a graph is known as the graph of cliques in [16] and as Charney’s graph in [3], and its strong connectivity plays a pivotal rôle in the study of trace monoids and braid monoids, respectively.*

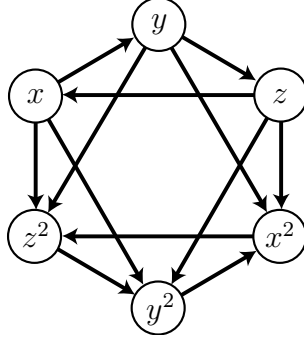


Fig. 3. The graph of hypercubes for $\langle x, y, z : x^2 = yz, y^2 = zx, z^2 = xy \rangle$.

4.3 A fellow traveller property for the language of right normal forms

The second key point is that the right normal forms have a nice behaviour towards both left *and* right multiplication.

Proposition 24 *Assume that M is a left divisibility monoid. Let a be an element in M with right normal form $h_m \cdot \dots \cdot h_1$ and let y, z be hypercubes. Then*

- (i) *the right normal form of ya is $h'_m \cdot \dots \cdot h'_1 \cdot y_0$ (or possibly $h'_{m-1} \cdot \dots \cdot h'_1 \cdot y_0$) with $y_m = y$, $y_{i-1} = \mathbf{h}(y_i h_i)$ and $y_i h_i = h'_i y_{i-1}$ for $1 \leq i \leq m$.*
- (ii) *the right normal form of az is $z_m \cdot h''_m \cdot \dots \cdot h''_1$ (or possibly $h''_m \cdot \dots \cdot h''_1$) with $z_0 = z$, $h''_i = \mathbf{h}(h_i z_{i-1})$ and $z_i h''_i = h_i z_{i-1}$ for $1 \leq i \leq m$.*

Proof. (Figure 4) Using the formula $\mathbf{h}(ab) = \mathbf{h}(\mathbf{h}(a)b)$ from Lemma 20, we obtain

$$\begin{aligned} \mathbf{h}(y_j h_j \cdots h_i) &= \mathbf{h}(\mathbf{h}(y_j h_j) h_{j-1} \cdots h_i) \\ &= \mathbf{h}(y_{j-1} h_{j-1} \cdots h_i) = \dots = \mathbf{h}(y_i h_i) = y_{i-1} \end{aligned}$$

for $1 \leq i < j \leq m$, and

$$\mathbf{h}(h_m \cdots h_i z_{i-1}) = \mathbf{h}(\mathbf{h}(h_m \cdots h_i) z_{i-1}) = \mathbf{h}(h_i z_{i-1}) = h_i''$$

for $1 \leq i \leq m$.

The only point remaining to be checked is that the h_i' 's and z_m are hypercubes. For this, it suffices to show that the right normal form of the product of two hypercubes has length at most two. Assume that a, b are two hypercubes. We denote by a' the element satisfying $ab = a'\mathbf{h}(ab)$. As b is a hypercube, we have $\mathbf{h}(ab) = a''b$ for some hypercube a'' . By right cancellation, we obtain $a = a'a''$. Then a' divides the hypercube a , and, therefore, a' is a hypercube too. This concludes the proof. \square

