Regular languages as local functions with small alphabets ⋆

Stefano Crespi Reghizzi and Pierluigi San Pietro

Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)
Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano I-20133
stefano.crespireghizzi@polimi.it pierluigi.sanpietro@polimi.it

Abstract. We extend the classical characterization (a.k.a. Medvedev theorem) of any regular language as the homomorphic image of a local language over an alphabet of cardinality depending on the size of the language recognizer. We allow strictly locally testable (slt) languages of degree greater than two, and instead of a homomorphism, we use a rational function of the local type. By encoding the automaton computations using comma-free codes, we prove that any regular language is the image computed by a length-preserving local function, which is defined on an alphabet that extends the terminal alphabet by just one additional letter. A binary alphabet suffices if the local function is not required to preserve the input length, or if the regular language has polynomial density. If, instead of a local function, a local relation is allowed, a binary input alphabet suffices for any regular language. From this, a new simpler proof is obtained of the already known extension of Medvedev theorem stating that any regular language is the homomorphic image of an slt language over an alphabet of double size.

1 Introduction

The family of regular languages has different characterizations using regular expressions, logical formulas or finite automata (FA). In the latter approach the more abstract formulation, often named Medvedev theorem [6, 8], uses a local language (i.e., a *strictly locally testable* (slt) [5] language of testability degree $k = 2$) and a letter-to-letter homomorphism: every regular language $R \subseteq \Sigma^*$ is the homomorphic image of a local language, called the *source*, over another alphabet Λ; the *alphabetic ratio* $\frac{|\Lambda|}{|\Sigma|}$ is in the order of the square of the number of FA states. Continuing a previous investigation [3] motivated by the attractive properties of slt encoding, we address the following question: how small can the source alphabet, or, better, the alphabetic ratio, be? We recall the answer provided by the generalized Medvedev theorem [3]: any regular language is the homomorphic image of a k-slt language over an alphabet Λ of cardinality $2|\Sigma|$ – but in general not less – where k is in the order of the logarithm of the FA size. Thus the minimal alphabetic ratio is independent from the FA size.

The present study concerns new possibilities of reducing the source alphabet size, while generalizing Medvedev theorem in a different direction: the homomorphism is replaced by a rational function [1] (also known as transduction) of the local type [9]. Loosely speaking, a *local function* defines a mapping from a source language $L \subseteq \Lambda^*$ to a target language $R \subseteq \Sigma^*$ by means of a partial local mapping from words

⋆ Work partially supported by CNR - IEIIT.

of fixed length $k \geq 1$, over Λ, to letters of Σ: parameter k is called the degree of locality of the function. To the best of our knowledge, the approach to regular language characterization using local rational functions instead of homomorphisms, has never been considered before, in this context.

Since a homomorphism is a function of locality degree one, the main question we address is whether every regular language is the image of a local function, defined on a source alphabet of cardinality smaller that $2|\Sigma|$; the latter, as said, is the minimum needed for a characterization using homomorphism.

Exploiting the properties of comma-free codes [2] to encode the computations of an FA, we obtain a series of results. First, the main question above bifurcates depending on the local function being length-preserving or not. If the local function is allowed to be length-decreasing, we show that every regular language is the target of a local function defined on a binary alphabet. Second, assuming that the local function preserves the input length up to a fixed constant value, we prove that the source alphabet of size $|\Sigma| + 1$ suffices to characterize any regular language using a local function. Moreover, for the subfamily of regular languages having polynomial density, we show that a binary source alphabet permits to define every language using a local length-preserving function.

In a further generalization, the second part of the paper moves from a local function to a *local relations*, i.e., a set of pairs of source and target words. Again, we assume the relation to be length-preserving, and we prove that the source alphabet can be taken to be binary, independently of the complexity of the target regular language. At last, the latter results permits to obtain a new, simpler proof of the already mentioned homomorphic characterization theorem in [3].

It is noteworthy that although the theorems differ with respect to their use of local functions/relations and on the length-preserving feature, all the proofs have a common structure and rely on a formal property of comma-free codes when they are mapped by a morphism and a local function/relation. Stating such property as a preliminary lemma permitted considerable saving in the following proofs.

Altogether, a rather complete picture results about the minimum alphabet size needed to characterize regular languages by means of local functions (including homomorphism as special case) and relations.

Paper organization. Sect. 2 lists the basic definitions for slt languages and rational local functions/relations; it also includes the definition and the number of comma-free codes, and states and proves the preliminary lemma mentioned above. Sect. 3 defines the local function that encode the labelled paths of an FA, proves the results for length-decreasing and then for length-preserving functions, and finishes with the case of languages having polynomial density. Sect. 4 presents the characterization of regular languages based on local relations, and the new proof of the homomorphic characterization result in [3]. Sect. 5 summarizes the main results.

2 Preliminaries

For brevity, we omit the basic classical definitions for language and automata theory and just list our notations. The empty word is denoted ε. The Greek upper-case letters

