Mathematical Structures in Computer Science (2019), **29**, pp. 1428–1443
doi:10.1017/S0960129519000057

CAMBRIDGE UNIVERSITY PRESS

PAPER

Decidability of regular language genus computation

Guillaume Bonfante[1] and Florian L. Deloup[2,*]

[1]LORIA, Université de Lorraine, Campus scientifique, B.P. 239, 54506 Vandoeuvre-lès-Nancy Cedex and [2]IMT, Université Paul Sabatier – Toulouse III, 118 Route de Narbonne, 31069 Toulouse cedex 9

*Corresponding author. Email: florian.deloup@math.univ-toulouse.fr

(Received 06 October 2016; revised 28 October 2018; accepted 20 February 2019; first published online 10 April 2019)

Abstract

This article continues the study of the genus of regular languages that the authors introduced in a 2013 paper (published in 2018). In order to understand further the genus $g(L)$ of a regular language L, we introduce the genus size of $|L|_{\text{gen}}$ to be the minimal size of all finite deterministic automata of genus $g(L)$ computing L. We show that the minimal finite deterministic automaton of a regular language can be arbitrarily far away from a finite deterministic automaton realizing the minimal genus and computing the same language, in terms of both the difference of genera and the difference in size. In particular, we show that the genus size $|L|_{\text{gen}}$ can grow at least exponentially in size $|L|$. We conjecture, however, the genus of every regular language to be computable. This conjecture implies in particular that the planarity of a regular language is decidable, a question asked in 1976 by R. V. Book and A. K. Chandra. We prove here the conjecture for a fairly generic class of regular languages having no short cycles. The methods developed for the proof are used to produce new genus-based hierarchies of regular languages and in particular, we show a new family of regular languages on a two-letter alphabet having arbitrary high genus.

Keywords: Deterministic finite automaton; regular language; topological genus; graph embedding; planarity

1. Introduction

Regular languages form a robust and well-studied class of languages: they are recognized by deterministic finite automata (DFA), as well as various formalisms such as Monadic Second-Order logic, finite monoids, regular expressions. Traditionally, the canonical measure of the complexity of a regular language is given by the number of states of its minimal deterministic automaton.

In this paper, we study an alternative measure of language complexity, with a more topological flavor. We will be interested in the topological genus of underlying graph structures of deterministic automata recognizing the language. A surface is a 2-manifold. We shall consider only connected compact and oriented surfaces in this paper. Recall that the genus of a surface Σ is the maximal number of mutually disjoint simple closed curves $C_1, \ldots, C_g \subset \Sigma$ such that the complement $\Sigma - (C_1 \cup \cdots \cup C_g)$ remains connected. This yields a natural notion of genus of a graph: a graph has genus n if it is embeddable in a surface of genus n but cannot be embedded in a surface of strictly smaller genus.

This definition was used in Bonfante and Deloup (2018) to define the genus of a regular language L as the minimal genus among the genera of all underlying graphs of deterministic automata recognizing L. In particular, L has genus 0 if and only if it can be recognized by a planar deterministic automaton.

One of the main questions is the computability of the genus of a regular language (Conjecture 4 below). This conjecture implies the decidability of the planarity of a regular language – a question raised in 1976 by R. V. Book and A.K. Chandra (Book and Chandra 1976). Our main result is the proof of the conjecture for the class of regular languages having no short cycle (Theorem 4).

The complexity of the computation of the genus is reflected on the cost of extra states needed to build a deterministic automaton of minimal genus. We show that the number of states required may be exponential in the size of the minimal automaton of the language (Theorem 1).

In an earlier paper Bonfante and Deloup (2018), we proved that there are regular languages of arbitrary high genus. We also provide new hierarchies of regular languages based on the genus, including for regular languages on two letters (Theorem 6).

Plan of the paper. Section 2 provides introductory examples (for the reader familiar with automata theory), background, definitions of genus and genus sizes, and examples, notably an example of language which features an exponential gap between its size and its genus size (Theorem 1). Section 3 is the most technical part of the paper: it introduces the notion of a language without short cycles. Within the class of languages without short cycles, we find a lower bound for the genus of the language in terms of the size of the language (Theorem 2). Section 4 states the computability conjecture of the genus of any regular language. The main results are the finiteness of complete DFA without short cycles of given genus (Theorem 3) and the computability of the genus of a regular language without short cycles (Theorem 4). We also provide examples where we show that the hypothesis of absence of short cycles cannot be removed (Proposition 1). Section 5 provides two new examples of genus-based hierarchies: in the first example, we give an exact closed formula for the genus (Theorem 5); the second example is a genus-based hierarchy of two-letter languages (Theorem 6). Section 6 contains the proof that the absence of short cycles in the minimal complete finite deterministic automaton is a property of the underlying language, a fact often used in this paper.

2. The genus and genus size of a regular language

2.1 Introductory examples

This paragraph is intended to provide motivation for the reader familiar with automata theory. For background and references, see Section 2.2. The Myhill–Nerode theorem provides constructive existence and uniqueness of a deterministic finite automaton with minimal number of states recognizing a given regular language.

Definition 1. *For each $k \geq 1$, we define the regular language on the alphabet $\mathbb{Z}/k\mathbb{Z}$:*

$$Z_k := \left\{ a_1 a_2 \dots a_n \mid \sum_{i=1}^{n} a_i \equiv 0 \mod k \right\}.$$

It will be convenient to denote $Z_k^{a_1,\dots,a_r}$ the regular language obtained from Z_k by restriction to the subalphabet $\{a_1, \dots, a_r\} \subseteq \mathbb{Z}/k\mathbb{Z}$.

Example 1. The language $Z_5^{0,1,2}$. Figure 1 depicts the minimal automaton $\mathtt{A}$. The transitions are of the form $i \xrightarrow{j} i + j \mod 5$. Since it contains the complete graph K_5, $\mathtt{A}$ is not planar. However, there exists a deterministic automaton with six states that is planar and computes the same language L: see Figure 2.

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press

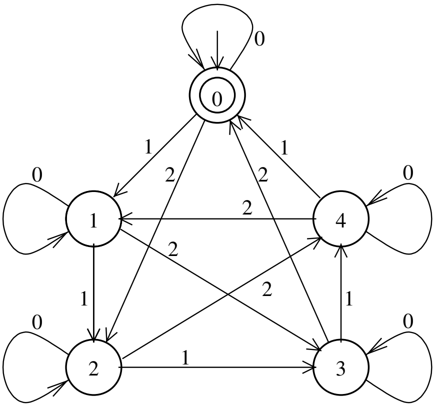


Figure 1. The minimal automaton for the language $Z_5^{0,1,2}$.

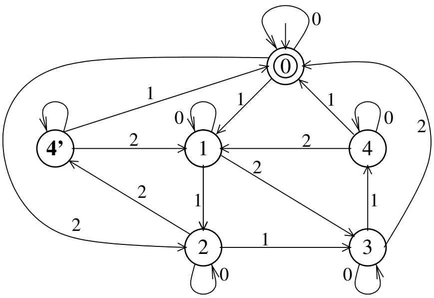


Figure 2. A planar automaton $\mathtt{B}$ computing L. Note that states 4 and $4'$ are equivalent: they produce the same output (they are the sources of the same transitions) and merging them yields back the previous automaton (see Figure 1).