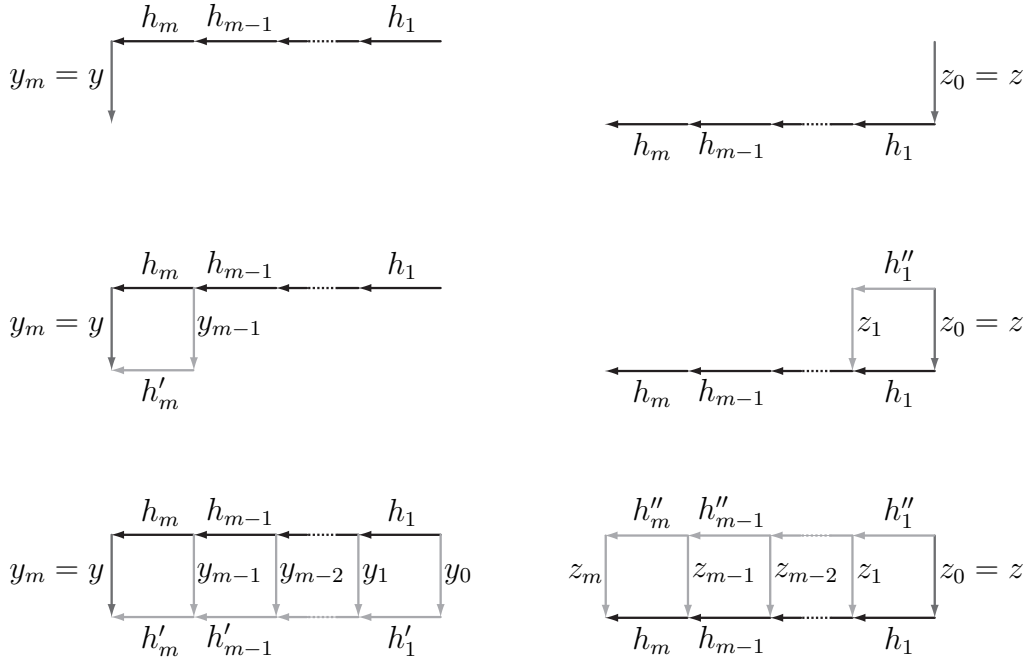


Fig. 4. The right normal forms of ya and az from the right normal form $h_m \cdots h_1$ of a .

Although quite natural, the latter result was not obvious beforehand. Indeed, putting in normal form a product of two hypercubes might have required say three hypercubes, since the condition for being normal discards some decompositions.

Proposition 24 is therefore exactly what we need to show that, in addition to the regularity of the language of its right normal forms, every left divisibility monoid satisfies the required deep geometric property :

Corollary 25 *The language of the right normal forms of every left divisibility monoid satisfies both the left and the right directed fellow traveller properties.*

Proof. We are going to show that, for every hypercube y (*resp.* every hypercube z), the left (*resp.* right) directed distance between the right normal form of an element a and the one of ya (*resp.* the one of az) is uniformly bounded by 2, what will establish the left (*resp.* right) directed fellow traveller property.

The point is to consider the set \mathcal{H} of its hypercubes as a generating alphabet for M .

(i) Let a, b in M satisfy $d_{\mathcal{H}}(a, b) \leq 1$. The result is trivial for $d_{\mathcal{H}}(a, b) = 0$. Assume $d_{\mathcal{H}}(a, b) = 1$. Then we can suppose, without loss of generality, that $b = ya$ holds for some y in \mathcal{H} . Let $h_m \cdot \dots \cdot h_1$ be the right normal form of a . By Proposition 24(i), the right normal form of b is $h'_m \cdot \dots \cdot h'_1 \cdot y_0$ (or possibly $h'_{m-1} \cdot \dots \cdot h'_1 \cdot y_0$) with $y_m = y$, $y_{i-1} = \mathbf{h}(y_i h_i)$ and $y_i h_i = h'_i y_{i-1}$ for $1 \leq i \leq m$. Therefore, for every $t > 0$, we find

$$d_{\mathcal{H}}(a^{[t]}, b^{[t]}) = d_{\mathcal{H}}(h_t \cdot \dots \cdot h_1, h'_{t-1} \cdot \dots \cdot h'_1 \cdot y_0) < 2.$$

Indeed, by definition, we have $y_{t-1} = \mathbf{h}(y_t h_t)$ and, as h_t is a hypercube, we have $y_{t-1} = y'_t h_t$ for some hypercube y'_t : we obtain $y'_t h_t \cdot \dots \cdot h_1 = y_{t-1} h_{t-1} \cdot \dots \cdot h_1 = h'_{t-1} \cdot \dots \cdot h'_1 y_0$.