$\Gamma, \Delta, \Theta, \Lambda$ and Σ denote finite terminal alphabets. For clarity, when the alphabet elements are more complex than single letters, e.g., when a finite set of words is used as alphabet, we may also embrace the alphabet name and its elements between "$\langle$" and "$\rangle$". For a word x, $|x|$ denotes the length of x. The i-th letter of x is $x(i)$, $1 \leq i \leq |x|$, i.e., $x = x(1)x(2)\dots x(|x|)$. For any alphabet, $\Sigma^{\leq k}$ stands for $\bigcup_{1\leq i\leq k} \Sigma^i$. Let $\#$ be a new character not present in the alphabets, to be used as word *delimiter* to shorten some definitions, but not to be counted as true input symbol.
A *homomorphism* $\xi : \Lambda^* \to \Sigma^*$ is called *letter-to-letter* if for every $b \in \Lambda$, $\xi(b)$ is in Σ.
A *finite automaton* (FA) A is defined by a 5-tuple $(\Sigma, Q, \to, I, F)$ where Q is the set of states, $\to$ the state-transition relation (or graph) $\to \subseteq Q \times \Sigma \times Q$; I and F are resp. the subsets of Q comprising the initial and final states. If $(q, a, q') \in \to$, we write $q \xrightarrow{a} q'$. The transitive closure of $\to$ is defined as usual, e.g., we also write $q \xrightarrow{x} q'$ with $x \in \Sigma^+$ with obvious meaning, and call it a *path*, with an abuse of language (for a nondeterministic FA, $q \xrightarrow{x} q'$ may actually correspond to more than one path in the transition graph). We denote the *label* x of the path $\alpha = q \xrightarrow{x} q'$ by *lab*(α). The starting and ending states are resp. denoted by *in*$(\alpha) = q$ and *out*$(\alpha) = q'$. If $q \in I$ and $q' \in F$, the path is called *accepting*.

Strictly locally testable language family There are different equivalent definitions of the family of strictly locally testable (*slt*) languages [5, 4]; without loss of generality, the following definition is based on bordered words and disregards for simplicity a finite number of short words that may be present in the language.
The following short notation is useful: given an alphabet Λ and for all $k \geq 2$, let

$$\Lambda^k_\# = \#\Lambda^{k-1} \cup \Lambda^k \cup \Lambda^{k-1}\#.$$

For all words x, $|x| \geq k$, let $F_k(x) \subseteq \Lambda^k_\#$ be the *set of factors of length* k present in $\#x\#$. The definition of F_k is extended to languages as usual.

Definition 1 (Strict local testability). *A language $L \subseteq \Lambda^*$ is k-strictly locally testable (k-slt), if there exist a set $M_k \subseteq \Lambda^k_\#$ such that, for every word $x \in \Lambda^*$, x is in L if, and only if, $F_k(x) \subseteq M_k$. Then, we write $L = L(M_k)$. A language is slt if it is k-slt for some value k, which is called the testability degree. A forbidden factor of M_k is a word in $\Lambda^k_\# - M_k$.*

The degree $k = 2$ yields the family of *local* languages. The k-slt languages form an infinite hierarchy under inclusion, ordered by k.

Local relations and functions Let Λ and Σ be finite alphabets, called the source and target alphabet, respectively. A *rational relation* (also called a transduction) [1, 9, 8] over Λ and Σ is a rational (i.e., regular) subset $r \subseteq \Lambda^+ \times \Sigma^*$. The image of a word $x \in \Lambda^+$ is the set of words $y \in \Sigma^*$ such that $(x, y) \in r$. The *source* and *target* languages of a rational relation are respectively defined as $\{x \in \Lambda^+ \mid \exists y \in \Sigma^* : (x, y) \in r\}$ and as $\{y \in \Sigma^+ \mid \exists x \in \Lambda^+ : (x, y) \in r\}$.
A rational relation r is *length-preserving* if, for all pair of related words, the length of the words differ by at most a constant value, i.e., there exists $m \geq 0$ such that for all $(x, y) \in r$, *abs*$(|x| - |y|) \leq m$.

Let r be a rational relation such that, for all $x \in \Lambda^*$, $|\{y \in \Sigma^+ \mid (x,y) \in r\}| \leq 1$. Then the mapping $f : \Lambda^* \to \Sigma^*$ defined by $f(x) = y$ is a (partial) *function*.
Next, we focus on the rational relations/functions called *local*[1] [9], where there exists $k > 0$ such that the image of each word $x \in \Lambda^+$ only depends on its factors of length k; such factors may be visualized as the contents of window of width k that slides frome left to right on the source word. More precisely, for every word $w \in \Lambda^* \cup \#\Lambda^* \cup \Lambda^*\# \cup \#\Lambda^*\#$, with $|w| \geq k$, we define the *scan* [9], denoted by $\Phi_k(w)$, as the sequence:

$$\Phi_k(w) = \langle w(1) \dots w(k) \rangle, \langle w(2) \dots w(k+1) \rangle, \dots, \langle w(|w|-k+1) \dots w(|w|) \rangle.$$

Clearly, a scan $\Phi_k(w)$ can be viewed as a word over the "alphabet" $\Lambda^k_\#$, that we denote $\langle \Lambda^k_\# \rangle$ to prevent confusion. Such alphabet comprises all k-tuples in $\#\Lambda^{k-1} \cup \Lambda^k \cup \Lambda^{k-1}\#$. For instance, $\Phi_3(\#abbab\#)$ is the word $\langle \#ab \rangle \langle abb \rangle \langle bba \rangle \langle bab \rangle \langle ab\# \rangle$.

Definition 2 (local function / relation). *A (partial) function $f : \Lambda^* \to \Sigma^*$ is local of degree k, $k \geq 1$, if there exist a finite set $T \subseteq \langle \Lambda^k_\# \rangle$, and a homomorphism $\nu : T^* \to \Sigma^*$, called associated, such that $f(x) = \nu(\Phi_k(\#x\#))$.*
A local relation $r \subseteq \Lambda^* \times \Sigma^*$ of degree k is similarly defined, using a finite substitution $\sigma : T^* \to 2^{\Sigma^*}$ instead of a homomorphism, as: $r = \{(x, \sigma(\Phi_k(\#x\#)))\}$.
A function (a relation) is called local if it is local of degree k for some $k \geq 1$.

It is obvious that the source language of a local function/relation is a k-slt language, defined by the finite set T of factors.