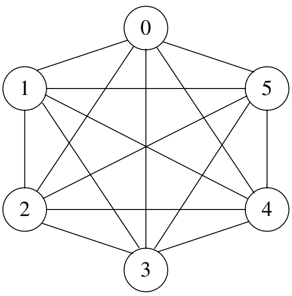


Figure 3. The minimal automaton of Z_6. For simplicity, the self-loop labeled 0 at each vertex is omitted and each edge represents two transitions in opposite directions.

In the previous example, adding just an extra state suffices to produce a planar automaton that recognizes the same language. The following example suggests that the general case may require many more states.

Example 2. The language Z_6. Figure 3 represents the minimal deterministic finite automaton $\mathtt{A}$ computing Z_6. Its state space is $\mathbb{Z}/6\mathbb{Z}$ and its transitions are $i \xrightarrow{j} i+j \mod 6$, for all $i, j \in \mathbb{Z}/6\mathbb{Z}$.

There is no planar representation for $\mathtt{A}$. (Since $\mathtt{A}$ has the complete graph K_5 as a minor, $\mathtt{A}$ is not planar.) However, there exists a deterministic automaton with 12 states that is planar and computes the same language L (Figure 4). We regard the additional six states as the price to pay in order to simplify the topology of an embedding of the automaton into a surface. Since any 6-state automaton has an underlying graph which is a subgraph of Z_6, it follows that any language of size $|L| \leq 6$ (which admits an automaton representation with 6 states or less) can be represented by a planar finite deterministic automaton with at most 12 states. (A detailed proof of this fact and a generalization of it will be published in another forthcoming paper.)

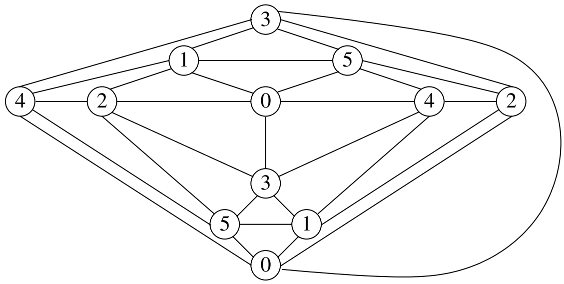


Figure 4. A deterministic automaton of minimal genus (planar) recognizing the same language Z_6 (with the same representation conventions as in Figure 3).

2.2 Automata and graphs

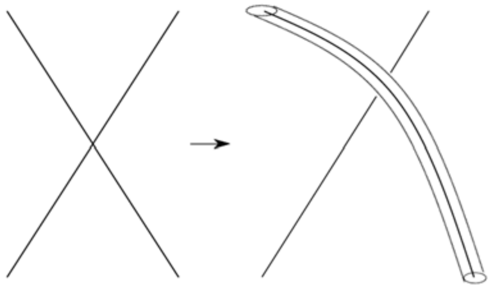
For a general reference on automata, we refer to Sakarovitch (2009). For a general reference on graphs, we refer to Gross and Tucker (2001). Here we give a brief review of the few notions used in this paper.

An automaton $\mathtt{A}$ consists of a set Q of states, with one distinguished initial state $i \in Q$ and a subset F of distinguished final states, a finite set (alphabet) $\mathcal{A}$ and a subset (the set of transitions) $T \subseteq Q \times \mathcal{A} \times Q$. The automaton is *finite* if the set of states is finite. The *size* of a finite automaton $\mathtt{A}$, denoted $|\mathtt{A}|$, is the number of states of $\mathtt{A}$, i.e., the cardinality of Q. A state $q \in Q$ is *accessible* if there is a sequence of transitions $(i, a_1, q_1), (q_1, a_2, q_2), \ldots, (q_{n-1}, a_n, q)$ connecting the initial state to the state q. A state $q \in Q$ is *co-accessible* if there is a sequence of transitions $(q, a_1, q_1), \ldots, (q_{n-1}, a_n, f)$ with $q \in F$ connecting the state q to some final state f. A *successful computation* is a word $w = a_1 \ldots a_n$ with $a_i \in \mathcal{A}$ for $1 \leq i \leq n$, such that there is a sequence of transitions $(i, a_1, q_1), (q_1, a_2, q_2), \ldots, (q_{n-1}, a_n, f)$ connecting the initial state i to a final state f. A state q is *complete* if for any $a \in \mathcal{A}$, there is at least one transition $(q, a, q') \in T$ for some state $q' \in Q$. The automaton is *complete*[1] if each state $q \in Q$ is complete. A state q is *deterministic* if for any $a \in \mathcal{A}$, there is at most one transition $(q, a, q') \in T$ for some state $q' \in Q$. The automaton is *deterministic* if each state $q \in Q$ is deterministic. An automaton $\mathtt{A}$ is deterministic and complete if and only if the transition relation T is the graph of a function $\delta : Q \times \mathcal{A} \to Q$, the *transition function* of $\mathtt{A}$. Two automata $\mathtt{A}$ and $\mathtt{A}'$ over the same alphabet are *isomorphic* if there is a bijective map sending each state of $\mathtt{A}$ to a state of $\mathtt{A}'$ which sends bijectively the set of transitions of $\mathtt{A}$ to the set of transitions of $\mathtt{A}'$. The language $L(\mathtt{A})$ *computed* (or *recognized*) by an automaton $\mathtt{A}$ is the set of all successful computations of $\mathtt{A}$. Two isomorphic automata compute the same language, but a language can be computed by two non-isomorphic automata.

Any automaton $\mathtt{A}$ gives rise to a directed graph: the set of vertices is the set of states, and the set of directed edges is defined by setting a directed edge from vertex q to vertex q' if and only if there is some letter $a \in \mathcal{A}$ such that $(q, a, q') \in T$. The graph may have multiple edges. This is sometimes referred to as a multigraph in the literature. Throughout this paper, graph will mean multigraph. If we wish to emphasize that a graph has no multiple edge, we shall say that the graph is *simple*. A directed graph is *strongly connected* if given any pair (u, v) of vertices, there is a directed path joining u to v.

Since any directed graph induces an undirected graph by forgetting the orientation of the edges, any automaton $\mathtt{A}$ also gives rise to an undirected graph. Let $k \geq 1$. A *cycle of length* k in $\mathtt{A}$ is a closed walk of length k in the underlying undirected graph, considered up to circular permutation. Note that a cycle may or may not respect the orientation of the original transitions. We say that the cycle *respects the direction* of (the underlying directed graph of) $\mathtt{A}$ if each oriented edge of the cycle respects the orientation of the original transition in $\mathtt{A}$. A cycle of length 1 is also called a *loop* (or a *self-loop*, for emphasis). A cycle is *simple* if it is represented by a closed walk in which no edge is used more than once. For instance, a closed walk in which one edge is traveled twice in opposite directions does not induce a simple cycle. The *genus* of a closed oriented surface Σ is defined as half the rank of the homology group $H_1(\Sigma)$. It is equal to the maximal number of nonintersecting simple closed curves $C_1, \ldots, C_g$ such that $\Sigma - (C_1 \cup \cdots \cup C_g)$ remains connected.