(ii) Let a, b in M satisfy $d^{\mathcal{H}}(a, b) \leq 1$. The result is trivial for $d^{\mathcal{H}}(a, b) = 0$. Assume $d^{\mathcal{H}}(a, b) = 1$. Then we can suppose, without loss of generality, that $b = az$ holds for some z in \mathcal{H} . Let $h_m \cdot \dots \cdot h_1$ be the right normal form of a . By Proposition 24(ii), the right normal form of b is $z_m \cdot h''_m \cdot \dots \cdot h''_1$ (or possibly $h''_m \cdot \dots \cdot h''_1$) with $z_0 = z$, $h''_i = \mathbf{h}(h_i z_{i-1})$ and $z_i h''_i = h_i z_{i-1}$ for $1 \leq i \leq m$. Two cases may occur. First, assume $z_m = 1$. Then, for every $t > 0$, we find

$$d^{\mathcal{H}}(a^{[t]}, b^{[t]}) = d^{\mathcal{H}}(h_m \cdot \dots \cdot h_{m-t+1}, h''_m \cdot \dots \cdot h''_{m-t+1}) < 2,$$

since we have $h_m \cdot \dots \cdot h_{m-t+1} = h''_m \cdot \dots \cdot h''_{m-t+1} z_{m-t}$. Now assume $z_m \neq 1$. Then, for every $t > 0$, we find

$$d^{\mathcal{H}}(a^{[t]}, b^{[t]}) = d^{\mathcal{H}}(h_m \cdot \dots \cdot h_{m-t+1}, z_m \cdot h''_m \cdot \dots \cdot h''_{m-t+2}) < 2.$$

Indeed, by definition, we have $h''_{m-t+1} = \mathbf{h}(h_{m-t+1} z_{m-t})$ and, as z_{m-t} is a hypercube, we have $h''_{m-t+1} = z'_{m-t} z_{m-t}$ for some hypercube z'_{m-t} : we obtain as required $h_m \cdot \dots \cdot h_{m-t+1} = z_m h''_m \cdot \dots \cdot h''_{m-t+2} z'_{m-t}$. This concludes the proof. \square

5 Biautomaticity and associated finite transducers

In this section, we establish the main theorems of the paper and illustrate them with several examples. We finally discuss about transducers and multiplier automata.

5.1 Biautomaticity

The results from the previous section make us ready to establish the biautomaticity of left divisibility monoids.

Proposition 26 *The language of the right normal forms provides a biautomatic structure to every left divisibility monoid.*

Proof. According to Theorem 12, in a cancellative monoid, every regular language satisfying the directed fellow traveller property for both left and right multiplication provides a biautomatic structure. \square

The latter provides an original and complete proof to the fact that every left divisibility monoid is automatic [18]. Actually, Proposition 26 allows to state :

Theorem 27 *Every left divisibility monoid is biautomatic.*

5.2 Finite transducers computing right normal forms

Following Thurston's original idea concerning the automaticity of the braid groups (see [11]), Dehornoy constructed in [7] an explicit finite transducer computing normal forms in every Garside monoid (see also [21]). We show here that these methods can be adapted to left divisibility monoids. All these transducers work similarly to the one in Example 4.

Theorem 28 *Every left divisibility monoid admits an explicit finite transducer which allows to compute right normal forms in quadratic time.*

Proof. Let M be a left divisibility monoid and Σ the set of its irreducible elements. The transducer can be built as follows. First, the set of the states is exactly the set \mathcal{H} of the hypercubes in M . Next, for every state a in \mathcal{H} and every irreducible x in Σ , there is an arrow from a to the state b defined to be the maximal hypercube in \mathcal{H} right-dividing ax , which is well-defined according to Lemma 18 ; this arrow is then labelled by $x|u$ where u is any word over Σ satisfying $a\bar{x} = \bar{u}b$.