Comma-free codes A finite set $X \subset \Lambda^+$ is a *code* [2] if every word in Λ^+ has at most one factorization in words (also known as *codewords*) of X, more precisely: for any $u_1 u_2 \dots u_m$ and $v_1 v_2 \dots v_n$ in X, where the u and v are codewords, the identity $u_1 u_2 \dots u_m = v_1 v_2 \dots v_n$ holds only if $m = n$ and $u_i = v_i$ for $1 \leq i \leq n$. We use a code X to represent a finite alphabet Γ by means of a one-to-one homomorphism, denoted by $[\![\ ]\!]_X : \Gamma^+ \to \Lambda^+$, called *encoding*, such that $[\![\alpha]\!]_X \in X$ for every $\alpha \in \Gamma$.
Let $n \geq 1$. A set $X \subset \Lambda^n$ is a *comma-free* code, if, intuitively, no codeword overlaps the concatenation of two codewords: more precisely, for any $t, u, v, w \in \Lambda^*$, if tu, uv, vw are in X, then $u = w = \varepsilon$, or $t = v = \varepsilon$.

Number of words of comma-free code We need the following result (see [7] and its references) on the number of codewords in a comma-free code of length k over an alphabet with cardinality $|\Lambda| = n$. Let $\ell_k(n) = \frac{1}{k} \sum \mu(d) n^{k/d}$, where the summation is extended over all divisors d of k, and μ is the Möbius function defined by

$$\mu(d) = \begin{cases} 1 \text{ if } d = 1 \\ 0 \text{ if } d \text{ has any square factor} \\ (-1)^r \text{ if } d = p_1\, p_2 \dots p_r \text{ where } p_1\, p_2 \dots p_r \text{ are distinct primes.} \end{cases}$$

Proposition 1. *For every alphabet with n letters and for every odd integer $k \geq 1$ there is a comma-free code of length k with $\ell_k(n)$ words.*

[1] Unfortunately, the adjective "'local"', for slt languages means of testability degree two, whereas for the locality degree of functions, it means any integer value.

The definition of the Möbius function μ is such that if k is a prime number the summation in the formula is just equal to $n^k - n$, i.e., for k prime:

$$\ell_k(n) = \frac{n^k - n}{k}. \tag{1}$$

Comma-free codes and local functions/relations The next lemma will be repeatedly invoked in later proofs.

Lemma 1. *Let Λ, Γ and Σ be finite alphabets and $X \subset \Lambda^k$ be a comma-free code of length k, for some $k > 1$, such that $|X| = |\Gamma|$. Let $L \subseteq \Gamma^+$ be the 2-slt language $L(M_2)$ defined by a set $M_2 \subset \Gamma^2_\#$.*

1. *The encoding of L by means of code X, i.e., the language $[\![L]\!]_X$, is a $2k$-slt language included in $(\Lambda^k)^*$.*
2. *Given a homomorphism $\pi : \Gamma^* \to \Sigma^*$, the language $\pi(L)$ is the target language of a local function $f : \Lambda^* \to \Sigma^*$ of degree $2k$, having $[\![L]\!]_X$ as source language.*
3. *Given a finite substitution $\sigma : \Gamma^* \to 2^{\Sigma^*}$, the language $\sigma(L)$ is the target of a local relation $r \subseteq \Lambda^* \times \Sigma^*$ of degree $2k$, having $[\![L]\!]_X$ as source language.*

Proof. We first claim that XX^+ is a $(2k)$-slt language. Let $F_{2k}(XX^+)$ be the set of the factors of length $2k$ of $\#XX^+\#$, hence it is obvious that $XX^+ \subseteq L(F_{2k}(XX^+))$. We prove the converse inclusion by contradiction. Let $z \in \Lambda^+$ be such that $F_2(z) \subseteq F_{2k}(XX^+)$ but $z \notin XX^+$. Since every word in $F_{2k}(XX^+)$ must have a code $x \in X$ has a factor, then for z not to be in $L(F_{2k}(XX^+))$, the set $F_{2k}(z)$ must include a word of the form $xy \in \Lambda^{2k}$, with $x \in X$, $y \notin X, y \in \Lambda^k$, or a word of the form $xy\#$, with $x \in X, y \in \Lambda^{<n}$. We only consider the former case, since the latter is analogous. Since $xy \in F_{2k}(XX^+)$, there is a word $p \in XX^+$ including xy as a factor of length $2k$.
Since $p \in XX^+$, p must be of the form $X^*txy\Lambda^+$, with $t \neq \varepsilon$ (otherwise $y \in X$), $|t| < k$, and there exist $u, v \in \Lambda^+$ such that $uv = x \in X$ and $tu \in X$; therefore, p has the form X^*tuvwX^*, with w being a non empty prefix of y and such that also $vw \in X$. By definition of comma-free code, since the three words tu, uv and vw are in X, either $t = v = \varepsilon$, or $u = w = \varepsilon$, a contradiction with the assumption that all those words are in Λ^+.
We need a few definitions. Let $\Psi_{2k} \subset \Lambda^{2k}_\#$ be the set of forbidden factors of M_2 when encoded with X, i.e., the set:

$$\begin{aligned}\Psi_{2k} = \{\#[\![\alpha]\!]_X \Lambda^{k-1} \mid \alpha \in \Gamma, \#\alpha \notin M_2\} \cup \{[\![\alpha\beta]\!]_X \mid \alpha, \beta \in \Gamma, \alpha\beta \notin M_2\} \cup \\ \{\Lambda^{k-1}[\![\beta]\!]_X\# \mid \beta \in \Gamma, \beta\# \notin M_2\}.\end{aligned} \tag{2}$$

To define language $[\![L]\!]_X$ we use the following set which avoids the forbidden factors:

$$M_{2k} = F_{2k}(XX^+) - \Psi_{2k}. \tag{3}$$

Clearly, the inclusion $L(M_{2k}) \subseteq X^+$ holds since $M_{2k} \subseteq F_{2k}(XX^+)$.

