Decidability of geometricity of regular languages*

Marie-Pierre Béal[1], Jean-Marc Champarnaud[2], Jean-Philippe Dubernard[2], Hadrien Jeanne[2], and Sylvain Lombardy[1]

[1] Université Paris-Est, Laboratoire d'informatique Gaspard-Monge CNRS UMR 8049, 5 boulevard Descartes, 77454 Marne-la-Vallée, France.
[2] Université de Rouen, LITIS, Avenue de l'Université - BP 8, 76801 Saint-Étienne-du-Rouvray Cedex, France.

Abstract. Geometrical languages generalize languages introduced to model temporal validation of real-time softwares. We prove that it is decidable whether a regular language is geometrical. This result was previously known for binary languages.

1 Introduction

A geometrical figure of dimension d is a connected set of sites in the lattice of dimension d which is oriented in the following sense: it has an origin O such that for any site P of the figure, there is a directed path with positive elementary step from O to P, a positive elementary step incrementing exactly one coordinate by 1. Finite geometrical figures are called *animals* [10].

A geometrical language is the set of finite words over a d-ary alphabet whose corresponding Parikh points are the sites of a geometrical figure. It is called the geometrical language of the figure. Geometrical languages were introduced by Blanpain *et al.* in [2] and have applications to the modeling of real-time task systems on multiprocessors (see [8], [2]). The definition of geometrical figures implies that all geometrical languages are prefix-closed (*i.e.* the prefix of any word of the language also belongs to the language).

Conversely, for any language of finite words over a d-ary alphabet, one can associate a set of sites corresponding to the Parikh points of the words of the language, the i-th coordinate of the Parikh point of a word counting the number of letters a_i contained in the word. If the language is prefix-closed, the figure that it defines is geometrical. It turns out that a prefix-closed language is always contained in the language of its geometrical figure but this inclusion may be strict, the geometrical languages being exactly the languages satisfying this property.

Studying properties of a geometrical language may help to obtain properties of its geometrical figure and get information on the task systems that it models.

* This work is supported by the French National Agency (ANR) through "Programme d'Investissements d'Avenir" (Project ACRONYME n°ANR-10-LABX-58).

It is also interesting from the language theory point of view. A main subclass of these languages is the one of regular geometrical languages. From this point of view, geometricity is a strong property which can be weakened. The class of semi-geometrical languages contains languages such that any two words with the same Parikh image define the same left residuals.

We consider the class of regular languages and address the algorithmic problem of checking whether a regular language is geometrical (or semi-geometrical). It is already known from [5] that it is decidable in polynomial time whether a regular binary language is geometrical. If n is the number of states of the minimal deterministic automaton accepting the language, an $O(n^3)$-time algorithm is obtained for extensible binary languages in [4], while an $O(n^4)$-time algorithm works for all binary languages in [5]. Two-dimensional geometry is used to prove the correctness of these algorithms. For alphabets in higher dimension, a non-polynomial algorithm has been obtained in the case where the minimal automaton of the language has one strongly connected component [3]. An exponential algorithm in [2] reduces the decidability of geometricity of a regular language to solving a system of Diophantine equations. Nevertheless, the system may be not linear in the general case and solving such a system is known to be undecidable.

In this paper, we give a decision scheme for all regular languages. The algorithm is nevertheless exponential and the existence of a polynomial algorithm to decide the geometricity of a ternary regular language for instance is still open. The problem may be NP-complete but this question is not addressed in the paper. Our solution uses only elementary automata theory and classical semilinear set theory to reduce the problem to a system of linear Diophantine equations. For binary alphabet, we show that a polynomial-time algorithm may be derived from the general solution. The algorithm is simpler that the $O(n^4)$-algorithm obtained in [5] but it has a worst-case time complexity of $O(n^6)$.

The paper is organized as follows. The second section recalls the definitions and main properties of geometrical languages. Section 3 recalls some semilinear set theory [12] useful for Section 4, where the decision procedures are exposed.