The right normal form $\mathbf{N}(w)$ of a given word w over Σ —formally defined to be $\mathbf{N}(\bar{w})$ according to Definition 19—is then computed as follows. During the reading of w by the just defined transducer, one concatenates the corresponding outputs (eventually empty, namely ε) of trodden arrows. At the end of the reading of w , the ambient state s is the first hypercube of the right normal form $\mathbf{N}(w)$ of w and the

word w' obtained by concatenating the various outputs is the word that remains to be normalized : we have $\mathbf{N}(w) = \mathbf{N}(w') \cdot s$. With λ and τ as in Definitions 1 and 2, we obtain $\mathbf{N}(\varepsilon) = \mathbf{1}$ and $\mathbf{N}(w) = \mathbf{N}(\lambda(w, \mathbf{1})) \cdot \tau(w, \mathbf{1})$ for $w \neq \varepsilon$. \square

Remark 29 Additional arrows allow the just defined transducers to become able to read words over the whole alphabet \mathcal{H} . Precisely, the associated augmented transducer can be defined as follows. Again, the set of the states is \mathcal{H} . Then, for every state a in \mathcal{H} and every hypercube h in \mathcal{H} , there is an arrow from a to the state b defined to be the maximal hypercube in \mathcal{H} right-dividing ah , which is well-defined according to Lemma 18 ; this arrow is then labelled by $h|k$ where k is the hypercube satisfying $ah = kb$, which is well-defined according to Proposition 24.

5.3 Three examples

We apply Theorem 28 on those left divisibility monoids from Example 14 and from Remark 23.

Example 30 Figure 5 displays the 5-state transducer which allows to compute right normal forms in the left divisibility monoid $\langle x, y, z : xy = yz \rangle$. Indeed, the hypercubes are $\mathbf{1} = \vee\{ \}$, $\mathbf{x} = \vee\{x\}$, $\mathbf{y} = \vee\{y\}$, $\mathbf{z} = \vee\{z\}$ and $\mathbf{w} = \overline{xy} = \overline{yz} = \vee\{x, y\}$. The initial state is the state $\mathbf{1}$. Next, the $5 \times 3 = 15$ labelled arrows are constructed as in the previous proof. For instance, there is a arrow between the state \mathbf{w} and the maximal hypercube right-dividing $\mathbf{w}z = \overline{xyz} = \overline{xw}$, namely the state \mathbf{w} itself ; this arrow is then labelled by $z|x$.

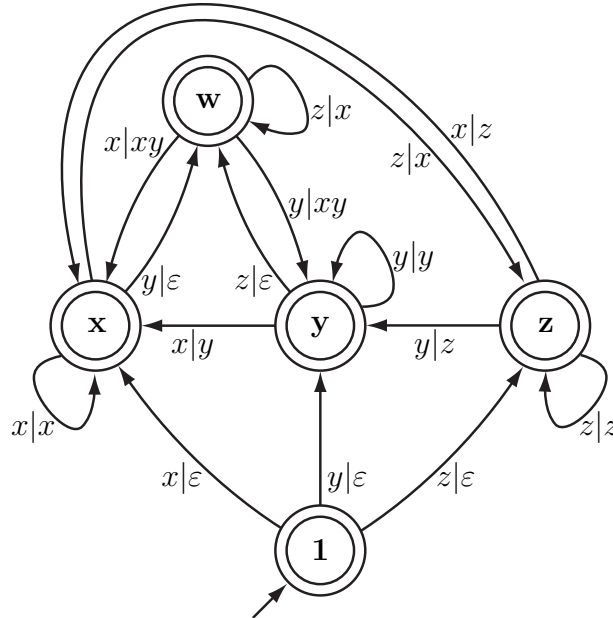


Fig. 5. The transducer for $\langle x, y, z : xy = yz \rangle$.