Part (1). We claim that $L(M_{2k})$ is exactly the language $[\![L]\!]_X$, i.e., for any $x \in \Lambda^+$, $x \in [\![L]\!]_X$ if, and only if, $F_{2k}([\![x]\!]_X) \subseteq M_{2k}$.

We prove $L(M_{2k}) \subseteq [\![L]\!]_X$. Let $x \in L(M_{2k})$ therefore $F_{2k}(x) \subseteq M_{2k}$ and, by contradiction, let $x \notin [\![L]\!]_X$. Since $x \in XX^+$ and $[\![L]\!]_X \subseteq X^+$, x must contain a factor of one of the forbidden forms (2) in Ψ_{2k}, a contradiction.
We prove $[\![L]\!]_X \subseteq L(M_{2k})$. Let $x \in [\![L]\!]_X$; it is enough to show that $F_{2k}(x) \subseteq M_{2k}$. By contradiction, assume that there is $w \in F_{2k}(x)$, with $w \notin M_{2k}$. Since $w \in F_{2k}(XX^+)$, it must be $w \notin \Psi_{2k}$. Therefore, w can only be of the form $y[\![\alpha]\!]_X z$, with $yz \in \Lambda^k$, for some $\alpha \in \Gamma$, with both $y, z \neq \varepsilon$, otherwise x could not be the comma-free encoding of a word of L while having a factor not in Ψ_{2k}. However, since $x \in X^+$, there exist β, γ such that $w' = [\![\gamma]\!]_X [\![\alpha]\!]_X [\![\beta]\!]_X$ is a factor of x, with y a suffix of γ and z a prefix of β. If at least one of $[\![\gamma]\!]_X [\![\alpha]\!]_X$, $[\![\alpha]\!]_X [\![\beta]\!]_X$ is in Ψ_{2k}, then $x \notin [\![L]\!]_X$, a contradiction. If both $[\![\gamma]\!]_X [\![\alpha]\!]_X$, $[\![\alpha]\!]_X [\![\beta]\!]_X \notin \Psi_{2k}$, then by definition of $F_{2k}(XX^+)$ it is necessary that $y[\![\alpha]\!]_X z \in M_{2k}$, also a contradiction.

Part (2). Define a homomorphism $\nu' : \langle \Lambda_{\#}^{2k} \rangle^* \to \Sigma^*$ for every $z \in \langle \Lambda_{\#}^{2k} \rangle$, by means of the following cases, for all $u \in \Lambda^+$:

$$\begin{array}{l} H_1 : \textit{if } z \textit{ has the form } \langle \#u \rangle, \textit{ let } \nu'(z) = \varepsilon \\ H_2 : \textit{if } z \textit{ has the form } \langle [\![\alpha]\!]_X \, [\![\beta]\!]_X \rangle, \textit{ for some } \alpha, \beta \in \Gamma, \textit{ let } \nu'(z) = \pi(\alpha) \\ H_3 : \textit{ if } z \textit{ has the form } \langle u \rangle, \textit{ with } u \neq \langle [\![\alpha]\!]_X \, [\![\beta]\!]_X \forall \alpha, \beta \in \Gamma, \textit{ let } \nu'(z) = \varepsilon \\ H_4 : \textit{ if } z \textit{ has the form } \langle u[\![\alpha]\!]_X \# \rangle, \textit{ let } \nu'(z) = \pi(\alpha). \end{array} \tag{4}$$

Loosely speaking, the image is a non-empty word in two cases: H_2, when the "sliding window" contains two codewords, and H_4, when the window ends with a codeword followed by $\#$.
The local function $f' : \Lambda^* \to \Sigma^*$, defined (as in Def. 2) by applying morphism ν' to the scan Φ_{2k}, is total, since it is defined for every $\gamma \in \Lambda^{2k}\Lambda^*$. This is useful in the following proof.
If, as usual, we consider M_{2k} as an alphabet, denoted as $\langle M_{2k} \rangle$, we can define a homomorphism $\nu : \langle M_{2k} \rangle^* \to \Sigma^*$, as $\nu(z) = \nu'(z)$ for every $z \in \langle M_{2k} \rangle \subseteq \langle \Lambda_{\#}^{2k} \rangle$. Let $f : \Lambda^* \to \Sigma^*$ be the local function defined as $f(x) = \nu(\Phi_{2k}(\#x\#))$, for all $x \in L(M_{2k})$, which is thus defined only over $L(M_{2k})$. We claim that the target language of f is $\pi(L)$.

i) We first prove that $\pi(L) \subseteq f(\Lambda^*)$. The proof is by induction on the length $n \geq 2$ of words in L (ignoring shorter words as usual). Precisely, the induction hypothesis is:

$$\text{if } z \in \Gamma^+ \text{ has length } n \geq 2, \text{ then } \nu'(\Phi_{2k}(\#[\![z]\!]_X)) = \pi(z).$$