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press



Given a graph G, it is easy to find a surface Σ such that G embeds in Σ: consider a generic immersion of G into the 2-sphere S^2 with only double points ("crossings of two distinct edges"); for each double point, create a handle $S^1 \times I$ which contains exactly one edge. Once there are no more double points, one obtains an embedding of G into a new surface Σ. The process of creating a handle increases the genus of the surface. In particular, the surface thus created, into which G embeds, may not have minimal genus. The genus of a graph G is the minimal genus among all closed oriented surfaces into which G embeds. An embedding into a surface of minimal genus will be called *minimal*. Any minimal embedding is *cellular*: the complement of the embedded graph in the surface is a finite disjoint union of cells (topological two-dimensional discs). Such a cell is called a *face* of the embedding. The boundary ∂c of a cell c is defined as usual as in simplicial or singular homology. Given a cellular embedding, a closed walk w induces a one-dimensional complex, namely the union of embedded edges defined by the closed walk. We say that a closed walk w bounds a cell c if the one-dimensional complex it induces bounds c. A closed walk bounding a cell is necessarily a cycle. The *length* of a cell is a well-defined positive integer: it is intuitively the length of a bounding cycle. For details, see Gross and Tucker (2001) or Bonfante and Deloup (2018, Section 4.2).

The *genus* $g(\mathtt{A})$ of an automaton $\mathtt{A}$ is defined as the genus of the underlying undirected graph (see, e.g., Gross and Tucker, 2001, Section 1.4.6).

2.3 The genus of a regular language

We start with the definition of the genus of a regular language, introduced in Bonfante and Deloup (2018). The basic idea is to consider among all finite automata computing a given regular language those that have the smallest genus (as graphs). As to why we need to restrict to deterministic automata in the definition below, see Bonfante and Deloup (2018, Section 8, Theorem 11).

Definition 2. *Let L be a regular language. Let $DFA(L)$ be the set of all DFA computing L. The* genus *$g(L)$ is*

$$g(L) = \min\{g(\mathtt{A}) \mid \mathtt{A} \in \mathrm{DFA}(L)\}.$$

A regular language is said to be planar *(resp.* toric*) if its genus is zero (resp. one).*

In other words, the genus of a regular language is the minimal genus among all genera of closed oriented surfaces into which a finite deterministic automaton recognizing the language embeds. There are many nonplanar languages. A hierarchy of languages of strictly increasing genus is explicitly constructed in Bonfante and Deloup (2018). We shall produce other examples of hierarchies in Section 5 (see also Remark 6).

Remark 1. Taking the minimum over all *complete and accessible* DFA does not change the genus of the language. See Bonfante and Deloup (2018, Proposition 3 & 4).

2.4 Genus and genus size

Given a regular language L, we let $\mathtt{A}_{\min}(L) = \mathtt{A}_{\min}$ be the minimal deterministic automaton associated with L. The *size* $|L|_{\text{set}}$ of the language L is the size of the minimal deterministic automaton $\mathtt{A}_{\min}$:

$$|L|_{\text{set}} = |\mathtt{A}_{\min}|.$$

Definition 3. *We define the* genus size *of L to be*

$$|L|_{\text{gen}} = \min\{|A| \mid L(A) = L,\ g(A) = g(L)\}$$

where the minimum is taken over all DFA recognizing L of minimal genus.

By definition $|L|_{\text{gen}} \geq |L|_{\text{set}}$ with equality if and only if the minimal automaton realizes the genus of L. From Bonfante and Deloup (2018, Section 5) we know that the genus size is in general reached by several non-isomorphic deterministic automata. In light of the previous examples, a number of natural questions arise. What is the trade-off between size and genus? Can a regular language be planar and its minimal automaton have an arbitrary high genus? Indeed, the following result shows that the genus size of L can grow at least exponentially in terms of (minimal automaton) size of L:

Theorem 1. *There is a family of planar regular languages $(L_n)_{n\in\mathbb{N}}$ and a positive number $K > 1$ such that*

$$|L_n|_{\text{gen}} = O(K^{|L_n|_{\text{set}}}).$$

The construction consists in building a sequence of planar languages L_n having increasingly high genus minimal automata $\mathtt{A}_{\min}(L_n)$. The language L_n will be finite, so there will be a spanning tree for L_n, ensuring planarity, while the high genus of the minimal automaton is produced by means of a cascade of n directed $K_{5,5}$'s, completed by one initial state and one single final state.

Proof of Theorem 1. On the alphabet $\mathbb{Z}/5\mathbb{Z}$, given $n \geq 0$, let us consider the automaton $A_n = (Q_n, i_n, F_n, \delta_n)$ defined as follows. The set of states is $Q_n = \mathbb{Z}/5\mathbb{Z} \times \{0, \cdots, n\} \cup \{p_0, \top, \bot\}$. The initial state is p_0, there is a unique final state $\top$. For all $a, b \in \mathbb{Z}/5\mathbb{Z}$, let $\delta_n(p_0, a) = (a, 0)$, $\delta_n((a, n), a) = \top$, if $a \neq b$, $\delta_n((a, n), b) = \bot$ and for $j < n$, $\delta_n((a, j), b) = (a + b, j + 1)$. A typical computation path is

$$p_0 \xrightarrow{a_0} (a_0, 0) \xrightarrow{a_1} (a_0 + a_1, 1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} (x = a_0 + \cdots + a_n, n) \xrightarrow{x} \top.$$

The corresponding language is $L_n = \{a_0 \cdots a_{n+1} \mid \sum_{i=0,n} a_i = a_{n+1}\}$.

It is straightforward that all states of A_n are accessible. Given a state (a, j), consider the language $L_{(a,j)}$ of suffixes (that consists of words sending (a, j) to the final state $\top$). For $j = n$, $L_{(a,n)} = \{a\}$; for $j = n - 1$, $L_{(a,n-1)} = \{a_1 a_2 \mid a_2 = a + a_1\}$; more generally, for $0 \leq j \leq n - 1$, $L_{(a,j)} = \{a_1 a_2 \cdots a_{n-j+1} \mid a + \sum_{i=0}^{n-j} a_i = a_{n-j+1}\}$: hence $L_{(a,j)} \neq L_{(b,k)}$ if $(a, j) \neq (b, k)$, so the states are pairwise nonequivalent and A_n is minimal. The language L_n is finite, thus planar. Indeed, one may span the complete tree of depth $n + 2$ to describe the language which has thus topological size smaller than 5^{n+2}. Let us suppose that $B_n = (R_n, j_n, G_n, \eta_n)$ is a minimal planar automaton recognizing L_n. The set R_n of states of B_n can be viewed as a finite subset of $Q \times \mathbb{N}$. Hence we can suppose that the states have the shape (s, t) with $s \in Q_n$ and $t \in T_s$, that is, $\pi : (s, t) \mapsto s$ defines the projection on the minimal automaton.

We qualify states of the shape (a, j, t) with $j < n$ to be internal states. For any internal state $s = (a, j, t)$, the transition function $\eta_n(s, \cdot) : \mathbb{Z}/5\mathbb{Z} \to R_n$ is injective, because $\delta_n = \pi \circ \eta_n$ is injective. Explicitly, for any $b \neq c \in \mathbb{Z}/5\mathbb{Z}$, we have $\eta_n(s, b) \neq \eta_n(s, c)$.

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press

Let G_n be the underlying graph of B_n. Since G_n is planar, we regard henceforth G_n as an embedded graph in the plane. Given $j \in \{0, \ldots, n-1\}$, let S_j be the subgraph of G_n where any vertices outside $\mathbb{Z}/5\mathbb{Z} \times \{j, j+1\} \times T$ have been removed with their incoming and outcoming edges. Being a subgraph of G_n, the graph S_j is planar. We denote K (respectively M) the set of states of B_n of the shape (a, j, t) (resp. $(a, j+1, t)$) and $k = |K|$ (resp. $m = |M|$).