Example 31 Figure 6 displays the 6-state transducer for the second left divisibility monoid in Example 14, namely the monoid $\langle x, y, z : x^2 = yz, yx = z^2 \rangle$. Let us observe how this transducer allows us to compute the right normal form of a word—say the word $w_0 = yzyxxz$ —in the associated monoid. The reading of w_0 from the state 1 leads to the state $s_1 = z^2$, and the concatenation of the corresponding outputs is the word $w_1 = xxyy$. Therefore, we have $\mathbf{N}(w_0) = \mathbf{N}(w_1) \cdot s_1$. The word w_1 obtained is the word that remains to be normalized : the reading of the word $w_1 = xxyy$ from the state 1 leads to the state $s_2 = y$, and the concatenation of the output labels of trodden arrows is the word $w_2 = xxy$. We obtain $\mathbf{N}(w_0) = \mathbf{N}(w_2) \cdot s_2 \cdot s_1$. Repeating the process twice again, we finally obtain the right normal form $\mathbf{N}(yzyxxz) = x^2 \cdot y \cdot y \cdot z^2$.

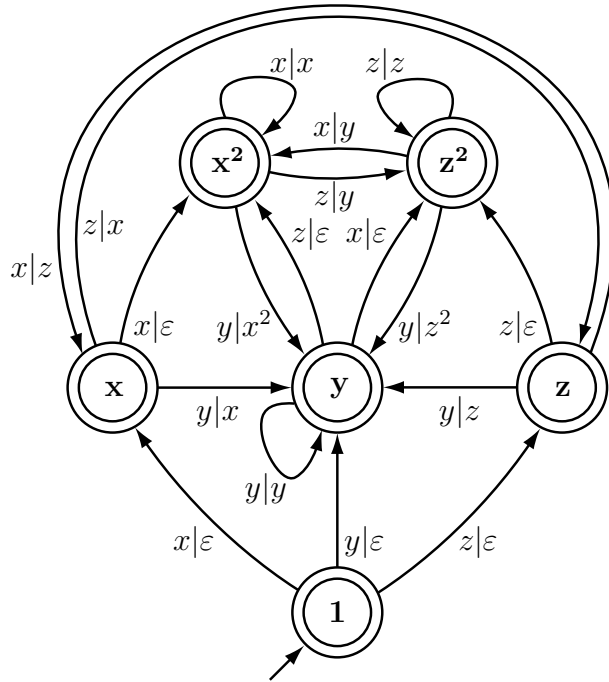


Fig. 6. The transducer for $\langle x, y, z : x^2 = yz, yx = z^2 \rangle$.

Example 32 Figure 7 displays the 8-state transducer allowing to compute right normal forms in the left divisibility monoid $\langle x, y, z : x^2 = yz, y^2 = zx, z^2 = xy \rangle$ introduced in Remark 23. The transducer can be compared with the corresponding graph of hypercubes of Figure 3. Moreover, this example of a divisibility monoid distinguishes from each of the two latter by its non width-boundedness, and so by its non rationality (see [18]). Indeed, for every $n \in \mathbb{N}$, the width of the lattice $\downarrow(x^n)$ equals the number of partitions of n into at most three parts, hence equals the nearest integer to $\frac{(n+3)^2}{12}$ (sequence A001399 from [25]).

Remark 33 The transducer of Figure 5 can be compared to Kuske's transducer of Figure 8 whose relevance was mentioned at the very end of Section 3. Recall that, in this rational case, a normal form is computed through only one run.