2 Geometrical languages

Let d be a positive integer representing a dimension. Let $x = (x_1, .., x_d)$, $y = (y_1, .., y_d)$ be two points in $\mathbb{N}^d$, we say that $x \prec_i y$ (or simply $x \prec y$) if there is exactly one dimension index $1 \le i \le d$ such that $x_i + 1 = y_i$ and $x_j = y_j$ for $j \neq i$.

Let x, y be two points in $\mathbb{N}^d$. We call a *directed path* from x to y a finite sequence of points $(z^{(i)})_{0 \le i \le k}$ contained in $\mathbb{N}^d$ such that $z^{(0)} = x$, $z^{(k)} = y$, and $z^{(i)} \prec z^{(i+1)}$ for $0 \le i \le k-1$.

A *geometrical figure* is either the empty set or a set of points in $\mathbb{N}^d$ containing the null point $(0, .., 0)$ and such that there is a directed path consisting of points belonging to the figure from the null point to any point of the set. Equivalently, for any nonnull point y in a nonempty geometrical figure, there is a point x in the figure such that $x \prec y$.

Let $A = \{a_1, .., a_d\}$ be a finite alphabet of cardinal d. The set of words on the alphabet A is denoted by A^*. The Parikh point associated to a word w of A^* is the point $(x_1, .., x_d)$ in $\mathbb{N}^d$ such that x_i is the number of occurrences of the letter a_i in w.

A *language* L over A is a subset of A^*. We say that a language is *prefix-closed* if any prefix of a word of the language belongs to the language.

The *geometrical figure associated to* a language L, denoted $\mathrm{fig}(L)$, is the set of Parikh points associated to the set of *all prefixes* of words of L. Conversely, the *language associated to a geometrical figure* F, denoted $\mathrm{lang}(F)$, is the set of words whose Parikh points belong to the figure. It is a prefix-closed set.

Let L be a prefix-closed language. We say that L is a *geometrical language* if L is the language associated to some geometrical figure. By extension, if L is not prefix-closed, it is *geometrical* if the set of its prefixes is geometrical. Hence we shall only consider prefix-closed languages.

If F is a geometrical figure, we have $F = \mathrm{fig}(\mathrm{lang}(F))$. If L is a prefix-closed language, we have $L \subseteq \mathrm{lang}(\mathrm{fig}(L))$ but the converse is not true as is shown in the example below.

Example 1. Let L_1 be the language $\{aabbb, aabba, bbaaa, bbaab\}$. The set of prefixes of L_1 is a geometrical language in dimension 2 whose geometrical figure F_1 is pictured in Figure 1. The figure contains the points $(0, 0)$, $(0, 1)$, $(0, 2)$, $(1, 0)$, $(2, 0)$, $(2, 1)$, $(1, 2)$, $(2, 2)$, $(2, 3)$ and $(3, 2)$.

Let now L_2 be the language $\{ab, b\}$. The set of its prefixes is $\{\varepsilon, a, ab, b\}$. It is not geometrical. Indeed the geometrical figure F_2 associated to L_2 contains the points $(0, 0)$, $(0, 1)$, $(1, 0)$, $(1, 1)$. Thus the language associated to F_2 contains the word ba which is not a prefix of a word in L_2.

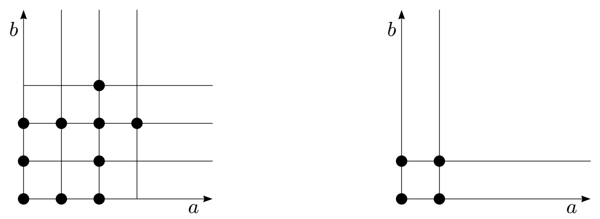

Fig. 1. The geometrical figures F_1 (on the left) and F_2 (on the right).

Proposition 1. *A prefix-closed language L is geometrical if and only if $L = \mathrm{lang}(\mathrm{fig}(L))$.*

Proof. If L is prefix-closed and geometrical, then there is a geometrical figure F such that $L = \mathrm{lang}(F)$. We get $\mathrm{lang}(\mathrm{fig}(L)) = \mathrm{lang}(\mathrm{fig}(\mathrm{lang}(F))) = \mathrm{lang}(F) = L$. Conversely, if $L = \mathrm{lang}(\mathrm{fig}(L))$, it is a geometrical language by definition.