Any state $s \in K$ is internal. We have seen above that $\eta_n(s, \cdot)$ is injective. Thus, there are exactly 5 outgoing edges from state s, each of which pointing to a different state. Two partial conclusions are drawn from this. First, let e be the number of edges in S_j, then $e = 5k$. Second, there are no bigons in S_j: none of the patterns $s \to s' \to s$ or $s \to s' \leftarrow s$ can happen.

Let f be the number of faces in S_j. Euler's formula for planar graphs applied in S_j gives us $k + m + f = 5k + 2$, that we can rewrite:

$$m + f = 4k + 2. \tag{1}$$

Let f_i be the number of i-faces in S_j. Thus, $f = \sum_{i \geq 1} f_i$. Observe that due to the definition of B_n, there are neither simple odd polygons (that is, no $2i+1$-gon for $i \in \mathbb{N}$), nor bigons as justified above. Thus, $f = f_4 + f_6 + \cdots = \sum_{i \geq 2} f_{2i}$. According to the usual counting argument (see, e.g., Bonfante and Deloup 2018, Lemma 7), $2 \times e = 4f_4 + 6f_6 + \cdots = 10k$. In other words, $\frac{5k}{2} = f_4 + \frac{6}{4} f_6 + \cdots \geq f_4 + f_6 + \cdots = f$. By relation (1), we get

$$m = 4k + 2 - f \geq \frac{3k}{2} + 2 \geq \frac{3k}{2} \tag{2}$$

Take $K = 3/2$. Denote by N_j the states in layer j, that is of the shape (a, j, t), and by n_j the cardinal of N_j. By induction on $j \geq 0$, we prove $n_j \geq 5 \times (3/2)^j$ for $j \leq n$. For the base case, observe that there are at least 5 states in each layer (there are 5 in the minimal automaton). The induction step is a direct consequence of the inequality (2). The result follows. ■

3. Genus Estimate

In order to study further the genus of a regular language, we introduce some classes of regular languages that "do not have short cycles."

It will be convenient to introduce the following function, defined on the set of natural numbers greater than or equal to 2. It should be understood as a nonincreasing function of the number of letters of the alphabet.

Definition 4. *Let $m \geq 2$. Set* $\rho(m) = \begin{cases} 3 & \textit{if } m \geq 4; \\ 4 & \textit{if } m = 3; \\ 5 & \textit{if } m = 2. \end{cases}$

Definition 5. *Let $j \geq 1$. A language L is said to have* no simple cycle of length $< j$ *if the underlying undirected graph of the minimal deterministic complete automaton $\mathtt{A}_{\min}$ for L has no simple cycle of length k for all $1 \leq k < j$.*

Recall that the underlying graph of an automaton is not a simple graph in general: it may have multiple edges. For instance, a double edge induces a simple cycle of length 2.

Example 3. The language $Z_5^{1,2}$ has no simple cycle of length < 3. Indeed, the minimal automaton for $Z_5^{1,2}$ is the one depicted in Figure 1 with all self-loops removed.

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press

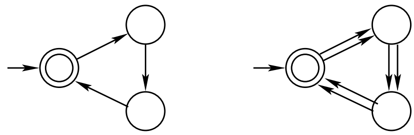


Figure 5. The minimal automata for L_1 and L_2. Note that they have the same underlying simple directed graph.

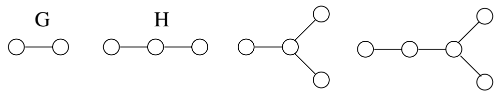


Figure 6. Planar graphs supporting nonsimple bounding closed walks: the complement in the 2-sphere of each of these embedded graphs is an open cell whose boundary is a nonsimple closed walk in the graph.

Remark 2. The role of the alphabet is crucial. The language $L_1 = Z_3^1 = (\{1\}^3)^* = (111)^*$ has no simple cycle of length < 3, while the language $L_2 = (\{1, 2\}^3)^*$ does have simple cycles of length 2. See Figure 5.

The following result is the central tool of this paper.

Theorem 2. Genus estimate. *Let $m \geq 2$. If a regular language L on an m-letter alphabet has no simple cycle of length $< \rho(m)$, then*

$$1 + \frac{(\rho(m) - 2)m - \rho(m)}{2\rho(m)} |L|_{\text{set}} \leq g(L) \leq 1 + \frac{(m-1)}{2} |L|_{\text{set}}. \tag{3}$$

The upper bound is a direct consequence of Euler's formula (see Bonfante and Deloup 2018, Proposition 2). The crucial information consists in the lower bound. Theorem 2 generalizes that of Bonfante and Deloup (2018, Theorem 8). The greater the alphabet is, the weaker the constraint on the required length of cycles is.

Proof of Theorem 2. We need to prove the stated lower bound. Given an integer $k \geq 1$ and a minimal embedding of a graph in a surface Σ, we let f_k denote the number of faces of length k.

Set

$$A(j) = \sum_{k \geq j} \frac{k(m-1) - 2m}{4m} f_k, \quad B(j) = \sum_{k \geq j} k f_k.$$

Then

$$A(j) \geq \left(\frac{m-1}{4m} - \frac{1}{2j} \right) B(j).$$

Let $\mathtt{A}$ be a complete finite deterministic automaton of minimal genus recognizing L, endowed with a minimal embedding of $\mathtt{A}$ into a genus $g(L)$ oriented closed surface Σ. By Bonfante and Deloup (2018, Theorem 5), $g(\mathtt{A}) = 1 + A(1)$. (This accounts for the definition of A.) It is readily verified that $\rho(m)$ is the smallest natural number j such that $\frac{m-1}{4m} - \frac{1}{2j} > 0$. (This accounts for the definition of ρ.) So now set $j = \rho(m)$. By hypothesis, $\mathtt{A}_{\min}$ has no simple cycle of length less or equal to $j - 1$. It follows from Lemma 1 (Section 6) that $\mathtt{A}$ has no simple cycle of length less or equal to $j - 1$. Consider now a face f in Σ.

Claim. If the length of the face is less than or equal to 4, then any cycle c in $\mathtt{A}$ bounding f must be simple.

In general, there are simple graphs supporting nonsimple bounding cycles. See Figure 6 for a few examples; a more sophisticated example of such a closed walk (of length 8) is given in Bonfante and Deloup (2018, end of Section 4.1). We need to rule them out in order to conclude that there is no small k-face for $k \leq j - 1$.

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press

Proof of the claim. Consider a nonsimple bounding closed walk c of length less than or equal to 4. By definition, there must be at least one edge in each graph that is traveled twice. Hence c contains at most three vertices. We can rule out the case where c is the closed walk that consists of the same self-loop traveled four times consecutively: indeed, one self-loop is a simple cycle of length 1, which is prohibited by $\mathtt{A}$ having no short simple cycles. So the only possible *simple* graphs underlying the nonsimple closed walk c are the first two graphs G and H represented on the left side in Fig. 6. Consider now the underlying (multi)graph of a complete 2-letter automaton $\mathtt{A}$ containing G (resp. H). An extremal vertex of G (resp. H) must have at least one extra outgoing edge: it is easy to see that the multigraph has either one self-loop (the outgoing edge is a self-loop itself) or a simple cycle of length 2 (the outgoing edge points to another vertex). Therefore, $\mathtt{A}$ has a simple cycle of length less than or equal to $j-1$, which contradicts the hypothesis. ■

End of the proof. It follows from the claim that the only possible k-faces for small $k \leq 4$ must be bounded by simple cycles. We deduce that there is no k-face for $k \leq j-1$: $f_1 = \cdots = f_{j-1} = 0$. Hence

$$\begin{aligned} g(\mathtt{A}) = 1 + A(j) &\geq 1 + \left(\frac{m-1}{4m} - \frac{1}{2j}\right) B(j) \\ &= 1 + \left(\frac{m-1}{4m} - \frac{1}{2j}\right) B(1) \\ &= 1 + \left(\frac{m-1}{4m} - \frac{1}{2j}\right) 2m|\mathtt{A}| \end{aligned}$$

The last equality is the usual counting argument relating faces and edges (see, e.g., Bonfante and Deloup 2018, Lemma 7). Therefore

$$g(\mathtt{A}) \geq 1 + \frac{(j-2)m - j}{2j}|\mathtt{A}|. \tag{4}$$

Since $|\mathtt{A}| \geq |\mathtt{A}_{\min}| = |L|_{\text{set}}$, we deduce the desired result. ■

Remark 3. The inequality (4) holds for any complete deterministic automaton of minimal genus recognizing L under the hypotheses of Theorem 2. It is in general sharper than the lower bound of the theorem.