From this the thesis follows immediately: if $y \in \pi(L)$, then $y = \pi(z)$ for some $z \in L \subseteq \Gamma^+$; obviously, $F_{2k}([\![z]\!]_X) \subseteq M_{2k}$ so f' is defined. Since f, f' have the same value where they are both defined, it follows that $f([\![z]\!]_X) = f'([\![z]\!]_X) = \pi(z)$.
Base case: if $|z| = 2$, then $z = \alpha\beta$, for $\alpha, \beta \in \Gamma$. By definition, the set M_2 contains $\#\alpha, \alpha\beta, \beta\#$. Thus, the set $F_{2k}([\![\alpha\beta]\!]_X) \subseteq M_{2k}$ comprises three words: $t_1 = \#[\![\alpha]\!]_X v$, $t_2 = [\![\alpha\beta]\!]_X$, $t_3 = u[\![\beta]\!]_X \#$, for suitable $u, v \in \Lambda^{k-1}$. By Eq. (4), $\nu'(t_1) = \varepsilon$, $\nu'(t_2) = \pi(\alpha)$, $\nu'(t_3) = \pi(\beta)$. Since $\Phi_{2k}(x) = t_1 t_2 t_3$, we have $\nu'(t_1 t_2 t_3) = \pi(\alpha)\pi(\beta)$.
Inductive step: assume now $|z| > 2$ and that the induction hypothesis holds for every word $z' \in \Gamma^+$ with $|z'| < |z|$. Word z can be factored into $\delta\alpha\beta\gamma$, where $\delta \in \Gamma^*$, $\alpha, \beta, \gamma \in \Gamma$. Let $z' = \delta\alpha\beta$: by induction hypothesis, $\nu'(\Phi_{2k}(\#[\![z']\!]_X)) = \pi(z') = \pi(\delta\alpha\beta)$. Let u be the suffix of length $k-1$ of β: Then, $\nu'(\Phi_{2k}(\#[\![\delta\alpha\beta\gamma]\!]_X \#)) =$

$\nu'(\Phi_{2k}(\#[\![\delta\alpha\beta]\!]_X)) \cdot \nu'(\Phi_{2k}(u[\![\gamma]\!]_X\#)) = \pi(\delta\alpha\beta) \cdot \nu'(\Phi_{2k}(u[\![\gamma]\!]_X\#))$.
By definition of ν' (case H_4), $\nu'(\Phi_{2k}(u[\![\gamma]\!]_X\#)) = \pi(\gamma)$, hence the thesis follows.

ii) We now show that $f(\Lambda^*) \subseteq \pi(L)$. It is enough to prove by induction on $n \geq 2k$ that

$$\text{for every } x \in XX^+ \subseteq \Lambda^+ \text{ of length } n, \text{ there exists } z \in \Gamma^+ \text{ s.t. } f'(x) = \pi(z). \quad (5)$$

In fact, to prove (ii), it suffices to notice that if $y \in f(\Lambda^*)$, then there is $x \in X^+$ such that $y = f(x)$ is defined (i.e., $F_{2k}(x) \subseteq M_{2k}$): since by (5) $f'(x) = \pi(z)$, we have $f(x) = f'(x) = \pi(z)$ (functions f and f' are the same where f is defined).
We prove the base case $n = 2k$ of (5). Let $x = [\![\alpha]\!]_X[\![\beta]\!]_X$ for $\alpha, \beta \in \Gamma$. As in the proof of Part (1), $F_{2k}(x) \subseteq M_{2k}$ is composed of three words: $t_1 = \#[\![\alpha]\!]_X u$, $t_2 = [\![\alpha\beta]\!]_X$, $t_3 = v[\![\beta]\!]_X\#$, for suitable $u, v \in \Lambda^{k-1}$. Since ν' is total, $f'(x) = \nu'(\Phi_{2k}(\#x\#)$ is by definition $\nu'(t_1)\nu'(t_2)\nu'(t_3) = \pi(\alpha)\pi(\beta) = \pi(\alpha\beta)$.
The inductive case is also trivial. Let $x \in XX^+, |x| = n$, with the induction hypothesis holding for words of length less than n. Word x can be factored into $x'x''$, with $x' \in X^+, x'' \in X$. By induction hypothesis, there exists $z' \in \Gamma^+$ such that $f'(x') = z'$. The proof is then analogous to the base case.

Part (3). (Sketch) We notice that for every $z \in \Gamma$, the substitution $\sigma(z)$ is a finite set of words over Σ, and we let m be the length of the longest word in $\sigma(\Gamma)$. We can thus define a finite alphabet $\langle\Theta\rangle$, whose elements are the subsets in $2^{(\Sigma^{\leq m})}$, and a new finite substitution $\tau : \langle\Theta\rangle^* \to 2^{(\Sigma^{\leq m})}$, associating every symbol in $\langle\Theta\rangle$ with its corresponding set of words. We define the homomorphism $\pi : \Gamma^* \to \langle\Theta\rangle^*$, as $\forall z \in \Gamma, \pi(z) = \langle\sigma(z)\rangle$.
Then, the substitution σ can be defined as the composition of substitution τ with homomorphism π, i.e., $\sigma(L) = \tau(\pi(L))$.
By Part (2), there is a local function $f : \Lambda^* \to \Theta^*$ such that its target language is equal to $\pi(L)$. It is then clear that $\tau(f(L))$ is a local relation. □

Example 1. We first illustrate Def. 2. Let $\Lambda = \{a, b\}$ and $\Sigma = \{0, 1\}$. We define a local function $f : \{a, b\}^* \to \{0, 1\}^*$ of degree 4. Let the set $T \subseteq \Lambda^4_\#$ be $T = F_4(\{aab, bab\}^+)$. To finish, let the associated homomorphism $\nu : T^* \to \Sigma^*$ be:

$$\begin{cases} \nu(\#aab) = 0,\ \nu(baab) = 0,\ \nu(bbab) = 1, \\ \text{for all other } z \in F_4(\{aab, bab\}^+) :\ \nu(z) = \varepsilon. \end{cases}$$

Notice that ν is undefined for all other words in $\Lambda^4_\#$, such as $\#a^3$ and $abab$. The target language of f is $0\{0, 1\}^*$; we show how to compute a value of f:

$$\begin{aligned} f(aab\,bab) &= \nu\left(\Phi_4(\#aab\,bab\#)\right) \\ &= \nu(\langle\#aab\rangle)\,\nu(\langle aabb\rangle)\,\nu(\langle abba\rangle)\,\nu(\langle bbab\rangle)\,\nu(\langle bab\#\rangle) \\ &= 0\varepsilon\varepsilon1\varepsilon = 01. \end{aligned}$$