In [4] is introduced the notion of semi-geometricity as follows. If u is a word and L a language, $u^{-1}L$ denotes the set of words w such that uw belongs to L. A prefix-closed language L is said *semi-geometrical* if $u^{-1}L = v^{-1}L$ for any two words u, v of L having the same Parikh point. It is proved in [4] that a geometrical language is semi-geometrical but the converse is false.

Proposition 2 ([4]). *A prefix-closed language which is geometrical is semi-geometrical.*

Proof. Suppose that $L = \mathrm{lang}(F)$ for some geometrical figure F. Let $u, v \in L$ having the same Parikh point. Let w be a word such that $uw \in L$. Then the Parikh point associated to uw belongs to F and the Parikh point associated to any prefix of vw belongs to F. Since $L = \mathrm{lang}(F)$, L contains the word vw. Hence $u^{-1}L \subseteq v^{-1}L$ and thus $u^{-1}L = v^{-1}L$.

A characterization of the geometricity of prefix-closed languages was obtained in [4] as follows.

Proposition 3 ([4]). *A prefix-closed language L over $A = \{a_1, .., a_d\}$ is geometrical if and only if $(ua_i)^{-1}L = (va_j)^{-1}L$ for any words u, v in L and letters a_i, a_j such that ua_i and va_j have the same Parikh point.*

Proof. Suppose that $L = \mathrm{lang}(F)$ is geometrical and w a word such that $ua_iw \in L$. Hence the Parikh point associated to any prefix of ua_iw belongs to F. Since the ua_i and va_j have the same Parikh point and $u, v \in L$, the Parikh point associated to any prefix of va_j belongs to F. It follows that the Parikh point associated to any prefix of va_jw belongs to F. As $L = \mathrm{lang}(F)$, it contains the word va_jw. Hence $ua_iw \in L$ if and only if $va_jw \in L$.

Conversely, let us assume that, for any word w, any words $u, v \in L$, any indexes i, j, we have $ua_iw \in L$ if and only if $ua_jw \in L$. Let $F = \mathrm{fig}(L)$. Let $s = s_1 \cdots s_n$ be a word of length n such that the Parikh point of any prefix of s belongs to F. Let us show that s belongs to L. Since the Parikh point of s_1 belongs to F, we have s_1 belongs to L. Let us assume that the prefix $s_1 \cdots s_k$ of s belongs to L. As the Parikh point x of $s_1 \cdots s_k s_{k+1}$ belongs to F and since $F = \mathrm{fig}(L)$, we get that x is the Parikh point of a word $t = t_1 \cdots t_k t_{k+1}$ in L. Set $u = t_1 \cdots t_k$, $a_i = t_{k+1}$, $v = s_1 \cdots s_k$, $a_j = s_{k+1}$. Since $ua_i \in L$, we get $va_j \in L$ and thus $s_1 \cdots s_k s_{k+1}$ belongs to L. By recurrence, we obtain that s belongs to L.

Note that the proof also shows that L is geometrical if $ua_i \in L$ if and only if $va_j \in L$ for any words u, v in L such that ua_i and va_j have the same Parikh point.

3 Semilinear sets

In this section, we present some definitions and known results about semilinear sets that will be useful in Section 4. We recall some results from [9] and [6] about rational sets of commutative monoids (see for instance [12, 3.3]) or [13, 7.4], [14]), and also [15], [7], [11] for complexity results).

Let $(M, +)$ be a commutative monoid. A *linear set* of M is a set of the form $u + V^{\oplus}$, where $u \in M$, V is a finite subset of M and $V^{\oplus}$ is the submonoid generated by V, *i.e.* the set of linear combinations over $\mathbb{N}$ of elements in V. Hence, if $V = \{v_1, .., v_n\}$, a linear set is a set of the form

$$\{u + x_1 v_1 + \cdots + x_n v_n \mid x_i \in \mathbb{N}, v_i \in V\}.$$

A *semilinear* set is a finite union of linear sets, hence of the form

$$\bigcup_{i=1}^{r} (u_i + V_i^{\oplus}).$$

The set of rational subsets of M contains the finite parts and is closed by the operations union, $+$, and $\oplus$. It is known that the rational subsets of M are exactly its semilinear sets.