Since the lower bound for the genus is strictly greater than 1, we observe the following fact:

Corollary 3.1. *Let $m \geq 2$. If a regular language L on an m-letter alphabet has no simple cycle of length $< \rho(m)$, then $g(L) > 1$.*

Remark 4. In the definition of "having no short cycle" for a language, the condition applies to the underlying graph of the *complete* minimal deterministic finite automaton. Considering complete automata here is crucial. For instance the language $L = (0123)^*$ on the four-letter alphabet $\{0, 1, 2, 3\}$ is represented by the deterministic finite automaton $\mathtt{A}$ whose set of states is $\mathbb{Z}/4\mathbb{Z}$, 0 being the initial and final state, with the transitions $i \xrightarrow{i} i+1$, $i \in \mathbb{Z}/4\mathbb{Z}$. Here $\mathtt{A}$ is minimal and has no simple cycle of length shorter than 4. (Note that $L = w^*$ with $w = 0123$, i.e., L is the homomorphic image of a 1-letter language.) In particular L is planar: $g(L) = 0$. Now the original automaton $\mathtt{A}$ is not complete since there is only one outgoing transition at each state, whereas there are four letters. It is left to the reader to check that if one completes $\mathtt{A}$ in order to obtain the complete minimal deterministic automaton for L, then $\mathtt{A}$ does have cycles of length 2 and 3, respectively. Hence L does have cycles of length 2 and, as a consequence, does not satisfy the hypothesis of Theorem 2.

4. The Computability Conjecture

Theorem 1 shows that the genus size of L can grow at least exponentially in terms of the minimal automaton size of L. Is this the worst possible case? If there exists a computable function that limits the growth of the genus size in terms of the minimal automaton size, the following conjecture would be proved.

Computability Conjecture for the Genus. *The genus $g(L)$ of every regular language L is computable.*

Note that the genus of a graph is computable in the following sense: given a surface Σ, there is a linear time algorithm, such that given any graph G, either finds an embedding of G in Σ or returns a subgraph H of G that is a minimal forbidden minor for embeddability in Σ (see Mohar (1996)). As explained above (Section 2.2), given a graph G, it is easy to find a surface Σ such that G embeds in Σ. The genus g of Σ may not be minimal. We apply the linear time algorithm for each surface of genus $g-1, g-2, \ldots, 1, 0$ until there is no embedding. This yields the genus of G in a finite number of steps.

Although the genus of a graph is computable, the Computability Conjecture for an arbitrary regular *language* is not obvious. Indeed, a regular language L is recognized by an infinite number of deterministic finite automata, and since the genus may be realized by an automaton much larger than the minimal deterministic finite automaton $\mathtt{A}_{\min}$ recognizing L, it is not a priori clear where and when to stop. How much larger? According to Theorem 1, we may need to go after an automaton whose size is at least exponential in the size of L. In order to prove the conjecture, one needs a priori bounds that depend on the intrinsic complexity (ideally the size) of the language.

We prove a partial case of the conjecture above. First, we state a result of special interest about a particular class of automata.

Theorem 3. *Let $m \geq 2$ and $g \geq 0$.*

(1) *If $g \leq 1$, then any genus g complete deterministic finite automaton has at least one simple cycle of length $< \rho(m)$.*
(2) *If $g \geq 2$, then there is a finite number of genus g complete deterministic finite automaton without simple cycle of length $< \rho(m)$.*

This result is crucial and ensures the computability of the genus for a fairly generic class of regular languages.

Proof of Theorem 3. Let $m \geq 2$. Set $j = \rho(m)$. According to (4) (see Remark 3),

$$1 + \left(\frac{(j-2)m - j}{2j}\right) |\mathtt{A}| \leq g(\mathtt{A}) = g$$

for any genus g complete deterministic finite automaton $\mathtt{A}$ without simple cycles of length $\leq j-1$. Therefore, the set

$$E(g) = \left\{ n \in \mathbb{N} \mid 1 + \left(\frac{(j-2)m - j}{2j} n\right) \leq g \right\}$$

(of possible sizes) is finite. For each size $n \in E(g)$, there is at most a finite number of finite automata of size n. Hence there is at most a finite number of genus g complete DFA of fixed size n and without simple cycles of length $\leq j-1$. This proves the second statement (2). Finally, since $\frac{(j-2)m-j}{2j} > 0$, we see that $g > 1$, i.e., $E(0)$ and $E(1)$ are empty. This proves the first statement (1). ∎

Definition 6. *Let $m \geq 2$. Let $\mathscr{C}(m)$ be the class of regular languages on m letters without simple cycles of length $< \rho(m)$.*

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press

Example 4. The languages $Z_5^{0,1,2}$ on three letters is not in $\mathscr{C}(3)$: its minimal automaton has self-loops (Figure 1). The language $Z_5^{1,2}$ on two letters is not in $\mathscr{C}(2)$ because it has simple cycles of length 3 (Figure 1 with self-loops removed).

Example 5. The language $Z_9^{1,2,3,4}$ on three letters is in $\mathscr{C}(4)$: it has no simple cycle of length < 3.

We now state our main result. It is really a corollary of Theorem 3; we state it as a theorem for emphasis.

Theorem 4. *Let $m \geq 2$ and $L \in \mathscr{C}(m)$. Then*

(1) *$g(L) > 1$ and there is a finite number of genus $g(L)$ complete deterministic finite automata $\mathtt{A}$ computing L.*
(2) *The genus size $|L|_{\text{gen}}$ and the genus $g(L)$ are computable.*

Proof of Theorem 4. The first statement (1) is a direct corollary of Theorem 3. As to the second statement, consider the set F of complete deterministic finite automata $\mathtt{A}$ computing L such that

$$1 + \left(\frac{(j-2)m - j}{2j} \right) |\mathtt{A}| \leq g(\mathtt{A}) \leq g(\mathtt{A}_{\min}).$$

Since the minimal automaton $\mathtt{A}_{\min}$ for L has no simple cycle of length $< \rho(m)$, any complete finite deterministic automaton $\mathtt{A}$ computing L will have the same property (Lemma 1) and therefore satisfies the left inequality above. Hence F is the set of all complete deterministic finite automata computing L and having genus smaller than or equal to the genus of the complete minimal automaton for L. The set F is finite by Theorem 3 and contains all complete deterministic finite automata computing L with minimal genus $g(L)$. Furthermore, the set F is computable: for each size $|\mathtt{A}|$, we can decide whether there exists a complete deterministic finite automaton computing L of size $|\mathtt{A}|$, construct each of them if they exist, and compute the genus of each of them (for the computation of the genus of a graph, see Mohar (1996)). The second statement (2) follows. ■

Corollary 4.1. *The planarity of a regular language $L \in \mathscr{C}(m)$ for $m \geq 2$ is decidable.*

As a corollary, we obtain that there is a finite number of regular languages of fixed genus without simple short cycles.