Observe that $X = \{aab, bab\}$ is a comma-free code of length 3, therefore $\{aab, bab\}^+$ is a 6-slt language, although in this particular case is also 4-slt. If we encode 0 and 1 resp. with the codewords aab and bab, then the function f can be defined as follows: $f\left([\![z]\!]_X\right) = \begin{cases} z, \textit{ if } z \in 0(0 \cup 1)^* \\ \perp, \textit{ otherwise} \end{cases}$. Clearly function f is not length-preserving, because of the definition of ν.
To illustrate Part 1 of Lemma 1, observe that $L = 0(0 \cup 1)^*$ is 2-slt, with $L = L(M_2)$ and $M_2 = \{\#0, 00, 01, 10, 11, 0\#, 1\#\}$. Since the code length is 3, the language

$[\![L]\!]_X$ is 6-slt; its defining set M_6 has the form of Eq. (3); we just list some factors: $M_6 = \{\#\mathbf{aab}aa, \#\mathbf{aab}ba, \mathbf{aabaab}, \ldots, ab\mathbf{aab}a, \ldots ab\mathbf{aab}\#\}$ where codewords are evidenced in bold.

3 Characterization of regular languages by local functions

By the extended Medvedev theorem [3] (reproduced below in Th. 5), every regular language over Σ is the homomorphic image of an slt source language over an alphabet Λ, where $|\Lambda| = 2|\Sigma|$, and a smaller alphabet does not suffice in general. Instead of a homomorphism, we study the use of a local function (of degree greater than one) such that its target language is exactly the regular language to be defined. Then, the main question is how small the source alphabet can be. The first answer (Th. 1) is that a binary source alphabet suffices if the local function is not required to be length-preserving. Second, Th. 2 says that for a local length-preserving function, a source alphabet containing just one more letter than the target alphabet suffices. Then, a specialized result (Th. 3) for regular languages of polynomial density, says that a length-preserving local function over a binary source alphabet suffices, irrespectively of the size of Σ.

Theorem 1. *For every regular language $R \subseteq \Sigma^*$, there exist a binary alphabet Δ and a local function $f : \Delta^* \to \Sigma^*$, such that the target language of f is R.*

Proof. Let $A = (\Sigma, Q, \to, I, F)$ be an FA recognizing R and let $\Gamma = \to$ be the set comprising the edges of A; let $m = |\Gamma|$. Choose a prime k such that in Eq. (1) $\ell_k(2) \geq m$: this is always possible since $\ell_k(2) = \frac{2^k-2}{k}$. Therefore, there exists a comma-free code $Z \subset \Delta^k$ such that $|Z| = m$, and $[\![q \xrightarrow{a} q']\!]_Z$ is the codeword for $\langle q \xrightarrow{a} q' \rangle$.
Define (as in the classical proof of Medvedev theorem) the 2-slt language $L = L(M_2) \subseteq \Gamma^+$, where $M_2 \subseteq \langle \Gamma^2_\# \rangle$ is the set:

$$M_2 = \left\{ \#\langle q \xrightarrow{a} q' \rangle \mid q \in I, a \in \Sigma, q' \in Q \right\} \cup \left\{ \langle q \xrightarrow{a} q' \rangle \langle q' \xrightarrow{b} q'' \rangle \mid a, b \in \Sigma, q, q', q'' \in Q \right\} \cup \left\{ \langle q \xrightarrow{a} q' \rangle \# \mid q \in Q, a \in \Sigma, q' \in F \right\}.$$

Define the homomorphism $\pi : \Gamma^* \to \Sigma^*$ by means of $\pi(\langle q \xrightarrow{a} q' \rangle) = a$. It is obvious that $\pi(L) = R$. From Lemma 1, Part 2), we have that $\pi(L)$ is the target language of a local function of degree $2k$. □

In general, the local function of Th. 1 is not length-preserving. A length-preserving function may require a source alphabet size depending on the target alphabet size. We prove that a source alphabet barely larger than the target one is sufficient, also improving on the alphabetic ratio of the generalized Medvedev theorem [3].

Theorem 2. *For every regular language $R \subseteq \Sigma^*$, there exist an alphabet Λ of size $|\Sigma| + 1$ and a length-preserving local function $f : \Lambda^* \to \Sigma^*$ such that the target language of f is R.*

We need some definitions and intermediate properties to prove the thesis. First, we define certain sets of paths of bounded length in the graph of the FA A that recognizes the language $R \subseteq \Sigma^*$.

Definition 3 (Bounded paths). *Let $A = (\Sigma, Q, \delta, I, F)$ and let $k \geq 1$. For $\sim \in \{<, \leq, =\}$, let $\Sigma^{\sim k}$ be the set of words in Σ^+ of length, respectively, less than, less or equal to, or equal to k. We define the following sets:*

$$P_{\sim k} = \{q \xrightarrow{y} q' \mid q, q' \in Q, y \in \Sigma^{\sim k}\},\ P_{\sim k,F} = \{q \xrightarrow{y} q_F \mid q \in Q, q' \in F, y \in \Sigma^{\sim k}\}.$$

We view the sets $P_{\sim k}, P_{\sim k,F}$ as finite alphabets, to be respectively written as $\langle P_{\sim k}\rangle$ and $\langle P_{\sim k,F}\rangle$. The language of the accepting paths of automaton A, of length $\geq k$, is denoted by $\mathcal{P}_k \subseteq \langle P_{=k}\rangle^+\langle P_{\leq k,F}\rangle$.