Proposition 4. *(see [12, Proposition 3.5]) Let M be a commutative monoid. A subset of M is rational if and only if it is semilinear.*

Furthermore, the construction of a semilinear expression from a rational expression is effective.

We will consider the case where $(M, +)$ is $(\mathbb{Z}^d, +)$. Checking whether a semilinear set of $\mathbb{Z}^d$ is empty or not is known to be decidable. It can be first reduced to the problem of checking whether a linear set is empty or not, which is decidable and NP-complete. A proof of the following Proposition can be found for instance in [12, Lemma 3.10] or in [13, Proposition 7.17].

Proposition 5. *It is decidable whether the equation*

$$x_1 u_1 + \cdots + x_k u_k = c,$$

where $u_i, c \in \mathbb{Z}^d$, has a solution in $\mathbb{N}^k$.

In [16] is proved that, if a solution exists, then there is one with coefficients bounded above by $(k+1)M_1$, where M_1 is the maximum of the absolute values of all sub-determinants of a $d \times (k+1)$ matrix made of the coefficients of u_i and c.

4 Regular geometrical languages

In this section, we address the problem of checking whether a regular language is geometrical. We do not make any restrictions on the dimension or on properties of the regular language or on its minimal deterministic automaton.

We consider a regular prefix-closed language L on the alphabet $A = \{a_1, .., a_d\}$. It is accepted by a unique minimal finite complete deterministic automaton $\mathcal{A} = (Q, E, q_0, T)$, where Q is the set of states and E the set of edges. The unique initial state is q_0 and the set of final states is T. If L is the full language, we have $Q = F = \{q_0\}$. Otherwise, Q has a non final sink state q_s and all states but q_s are final since L is prefix-closed. We denote by $\delta(q, u)$ the state ending the unique path labeled u starting at q.

By definition of the semi-geometricity, we get from Proposition 2 the following characterization of semi-geometrical regular prefix-closed languages.

Proposition 6 ([4]). *A regular prefix-closed language L is semi-geometrical if and only if $\delta(q_0, u) = \delta(q_0, v)$ for any two words u, v of L having the same Parikh point.*

It also follows directly from Proposition 3 the following characterization of geometrical regular prefix-closed languages.

Proposition 7 ([4]). *A regular prefix-closed language L is geometrical if and only if $\delta(q_0, ua_i) = \delta(q_0, va_j)$ for any words u, v in L such that ua_i and va_j have the same Parikh point.*

The main result of the paper is the following.

Proposition 8. *It is decidable whether a regular prefix-closed language is geometrical (resp. semi-geometrical).*

Proof. Let $\mathcal{A} = (Q, E, q_0, T)$ be the minimal deterministic complete automaton accepting the language L. We consider the automaton $\mathcal{B} = (Q \times Q, E', (q_0, q_0), T \times T)$ labeled on $\mathbb{Z}^d$, where the edges are defined as follows. There is an edge

$$(p, q) \xrightarrow{(0,..,\overset{\overset{i}{\downarrow}}{+1},..,\overset{\overset{j}{\downarrow}}{-1},..0)} (p', q')$$

with $+1$ positioned at the index i and -1 at the index j, whenever there are two edges in $\mathcal{A}$

$$p \xrightarrow{a_i} p' \quad \text{and} \quad q \xrightarrow{a_j} q'.$$

There is an edge

$$(p, q) \xrightarrow{(0,..,0)} (p', q')$$

whenever there are two edges in $\mathcal{A}$

$$p \xrightarrow{a_i} p' \quad \text{and} \quad q \xrightarrow{a_i} q'.$$

The automaton $\mathcal{B}$ accepts a regular set of $\mathbb{Z}^d$.