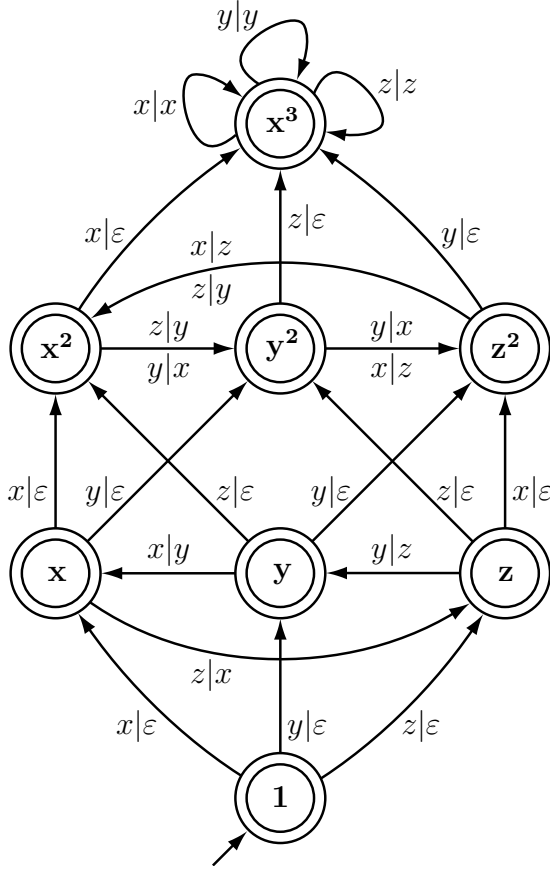


Fig. 7. The transducer for $\langle x, y, z : x^2 = yz, y^2 = zx, z^2 = xy \rangle$.

5.4 Transducers vs multiplier automata

In the context of computing normal forms in divisibility monoids, the afore described transduction machinery seems to afford several advantages over the classical multiplier automata. These advantages revolve around two main ideas : legibility and efficiency. The legibility of our transducer comes from the relative compactness of the data, and, above all, from the fact that its graph structure mimics the lattice structure that the set of hypercubes is endowed with.

As regards efficiency, one can observe that, even if the multiplier automata—which are chosen deterministic—can be viewed and used as transducers, the latter are in general neither subsequential nor even subsequenciable. The time efficiency is known to be substantially increased when subsequential machines are used (see [19] for instance). Moreover, it is worth noticing that our transducers are able to compute the rightmost hypercube of the right normal form in linear time.

In order to illustrate the purpose, the reader is invited to compare Figures 5 and 9. The latter shows the equality recognizer automaton \mathcal{M}_1 and the multiplier automata $\mathcal{M}_x, \mathcal{M}_y, \mathcal{M}_z$ and \mathcal{M}_w . In order to compute the right normal form of a

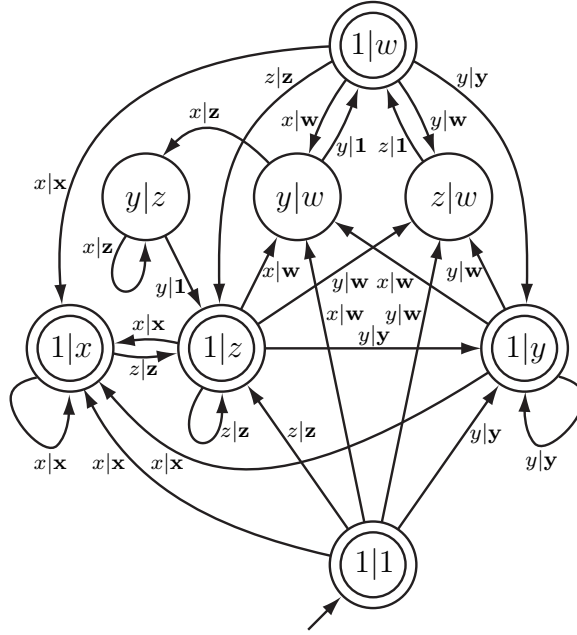


Fig. 8. Kuske's transducer for $\langle x, y, z : xy = yz \rangle$.

word $x_1 \cdots x_n$ over Σ or even over \mathcal{H} , the right normal forms of successive prefixes $x_1 \cdots x_i$ are computed by applying \mathcal{M}_{x_i} to the normal form $x_1 \cdots x_{i-1}$.

Finally, it appears that, in the case of Artin's braid monoids, of Garside monoids, of free partially commutative monoids, and now in the case of divisibility monoids, the biautomaticity and the associated transducers are designed from the particular structure—namely a lattice or semi-lattice structure—of the divisibility relation. Even if a global approach seems to be out of reach, few experimental investigations indicate that wider classes of automatics monoids could be study with similar tools and afford a new insight on the subject.