Corollary 4.2. *Let $m \geq 2$. For any $L \in \mathscr{C}(m)$, $g(L) \geq 2$. Furthermore, for each $g \geq 2$, there is a finite number of regular languages $L \in \mathscr{C}(m)$ such that $g(L) = g$.*

A few comments may be useful. The hypotheses about the absence of small short cycles and the fixed size of the alphabet are essential. For instance, let $n, p \geq 3$ and consider the language on two letters

$$L_{n,p} = \{w \in \{0, 1\}^* \mid |w|_0 = 0 \bmod n, \ |w|_1 = 0 \bmod p\}$$

(where $|w|_a$ denotes the number of occurrences of letter a in the word w) which can be regarded as the shuffle of Z_n^1 and Z_p^1 (Sakarovitch, 2009, p. 65). The minimal automaton for $L_{n,p}$ is obtained as the shuffle product of the minimal automata of Z_n^1 and Z_p^1, respectively. This automaton computing $L_{n,p}$ clearly embeds into the torus. See Figure 7.

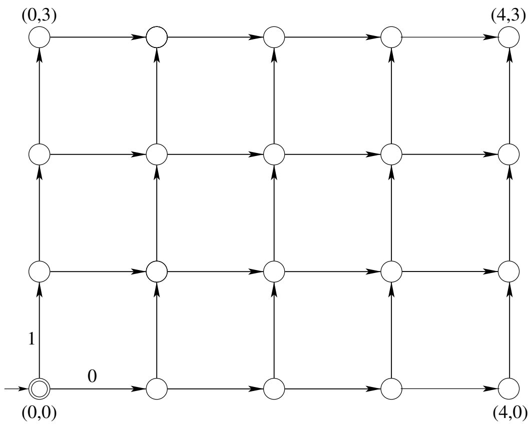


Figure 7. The minimal automaton for $L_{4,3}$ and its embedding in the torus. The states $(k, 0)$ and $(k, 3)$ $(0 \leq k \leq 4)$ and the states $(0, l)$ and $(4, l)$ $(0 \leq l \leq 3)$ are to be identified as well as the corresponding transitions (the resulting automaton having exactly $12 = 4 \times 3$ states and $24 = 2 \times 12$ transitions) so that the picture represents an embedding of the minimal automaton in the torus.

Proposition 1. For $n, p \geq 4$, $L_{n,p}$ is toric.

It follows from Proposition 1 that by contrast to Corollary 4.2, there is an *infinite* family of toric languages on a two-letter alphabet. Since $g(L_{n,p}) = 1$ and the lower bound of Theorem 2 is always greater than 1, $L_{n,p}$ must have short simple cycles. Indeed, for any n, p, the minimal automaton has simple cycles of length 4, so $L_{n,p} \notin \mathscr{C}(2)$.

The simplest example in the series of toric languages $L_{n,p}$ is $n = p = 4$ and has 16 states. (Note that $|L_{n,p}|_{\text{set}} = |L_{n,p}|_{\text{gen}}$.) Compare with Book and Chandra (1976) where a two-letter nonplanar language with 35 states is constructed.

Proof of Proposition 1. Since the minimal automaton $\mathtt{A}_{\min}$ of $L_{n,p}$ embeds in a torus, $g(L_{n,p}) \leq 1$. We have to prove that $L_{n,p}$ is nonplanar, i.e., $g(L_{n,p}) \geq 1$, for $n, p \geq 4$. Let $\mathtt{A}$ be a complete finite deterministic automaton for $L_{n,p}$. The canonical epimorphism $\mathtt{A} \to \mathtt{A}_{\min}$ (note that $\mathtt{A}_{\min}$ is a complete automaton) induces a graph epimorphism $\pi : \mathscr{G}(\mathtt{A}) \to \mathscr{G}(\mathtt{A}_{\min})$. Since $n, p \geq 4$, $\mathscr{G}(\mathtt{A}_{\min})$ has no simple cycle of length ≤ 3. Applying Lemma 1, we see that neither has $\mathscr{G}(\mathtt{A})$. Consider now a minimal embedding of $\mathtt{A}$ into some closed oriented surface Σ.

Claim. Each face of the embedding has length at least 4.

Proof of the claim. Suppose the contrary. There is a face f of length ≤ 3. The boundary of f must be a nonsimple cycle $c = \partial f$. Since there is no self-loop in $\mathtt{A}$, c has length 2 and consists in exactly one edge e with immediate backtracking. It follows that e is monofacial. Let $\vec{e}$ be the original oriented edge in $\mathtt{A}$. Since e is monofacial, one of the endpoints, $s(\vec{e})$ or $t(\vec{e})$, has total degree 1, which is a contradiction since the alphabet has two letters.

For $j \geq 1$, let f_j be the number of faces of length j. By the claim above, $f_1 = f_2 = f_3 = 0$. Therefore, according to the genus formula (Bonfante and Deloup, 2018, Theorem 5),

$$g(L) = g(\mathtt{A}) = 1 + \sum_{j \geq 1} \frac{j-4}{8} f_j = 1 + \sum_{j \geq 4} \frac{j-4}{8} f_j \geq 1.$$

This is the desired result. ∎

Remark 5. Inspection of the proof shows that if L is a two-letter language that does not have any simple cycle of length ≤ 3, then $g(L) \geq 1$.

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press

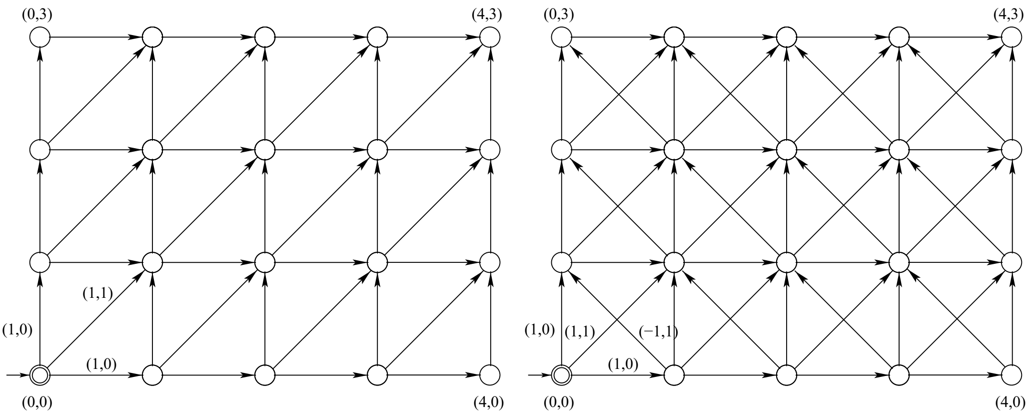


Figure 8. The minimal automata for $Z_{4,3}^{(1,0),(0,1),(1,1)}$ and $Z_{4,3}^{(1,0),(0,1),(1,1),(-1,1)}$ respectively (with the same identification convention as in Figure 7). The first one embeds into the torus, the second one (nor any deterministic automaton equivalent to it) does not.