By construction, there is a path in $\mathcal{B}$ from (q_0, q_0) to (p, q) labeled by the null vector of $\mathbb{Z}^d$ if and only if there are two words u, v with the same Parikh point such that $\delta(q_0, u) = p$ and $\delta(q_0, v) = q$. Let $B_{(p,q)}$ denote the regular subset of $\mathbb{Z}^d$ of labels of paths of $\mathcal{B}$ from (q_0, q_0) to (p, q). Thus checking whether L is semi-geometrical consists in checking whether there exists no pair of states (p, q) with $p \neq q$ and p, q final, such that $B_{(p,q)}$ contains the null vector.

Similarly, there is a path in $\mathcal{B}$ from (q_0, q_0) to (p, q) labeled by the $\mathbb{Z}^d$-vector $\mathbf{x}_{(i,j)} = (0, \ldots, 0, -1, 0, \ldots, 0, +1, 0, \ldots, 0)$ (with -1 positioned at the index i and $+1$ at the index j) if and only if there are two words u, v such that $\delta(q_0, u) = p$ and $\delta(q_0, v) = q$, and such that ua_i and va_j have the same Parikh point. Thus checking whether L is geometrical consists in checking whether, when $B_{(p,q)}$ contains $\mathbf{x}_{(i,j)}$ for some pair of states (p, q) with $p \neq q$ and p, q final, we have $\delta(p, a_i) = \delta(q, a_j)$.

As a consequence both geometricity and semi-geometricity can be reduced to check whether the regular language $B_{(p,q)}$ of $\mathbb{Z}^d$ contains a given point of $\mathbb{Z}^d$. If we find such a language $B_{(p,q)}$ containing $\mathbf{x}_{(i,j)}$, we check whether $\delta(p, a_i) = \delta(q, a_j)$ and conclude that L is not geometrical if this condition does not hold.

We know from Section 3 that any set $B_{(p,q)}$ is semilinear, and the effective construction of Proposition 5 can be performed a finite number of times to decide whether $B_{(p,q)}$ contains some vector $\mathbf{x}_{(i,j)}$. Thus both geometricity and semi-geometricity are decidable.

The time complexity of the algorithm is exponential. Indeed, the automaton $\mathcal{A}$ being given, the construction of $\mathcal{B}$ can be done in polynomial time. Finding a rational expression of a set $B_{(p,q)}$ is exponential (the size of the expression itself can be exponential). Finding a semi-linear expression from a rational expression is a polynomial step. Finally, solving a linear Diophantine equation is exponential.

Example 2. We consider again the language $L_2 = \{ab, b\}$. The set of its prefixes $\{\varepsilon, a, ab, b\}$ is accepted by the minimal deterministic complete finite automaton $\mathcal{A}$ pictured in the left part of Figure 2. The automaton $\mathcal{B}$ constructed in the proof of Proposition 8 is pictured in the right part. We have $B_{(2,3)} = \{(1, -1)\}$. It contains $(1, -1)$ and $\delta(2, b) \neq \delta(3, a)$. As a consequence L_2 is not geometrical. It is semi-geometrical since neither $B_{(2,3)}$ nor $B_{(3,2)}$ contains the null vector.

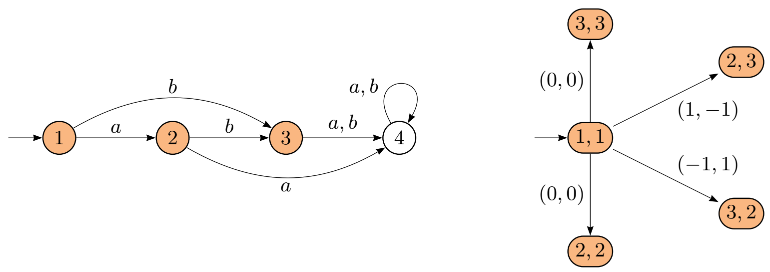

Fig. 2. The automaton $\mathcal{A}$ (on the left), where the final states are colored, accepting the set of prefixes of L_2, and the automaton $\mathcal{B}$ (on the right). Only the final states of $\mathcal{B}$ are represented.