Of course, $P_{<k} \subseteq P_{\leq k}$, $P_{=k} \subseteq P_{\leq k}$ and $P_{\sim k,F} \subseteq P_{\sim k}$.
The following statement is obvious.

Lemma 2. *The language of the accepting paths of an FA A, $\mathcal{P}_k \subseteq \langle P_{=k}\rangle^+\langle P_{\leq k,F}\rangle$, is the 2-slt language $\mathcal{P}_k = L(M_2)$ defined by the following set:*

$$\begin{aligned} M_2 = &\{\#\alpha \mid \alpha \in \langle P_{=k}\rangle, in(\alpha) \in I\} \cup \\ &\{\alpha\alpha' \mid \alpha \in \langle P_{=k}\rangle, \alpha' \in \langle P_{=k}\rangle \cup \langle P_{\leq k,F}\rangle, out(\alpha) = in(\alpha')\} \cup \\ &\{\alpha\# \mid \alpha \in \langle P_{\leq k,F}\rangle\}. \end{aligned} \tag{6}$$

Next, we define the homomorphism

$$\pi : (\langle P_{=k}\rangle \cup \langle P_{\leq k,F}\rangle)^* \to \Sigma^* \text{ as: } \pi(\alpha) = lab(\alpha). \tag{7}$$

It is obvious that $\pi(\mathcal{P}_k) = L(A) \cap \Sigma^{\geq k}$.
Now, we encode every path in $P_{\leq k}$ with a comma-free code X of the same length k.

Proposition 2. *There exist $k > 0$, an alphabet Λ of cardinality $|\Sigma| + 1$ and a comma-free code $X \subset \Lambda^k$ such that $|P_{\leq k}| = |X|$.*

Proof. The set $P_{\leq k}$ can be viewed as a subset of $Q \times (\cup_{1\leq i\leq k}\Sigma^i) \times Q$. By posing $n = |\Sigma|$, it follows that $|P_{\leq k}| \leq |Q|^2 \sum_{1\leq i\leq k} n^i \leq |Q|^2 n^{k+1}$. By Eq. (1), if k is prime then $\ell_k(n+1) = \frac{(n+1)^k - n - 1}{k}$. To have $\ell_k(n+1) \geq |P_{\leq k}|$, we need to choose k so that $|Q|^2 n^{k+1} \leq \frac{(n+1)^k - n - 1}{k}$, i.e., $|Q|^2 k n^{k+1} + n + 1 \leq (n+1)^k$. For fixed n and fixed Q , the inequality holds for all sufficiently large k. □

Thus, each path $\alpha \in P_{\leq k}$ is encoded by a word $[\![\alpha]\!]_X$ of X and the following inequality holds, to be used to prove the length-preserving property of the local function:

$$\forall \beta \in P_{\leq k} : |lab(\beta)| \leq |[\![\beta]\!]_X| \leq |lab(\beta)| + k - 1. \tag{8}$$

Proof of Theorem 2 To finish the proof, we apply Lemma 1, Part 2) with the following correspondence between mathematical entities:

- The alphabet Γ is the set of paths $P_{\leq k}$ of Def. 3
- The code X is the one defined in Prop. 2
- The language $L \subseteq \Gamma^+$ is $\mathcal{P}_k$ of Lemma 2
- The homomorphism π is defined in Eq. (7).

Hence a local function f of degree $2k$ exists, length-preserving by inequality (8). □

The case of polynomial density languages Here we focus on the family of regular languages that have polynomial density. The *density function* [10] of a language $R \subseteq \Sigma^*$ counts the number of words of length n in R and is defined as $\rho_R(n) = |R \cap \Sigma^n|$. Language R has *polynomial density* if $\rho_R(n) = \mathcal{O}(n^k)$ for some integer $k \geq 0$. Clearly, a language R has polynomial density if, and only if, a deterministic trim FA that recognizes R is such that, for any states $q, q' \in Q$, the number of distinct paths of length n from q to q' is polynomial. We prove that if a regular language has polynomial density, then in Th. 2 a binary source alphabet suffices.

Theorem 3. *Let $R \subseteq \Sigma^*$ be regular language of polynomial density. There is a binary alphabet Δ and a length-preserving local function $f : \Delta^* \to \Sigma^*$ such that $f(\Delta^*) = R$.*

Proof. The number of words of length h is $\mathcal{O}(h^{m-1})$, where m is the number of states of a deterministic FA recognizing R. By letting in Eq. (1) (Prop. 1) $n = |\Lambda| = 2$ and choosing a prime value for k, we have that $\ell_k(2) = \frac{2^k-2}{k}$, which is $\mathcal{O}(2^k)$, i.e., there is a comma-free code X with $|X|$ being $\mathcal{O}(2^k)$. If the FA is trim, the number of different k-paths is at most polynomial in k, hence for suitably large k it will be smaller than $|X|$. Therefore, the proof of Th. 2 still holds with a binary comma-free code. □

4 Other results

Th. 2 above says that any regular language is the result of a local length-preserving function applied to words over an alphabet containing one more letter. The next theorem positively answers the question whether any improvement over the previous result is possible if the image is defined by means of a local relation instead of a function.