Acknowledgements

The author is grateful to Manfred Droste, Dietrich Kuske, Sylvain Lombardy, Jean Mairesse and Jacques Sakarovitch for several helpful conversations and the anonymous referees for their careful readings of the paper.

References

- [1] G. Baumslag, S. M. Gersten, M. Shapiro & H. Short, *Automatic groups and amalgams*, J. Pure Appl. Algebra **76** (1991) 239–316.
- [2] J. Berstel, *Transductions and context-free languages*, B.G. Teubner (1979).

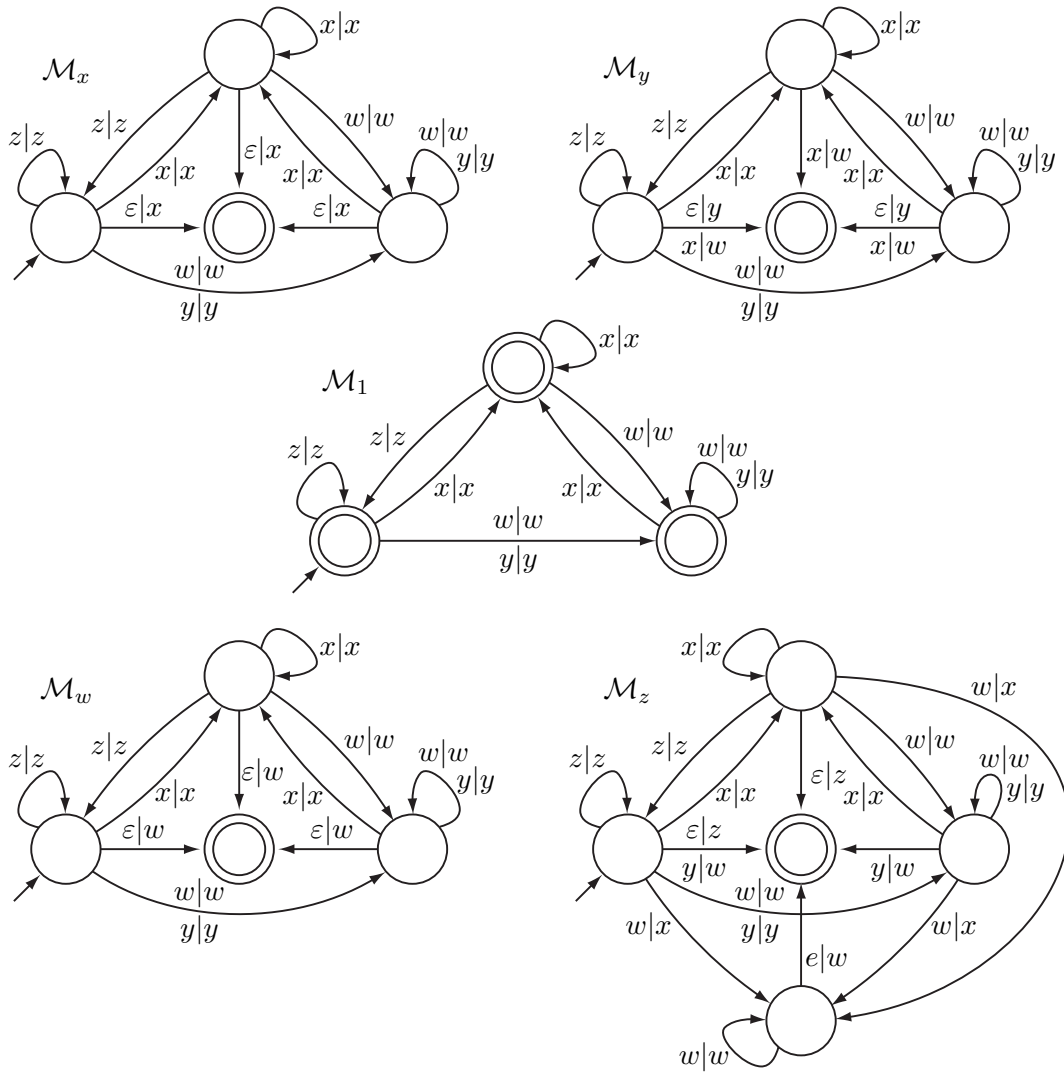


Fig. 9. Multiplier automata for $\langle x, y, z : xy = yz \rangle$.