We now come to the particular case of a two-letter alphabet $A = \{a, b\}$. It is proved in [5] that it is decidable in polynomial time whether a regular binary language is geometrical. An $O(n^3)$-time algorithm is obtained for an extensible binary language in [4], an $O(n^4)$-time algorithm works for all binary languages in [5]. We give below another polynomial-time algorithm for deciding the geometricity of binary regular languages which is based on the construction used in the proof of Proposition 8. It also uses an algorithm of [1] for computing the closure of an automaton under some rewriting rules. This algorithm has an $O(n^6)$ time complexity which is worse than the complexity of the algorithm given in [5], but it is simpler.

Proposition 9. *([5]) It is decidable in polynomial time whether a regular prefix-closed language on a two letter alphabet is geometrical (resp. semi-geometrical).*

Proof. Let $\mathcal{A} = (Q, E, q_0, T)$ be the n-state minimal deterministic complete automaton accepting the language L. We first construct an automaton $\mathcal{B}'$ over $\mathbb{Z}$ which plays the same role as the automaton $\mathcal{B}$ in the proof of Proposition 8 but has its labels in $\mathbb{Z}^{d-1}$. Let $\mathcal{B}' = (Q \times Q, E', (q_0, q_0), T \times T)$ labeled on $\mathbb{Z}$. The edges of $\mathcal{B}'$ are defined as follows. There is in $\mathcal{B}'$ an edge

$$\begin{aligned}
(p, q) &\xrightarrow{+1} (p', q') \quad \text{if } p \xrightarrow{a} p' \text{ and } q \xrightarrow{b} q' \text{ are edges of } \mathcal{A},\\
(p, q) &\xrightarrow{-1} (p', q') \quad \text{if } p \xrightarrow{b} p' \text{ and } q \xrightarrow{a} q' \text{ are edges of } \mathcal{A},\\
(p, q) &\xrightarrow{0} (p', q') \quad \text{if } p \xrightarrow{\ell} p' \text{ and } q \xrightarrow{\ell} q' \text{ are edges of } \mathcal{A},
\end{aligned}$$

where $\ell = a$ or $\ell = b$.

Let $B'_{(p,q)}$ denote the regular subset of $\mathbb{Z}$ of labels of paths of $\mathcal{B}'$ from (q_0, q_0) to (p, q). There is a path in $\mathcal{B}'$ from (q_0, q_0) to (p, q) labeled by -1 if and only if there are two words u, v such that $\delta(q_0, u) = p$ and $\delta(q_0, v) = q$ and such that ua_i and va_j have the same Parikh point. Thus checking whether L is geometrical consists in checking whether when $B'_{(p,q)}$ contains -1 for some pair of states (p, q) with p, q final, we have $\delta(p, a_i) = \delta(q, a_j)$. Note that $B'_{(p,q)}$ contains -1 if and only if $B'_{(q,p)}$ contains 1. Adding an extra initial edge labeled 1 reduces the problem to checking whether $B'_{(p,q)}$ contains 0.

The automaton $\mathcal{B}'$ is an n^2-state non-deterministic automaton labeled in the subset $X = \{-1, 0, 1\}$ of the group $\mathbb{Z}$. By definition, the number of transitions of $\mathcal{B}'$ is at most $4n^2$. We say that the pair of consecutive edges of $\mathcal{B}'$

$$s \xrightarrow{\ell} t \xrightarrow{m} u,$$

is *reducible* if $\ell + m \in X$.

We construct an automaton $\mathcal{C}$ which is a closure of $\mathcal{B}'$ in the following sense. Whenever there is a reducible pair of consecutive edges of $\mathcal{B}'$ as above, we add in $\mathcal{C}$ the edge

$$s \xrightarrow{\ell+m} u.$$

This construction is an instance of the algorithm used in [1] for computing the set of descendants of a regular set for Thue systems of a certain type. The rewriting rules that we consider are given by pairs of words in $X^* \times X^*$ which are $((-1)1, 0), (1(-1), 0)$, $(00, 0), (01, 1)$, $(10, 1)$, $((-1)0, -1), (0(-1), -1)$.