Remark 6. Let $2 \leq n_1 \leq n_2 \leq \cdots \leq n_r$ be integers. For any finite sequence $(w_1, \ldots, w_s) \in (\mathbb{Z}/n_1\mathbb{Z} \times \cdots \mathbb{Z}/n_r\mathbb{Z})^s$, let

$$Z_{n_1,\ldots,n_r}^{w_1,\ldots,w_s} = \left\{ a_1 \ldots a_k \in \{w_1, \ldots, w_s\}^* \mid \sum_{i=1}^{k} a_i = 0 \in \mathbb{Z}/n_1\mathbb{Z} \times \cdots \mathbb{Z}/n_r\mathbb{Z} \right\}.$$

This is a generalization of Definition 1 where $r = 1$. The language $L_{n,p}$ considered above is also a particular case with $r = 2$: $L_{n,p} = Z_{n,p}^{(0,1),(1,0)}$. Observe that $Z_{n,p}^{(0,1),(1,0),(1,1)}$ is again a toric language, this time with three letters, and the minimal automaton has simple cycles of length 3, so it does not belong to $\mathscr{C}(3)$. However, the language $Z_{n,p}^{(0,1),(1,0),(1,1),(-1,1)}$ has four letters and no simple cycles of length ≤ 2, so by Theorem 2, its genus is bounded below by $1 + \frac{1}{6}np$. See Figure 8. This provides another example of hierarchy based on the genus.

Remark 7. Genus versus syntactic monoid. By the previous remark, the genus distinguishes between the languages $Z_{n,p}^{(1,0),(0,1),(1,1)}$ and $Z_{n,p}^{(1,0),(0,1),(1,1),(-1,1)}$:

$$g\left(Z_{n,p}^{(1,0),(0,1),(1,1)}\right) = 1, \; g\left(Z_{n,p}^{(1,0),(0,1),(1,1),(-1,1)}\right) > 1.$$

However, the syntactic monoid does not distinguish between the languages $Z_{n,p}^{(0,1),(1,0),(1,1)}$ and $Z_{n,p}^{(0,1),(1,0),(1,1),(-1,1)}$, since

$$\mathfrak{M}\left(Z_{n,p}^{(0,1),(1,0),(1,1)}\right) = \mathfrak{M}\left(Z_{n,p}^{(0,1),(1,0),(1,1),(-1,1)}\right) = \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z}.$$

(To see this, let $C(w)$ denote the context of a word $w = w_1 \cdots w_r$ where each letter w_i lies in $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z}$. Observe that $C(w) = C_L(\sum_i w_i)$.)

Another observation is that given a language $L \in \mathscr{C}(m)$, it is easy to build an infinite number of languages of the same genus $g(L)$ with short simple cycles. For instance, if A denotes the alphabet of L and has at least two letters, then for any $k \geq 0$, $g(A^k \cdot L) = g(L)$.

A systematic study of graph transformations on the minimal automaton that preserve the genus of the language will be undertaken in a forthcoming paper. Here we remark that given a genus-minimal automaton for L, an automaton of the same genus can be built for the composition $A \cdot L$: it is easily seen to have one simple cycle of length 2, see Figure 9 (for a two-letter language).

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press

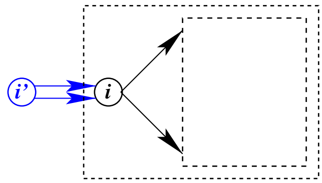


Figure 9. A genus-minimal automaton for L with initial state i; the genus-minimal automaton for the language $A \cdot L$ with initial state i'.

Note that in general, this add-on operation changes nontrivially the syntactic monoid. For instance, for any given alphabet A, $\mathfrak{M}(A^k \cdot A^*) = \mathbb{Z}/k\mathbb{Z}$. In particular, the syntactic monoid $\mathfrak{M}(A^*)$ of the free monoid A^* is trivial. Hence for $k \geq 1$, $\mathfrak{M}(A^k \cdot A^*) \neq \mathfrak{M}(A^*)$ whereas $g(A^k \cdot A^*) = g(A^*)$.

5. Genus-based hierarchies

Examples of nonplanar regular languages actually abound. The difficulty lies in classifying them by their genus. A number of examples were given in Book and Chandra (1976) and Bonfante and Deloup (2018). Here we show new examples of genus-based hierarchies. The first example yields a closed formula for the genus.

Theorem 5. *Let $\lceil \cdot \rceil$ denote the ceiling function on the real numbers, which maps x to the least integer $\lceil x \rceil$ that is equal to or greater than x. Let $k \geq 4$. The language $Z_{2k+1}^{1,2,\ldots,k}$ has genus $\lceil \frac{(2k-2)(2k-3)}{12} \rceil$. In particular, $g(Z_{2k+1}^{1,2,\ldots,k}) \underset{k \to +\infty}{\to} +\infty$.*

This result is remarkable in that it yields an explicit, closed formula for the genus. In general, the computation of the genus is nontrivial, as explained in the previous section.

Proof of Theorem 5. The language $Z_{2k+1}^{1,2,\ldots,k}$ is computed by the following automaton, denoted $A = A_{2k+1}^{1,2,\ldots,k}$. The set of states is $Q = \mathbb{Z}/(2k+1)\mathbb{Z}$, with initial and final state 0. The transitions are given by the rule $i \overset{j}{\to} i+j$ for $i \in Q$ and $j \in \{1, 2, \ldots, k\} \subset \mathbb{Z}/(2k+1)\mathbb{Z}$. It follows from the definition that A is complete. Consider a state $q \in \mathbb{Z}/(2k+1)\mathbb{Z}$. Its language L_q of suffixes (words that send the state q to the final state 0) consists of words $w = a_1 \cdots a_n$ in the alphabet $\{1, \ldots, k\}$ such that $\sum_{i=1}^n a_i = q \bmod 2k+1$. The languages L_q, $q \in Q$, are all distinct since the word 1^q is contained only in L_q. Hence states are inequivalent and A is minimal. The underlying unoriented multigraph is the complete graph K_{2k+1}. We verify two properties:

- K_{2k+1} has no self-loop (clear from the definition of the transitions) and has no simple cycle of length 2 (for a cycle $i \overset{j}{\to} i+j \overset{k}{\to} i$ would imply $j + k = 0 \bmod 2k+1$, hence either j or k is not in $\{1, 2, \ldots, k\}$). Therefore, the minimal length of a simple cycle is 3.
- The cardinality of the alphabet is $k \geq 4$.