- [3] M. Bestvina, *Non-positively curved aspects for Artin groups of finite type*, *Geometry & Topology* **3** (1999) 269–302.
- [4] G. Birkhoff, *Lattice theory*, Third Edition, Colloq. Publ. 25, American Math. Soc., Providence (1967).
- [5] C. M. Campbell, E. F. Robertson, N. Ruškuc & R. M. Thomas, *Automatic semigroups*, *Theoretical Computer Science* **250** (2001) 365–391.
- [6] P. Cartier & D. Foata, *Problèmes combinatoires de commutation et réarrangements*, *Lecture Notes in Mathematics* **85** (1969).
- [7] P. Dehornoy, *Groupes de Garside*, *Annales Scientifiques École Normale Supérieure* **35** (2002) 267–306.
- [8] V. Diekert & G. Rozenberg, *The book of traces*, World Scientific, Singapour (1995).
- [9] M. Droste, *Recognizable languages in concurrency monoids*, *Theoretical Computer Science* **150** (1995) 77–109.

- [10] M. Droste & D. Kuske, *Recognizable languages in divisibility monoids*, Mathematical Structures in Computer Science **11** (2001) 743–770.
- [11] D. B. A. Epstein *et al.*, *Word processing in groups*, Jones & Bartlett Publishers, Boston (1992).
- [12] M. Hoffmann, *Automatic semigroups*, PhD Thesis, Dpt of Mathematics and Computer Science, University of Leicester (2000).
- [13] M. Hoffmann & R. M. Thomas, *Notions of automaticity in semigroups*, Semigroup Forum **66** (2003) 337–367.
- [14] J. E. Hopcroft & J. D. Ullmann, *Introduction to automata theory, languages, and computation*, Addison-Wesley (1979).
- [15] J. F. P. Hudson, *Regular rewrite systems and automatic structures*, Semigroups, automata and languages (1994) 145–152.
- [16] D. Krob, J. Mairesse & I. Michos, *Computing the average parallelism in trace monoids*, Discrete Mathematics **273** (2003) 131–162.
- [17] D. Kuske, *On rational and on left divisibility monoids*, Technical Report, TU Dresden MATH-AL-3 (1999) 1–28.
- [18] D. Kuske, *Divisibility monoids : presentation, word problem, and rational languages*, Lecture Notes in Computer Science **2138** (2001) 227–239.
- [19] M. Mohri, *Finite-State Transducers in Language and Speech Processing*, Computational Linguistics **23-2** (1997) 269–311.
- [20] F. Otto & N. Ruškuc, *Confluent monadic string-rewriting systems and automatic structures*, J. Autom. Lang. Comb. **6-3** (2001) 375–388.
- [21] M. Picantin, *Automatic structures for torus link groups*, J. Knot Theory & its Ramifications **12-6** (2003) 833–866.
- [22] —, *Garside monoids vs divisibility monoids*, Mathematical Structures in Computer Science **15-2** (2005) 231–242.
- [23] J. Sakarovitch, *Easy multiplications. I. The realm of Kleene’s theorem*, Information and Computation **74** (1987) 173–197.
- [24] P. V. Silva & B. Steinberg, *A geometric characterization for automatic monoids*, The Quarterly Journal of Mathematics **55-3** (2004) 333–356.
- [25] N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, published electronically at www.research.att.com/~njas/sequences.
- [26] R. P. Stanley, *Enumerative combinatorics*, Camb. Stud. Adv. Math. **49** (1998).