The computation of the automaton $\mathcal{C}$ can be done as follows. We keep a queue of edges of $\mathcal{C}$ containing initially the edges of $\mathcal{B}'$. For each edge $e = s \xrightarrow{\ell} t$ of this queue, we consider the edges $f = t \xrightarrow{m} u$ following e and the edges $g = u \xrightarrow{m} s$ preceding e, in order to check whether ef of fe is a reducible pair of edges. In that case, we add a new edge in the queue $s \xrightarrow{\ell+m} u$ (or $u \xrightarrow{\ell+m} t$).

The number of edges of $\mathcal{C}$ is at most $3n^4$ and each edge is added and removed only once in the queue. Whenever an edge (s, ℓ, t) is removed, the edges going out of t and coming in s are checked. There are at most $6n^2$ such edges. Thus the time complexity the algorithm is $O(18n^6)$.

We claim that there is a path in $\mathcal{B}'$ from s to t labeled by 0 if and only if there is an edge in $\mathcal{C}$ from s to t labeled by 0. Indeed, by construction, if there is an edge in $\mathcal{C}$ from s to t labeled by 0, then there is a path in $\mathcal{B}'$ from s to t labeled by 0. Conversely, let

$$s \xrightarrow{\ell_1} s_1 \xrightarrow{\ell_2} \ldots \xrightarrow{\ell_r} s_r = t$$

be a path in $\mathcal{B}'$ labeled by 0 of minimal length. This path contains no consecutive reducible pair of edges as factor since otherwise we could get a shorter path labeled with the same label, origin and end. As a consequence the factors $(-1)1$, $1(-1)$, 01, 10, $0(-1)$, $(-1)0$, or 00, are forbidden in the sequence $\ell_1 \ldots \ell_r$. This implies that all ℓ_i are equal. Since $\ell_1 + \cdots + \ell_r = 0$, we get $r = 1$ and $\ell_1 = 0$.

The algorithm can be implemented as follows. Let $\ell \in X$. We set $B'_\ell[s, t] =$ true if there is an edge (s, ℓ, t) in $\mathcal{B}'$ and $B'_\ell[s, t] =$ false otherwise. We define the matrices C_ℓ similarly.

A pseudocode for computing the matrices C_ℓ from the matrices B'_ℓ is given in the procedure CLOSURE below.

CLOSURE (transition matrices B'_ℓ)

for all $\ell \in X$
    do $C_\ell \leftarrow B'_\ell$
edgeQueue $\leftarrow$ the edges of $\mathcal{B}'$
while *edgeQueue* is nonempty
    do remove an edge $s \xrightarrow{\ell} t$ from *edgeQueue*
        for all states u, all m such that $l + m \in X$,
            do if $C_{\ell+m}[s, u] =$ false
                then $C_{\ell+m}[s, u] \leftarrow$ true
                    add $s \xrightarrow{\ell+m} u$ to *edgeQueue*
              if $C_{\ell+m}[u, t] =$ false
                then $C_{\ell+m}[u, t] \leftarrow$ true
                    add $u \xrightarrow{\ell+m} t$ to *edgeQueue*
return C_ℓ

Example 3. We consider the language $L_3 = \{ab, ba\}^*$. The set of its prefixes is accepted by the deterministic complete finite automaton $\mathcal{A}$ pictured in the left part of Figure 3. The automaton $\mathcal{B}'$ constructed in the proof of Proposition 9 is pictured in the right part of the figure. The closure automaton $\mathcal{C}$ of $\mathcal{B}'$ is pictured in Figure 4. Since $\mathcal{C}$ has no edge labeled by 0 from $(1,1)$ to either $(2,3)$ or $(3,2)$, the language L_3 is a geometrical language.

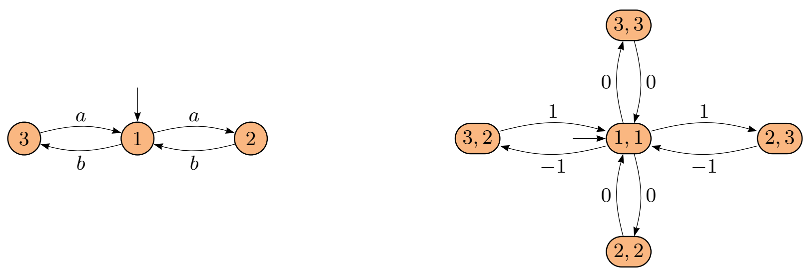

Fig. 3. The automaton $\mathcal{A}$ (on the left) accepting the set of prefixes of $L_3 = \{ab, ba\}^*$, and the automaton $\mathcal{B}'$ (on the right). Only the final states are represented.