According to Theorem 2 (see also (Bonfante and Deloup, 2018, Theorem 8)), $g(Z_{2k+1}^{1,2,\ldots,k}) \geq 1 + \frac{(k-3)(2k+1)}{6}$. To prove that this lower bound for the genus is actually an equality, we notice that the genus of the minimal automaton provides an upper bound. So

$$1 + \frac{(k-3)(2k+1)}{6} \leq g\big(Z_{2k+1}^{1,2,\ldots,k}\big) \leq g(A) = g(K_{2k+1}) = \left\lceil \frac{(2k-2)(2k-3)}{12} \right\rceil.$$

The last equality is the exact formula for the genus of the complete graph on $2k+1$ vertices (Ringel and Youngs, 1968). Since the genus is a natural number, one can take the ceiling function of the lower bound. It remains to observe that the ceiling function of the lower bound is exactly the upper bound. This is the desired result. ■

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press

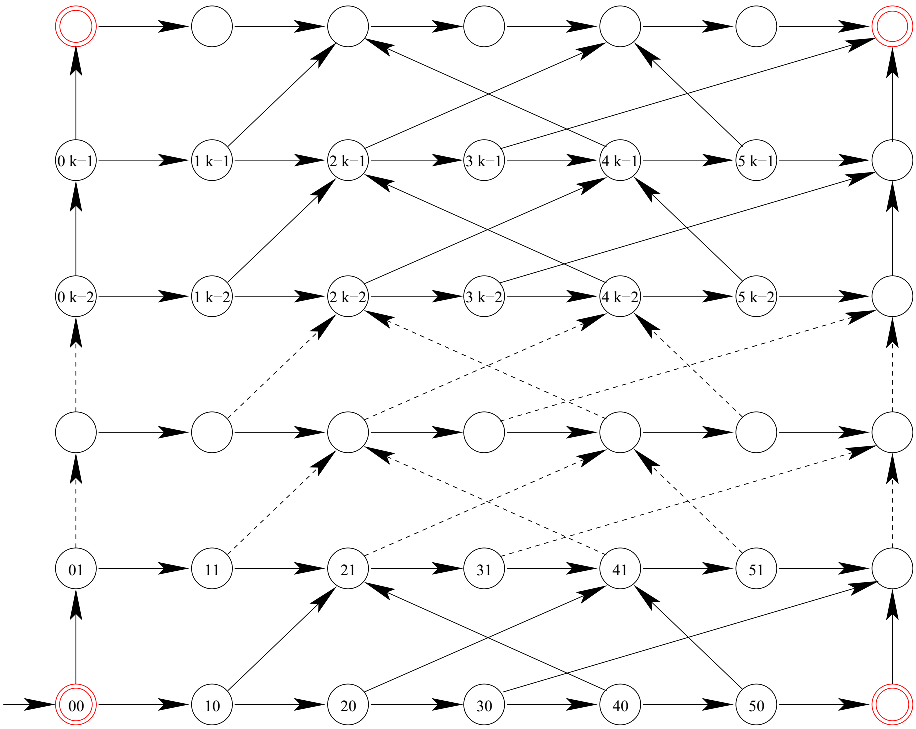


Figure 10. The automaton $\mathtt{A}_k$ drawn (with crossings) on a torus: the states $(j, 0)$ and (j, k) for $0 \le j \le 6$ (resp. the states $(0, l)$ and $(6, l)$ for $0 \le l \le 6$) are to be identified, as well as the corresponding transitions.

Examples of genus-based hierarchies are provided in Bonfante and Deloup (2018) with a fixed 4-letter (or more) alphabet. This left out regular languages on an alphabet with fewer letters, namely 2 or 3 letters. (Regular languages on a 1-letter alphabet are easily seen to be planar. See, e.g., Bonfante and Deloup (2018).) R. V. Book and A. K. Chandra have built a regular language on two letters that is nonplanar (Book and Chandra, 1976). We shall prove here the following result which is constructive and explicit; it also implies the existence of a genus hierarchy of regular languages on any m-letter alphabet for $m \ge 2$.

Theorem 6. *There is a genus hierarchy of regular languages on only 2-letters: for any nonnegative integer $n \ge 0$, there exists a regular language L on a 2-letter alphabet such that $g(L) \ge n$.*

Proof of Theorem 6. Let $A = \mathbb{Z}/2\mathbb{Z}$ be the alphabet. For $k \ge 5$, consider the finite deterministic automaton $\mathtt{A}_k$ defined as follows. The set of states is $Q_k = \mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/k\mathbb{Z}$. The transitions are

$$(i, j) \xrightarrow{0} (i + 1, j), \quad (i, j) \xrightarrow{1} (2i, j + 1).$$

Pick the state $(0, 0)$ as the initial and unique final state. See Figure 10 for a picture of the automaton $\mathtt{A}_k$.

It is easily seen that $\mathtt{A}_k$ is deterministic and complete. Given a state (i, j), denote by $L_{(i,j)}$ the languages of suffixes from state (i, j) (that is, words sending the state (i, j) to the final state). Observe that $L_{(i,j)}$ contains the word 0^r1^s if and only if $i + r = 0 \bmod 6$ and $j + s = 0 \bmod k$. This shows that the states are pairwise nonequivalent. Hence the automaton $\mathtt{A}_k$ is also minimal. It is readily verified that $\mathtt{A}_k$ has no simple cycle of length less than or equal to 4. Therefore, Theorem 2 applies: the language L_k recognized by $\mathtt{A}_k$ has genus $1 + \frac{3k}{20} \le g(L_k)$. This implies the desired result. ■

https://doi.org/10.1017/S0960129519000057 Published online by Cambridge University Press

Remark 8. Theorem 2 actually implies $1 + \frac{3k}{20} \leq g(L_k) \leq 1 + 3k$. However, the exact computation of the genus of L_k is unknown to the authors.

6. The cycle property

The main result of this section is that the absence of short cycles in the minimal automaton is a property of the language (Corollary 6.1).

Lemma 1. *Let $k \geq 1$. Assume that the underlying graph G of a minimal automaton of a language L has no simple cycle of length less than or equal to k. Then neither has the underlying graph $\tilde{G}$ of any automaton recognizing L.*

Proof. The canonical epimorphism from the automaton to the minimal deterministic automaton induces a graph epimorphism $\tilde{G} \to G$. Suppose that $\tilde{G}$ has a simple cycle c' of length $l \leq k$. Its image in G is a closed path c of length $l' \leq l$. The closed path c admits a decomposition into a product of cycles, each of which has length less than or equal to $l' \leq l \leq k$. At least one of these cycles is simple in G. □

Corollary 6.1. *The property for the minimal complete deterministic automaton* $\mathtt{A}$ *to have no simple cycle of length l for all $l \leq k$ is a property of the language* $L(\mathtt{A})$.

Acknowledgements. The authors wish to thank the referees. The manuscript benefited greatly from their remarks. This work was partially supported by the CIMI (Centre International de Mathématiques et d'Informatique), ANR-11-LABX-0040-CIMI within the program ANR-11-IDEX-0002-02 and by ANR-14-CE25-0005-3, ELICA.

Notes

1 The term "complete" has a different meaning in graph theory: a graph is complete if every pair of distinct vertices is connected by an edge. Hopefully this should not cause any confusion in this paper.

References

Book, R. V. and Chandra, A. K. (1976). Inherently nonplanar automata. *Acta Informatica* **6** (1) 89–94.

Bonfante, G. and Deloup, F. (2018). The genus of regular languages. Mathematical Structures in Computer Science **18** (1) 14–44.

Bezáková, I. and Pál, M. (1999). Planar finite automata. Technical report, Student Science Conference, Comenius University.

Eilenberg, S. (1974). *Automata, Languages and Machines*, vol. A. Academic Press, New York.

Gross, J. and Tucker, T. W. (2001). *Topological Graph Theory,* Dover Publications.

Mohar, B. (1996). Embedding graphs in an arbitrary surface in linear time. In: *Proc. 28th Ann. ACM STOC*, Philadelphia, ACM Press, 392–397.

Ringel, G. and Youngs, J. W. T. (1968). Solution of the Heawood map-coloring problem. *Proc. Nat. Acad. Sci. U.S.A.* **60** 438–445.

Sakarovitch, J. (2009). *Elements of Automata Theory*. Cambridge University Press, Cambridge.

Cite this article: Bonfante G and Deloup FL (2019). Decidability of regular language genus computation. *Mathematical Structures in Computer Science* **29**, 1428–1443. https://doi.org/10.1017/S0960129519000057