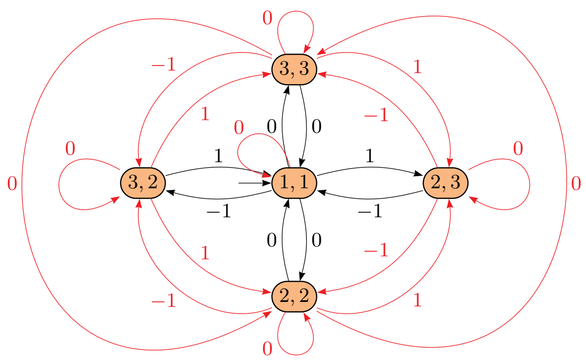

Fig. 4. The automaton $\mathcal{C}$ which is the closure of the automaton $\mathcal{B}'$.

References

1. M. Benois and J. Sakarovitch, *On the complexity of some extended word problems defined by cancellation rules*, Inform. Process. Lett., 23 (1986), pp. 281–287.
2. B. Blanpain, J.-M. Champarnaud, and J.-P. Dubernard, *Geometrical languages*, in LATA (Languages and Automata Theoretical Aspects), vol. Preproceedings, Report 35/07 of GRLMC Universitat Rovira I Virgili, 2007, pp. 127–138.

3. J.-M. Champarnaud, J.-P. Dubernard, F. Guingne, and H. Jeanne, *Geometrical regular languages and linear diophantine equations*, in DCFS, 2011, pp. 107–120.
4. J.-M. Champarnaud, J.-P. Dubernard, and H. Jeanne, *An efficient algorithm to test whether a binary and prolongeable regular language is geometrical*, Int. J. Found. Comput. Sci., 20 (2009), pp. 763–774.
5. ———, *Geometricity of binary regular languages*, in LATA, 2010, pp. 178–189.
6. S. Eilenberg and M. P. Schützenberger, *Rational sets in commutative monoids*, J. Algebra, 13 (1969), pp. 173–191.
7. M. J. Fischer and M. O. Rabin, *Super-exponential complexity of Presburger arithmetic*, in Complexity of computation (Proc. SIAM-AMS Sympos., New York, 1973), Amer. Math. Soc., Providence, R.I., 1974, pp. 27–41. SIAM–AMS Proc., Vol. VII.
8. D. Geniet and G. Largeteau, *WCET free time analysis of hard real-time systems on multiprocessors: a regular language-based model*, Theoret. Comput. Sci., 388 (2007), pp. 26–52.
9. S. Ginsburg and E. H. Spanier, *Bounded* ALGOL-*like languages*, Trans. Amer. Math. Soc., 113 (1964), pp. 333–368.
10. S. W. Golomb, *Polyominoes*, Princeton University Press, Princeton, NJ, second ed., 1994. Puzzles, patterns, problems, and packings, With diagrams by Warren Lushbaugh, With an appendix by Andy Liu.
11. T.-D. Huynh, *The complexity of semilinear sets*, in Automata, languages and programming (Proc. Seventh Internat. Colloq., Noordwijkerhout, 1980), vol. 85 of Lecture Notes in Comput. Sci., Springer, Berlin, 1980, pp. 324–337.
12. C. Reutenauer, *Aspects mathématiques des réseaux de Petri*, Collection Études et Recherches en Informatique, Masson, Paris, 1989, ch. 3.
13. J. Sakarovitch, *Éléments de théorie des automates*, Vuibert, 2003.
14. ———, *Elements of Automata Theory*, Cambridge University Press, 2009.
15. J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, Cambridge, UK, second ed., 2003. First edition 1999.
16. J. von zur Gathen and M. Sieveking, *A bound on solutions of linear integer equalities and inequalities*, Proceedings of the American Mathematical Society, 72 (1978), pp. 155–158.