Theorem 4. *For every regular language $R \subseteq \Sigma^*$, there exist a binary alphabet Δ and a length-preserving local relation $r \subseteq \Delta^+ \times \Sigma^+$ such that the target language of r is R.*

Proof. Let $A = (\Sigma, Q, \to, I, F)$ be an FA. Refer to Lemma 1, and assume that $\Lambda = \Delta$, $X \subseteq \Delta^k$ is a comma-free code of length k, and $\Gamma = \{(q, q') \mid q, q' \in Q, \exists \alpha \in P_{\leq k}, q = \mathit{in}(\alpha), q' = \mathit{out}(\alpha)\}$. We can safely assume that k is large enough so that $|X| = |Q|^2$, hence we can define a codeword $[\![(q, q')]\!]_X$ for every pair (q, q') of states of Q. The proof resembles the proof of Th. 2 but, instead of encoding every labelled accepting path, we just encode the two end states of the same path, omitting the path label. Let $\xi : \langle P_{\leq k} \rangle^* \to \Delta^*$ be the homomorphism that erases the label of a path $\alpha \in \langle P_{\leq k} \rangle$, and returns its encoding by X, more precisely: $\xi(\alpha) = [\![\langle in(\alpha), out(\alpha) \rangle]\!]_X$. Define the 2-slt language $L = L(M_2)$ specified by the following set M_2 over the alphabet $\langle \Gamma^2_{\#} \rangle$:

$$\begin{aligned} M_2 = &\{\#\langle q, q' \rangle \mid q \in I, \exists \alpha \in P_{=k}, \langle q, q' \rangle = \xi(\alpha)\} \cup \\ &\{\langle q, q' \rangle \langle q', q'' \rangle \mid \exists \alpha \in P_{=k}, \beta \in P_{\leq k}, \xi(\alpha) = \langle q, q' \rangle, \xi(\beta) = \langle q', q'' \rangle\} \cup \\ &\{\langle q, q' \rangle \# \mid q' \in F, \exists \alpha \in P_{=k}, \langle q, q' \rangle = \xi(\alpha)\} . \end{aligned}$$

We define a finite substitution $\sigma : \langle \Gamma \rangle^* \to 2^{\Sigma^*}$ as follows: $\forall z \in \langle \Gamma \rangle^*$, $\sigma(z) = \mathit{lab}\left(\xi^{-1}\left([\![z]\!]_X\right)\right)$. From Lemma 1, Part 3), we have that $\sigma(L)$ is the target language of a local relation of degree $2k$. □

Characterization of regular languages as homomorphic images of slt languages Our last contribution is a new simpler proof, based on Th. 4, of the known result (Th. 8 of [3]) that every regular language over an alphabet Σ is the homomorphic image of an slt language over an alphabet of size $2|\Sigma|$. The new proof sets a connection between the old result and the preceding theorems. Overall, we obtain a fairly complete picture of the alphabetic ratio needed for computing regular language by means of local functions, local relations, and homomorphic images of slt languages.
It is convenient to introduce a binary operation that merges two strings into one. Given two alphabets Δ, Σ, define the operator $\otimes : \Delta^+ \times \Sigma^+ \to (\Delta \times (\Sigma \cup \varepsilon))^+$ as follows. For every $u \in \Delta^+, v \in \Sigma^+$ such that $j = |u| \geq |v| = k$, let

$$u \otimes v = \langle u(1), v(1)\rangle \ \ldots \ \langle u(k), v(k)\rangle \ \langle u(k+1), \varepsilon\rangle \ \ldots \ \langle u(j), \varepsilon\rangle \, .$$

E.g., if $u = 010001$ and $v = abbab$, then $u \otimes v = \langle 0, a\rangle\langle 1, b\rangle\langle 0, b\rangle\langle 0, a\rangle\langle 0, b\rangle\langle 1, \varepsilon\rangle$. The operator can be extended to languages over the two alphabets as usual. We also need the projections, resp. denoted by $[\,]_\Delta$ and $[\,]_\Sigma$ onto the alphabets Δ and Σ, defined as: $[u \otimes v]_\Delta = u$, $[u \otimes v]_\Sigma = v$.

Proposition 3. *If $X \subset \Delta^k$ is a comma-free code of length $k > 1$, then every subset Z of $X \otimes \Sigma^{\leq k}$ is also a comma-free code of length k.*

Proof. By contradiction, assume that a word $w \in Z^+$ can be factored as $w = uzv$ and as $w = uu'z'v'$, where $|u'| < k$ and both $z, z' \in Z$, i.e., z, z' do overlap in w. By definition, $z = x \otimes y$ and $z' = x' \otimes y'$, for $x, x' \in X, y, y' \in \Sigma \leq k$; therefore $[z]_\Delta$ and $[z']_\Delta$ are codewords of X, but they also overlap in $[w]_\Delta$, the projection of w to Δ, a contradiction of the definition of comma-free code. □

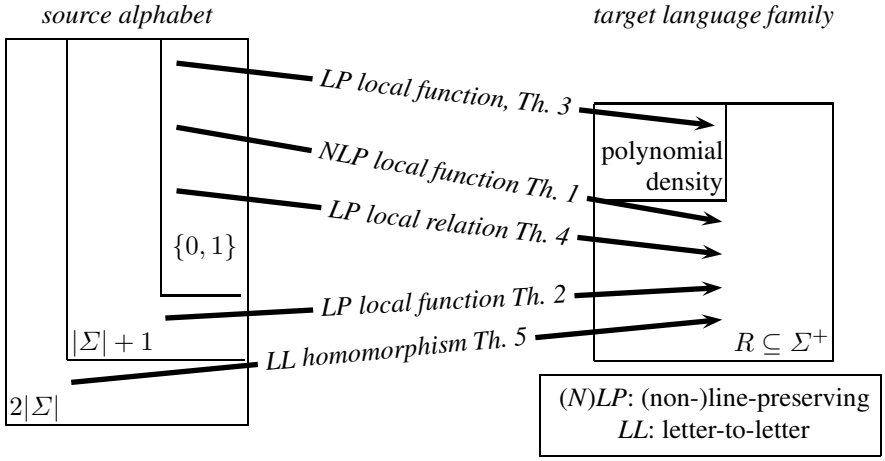
Theorem 5 (part of Theorem 8 of [3]). *For any language $R \subseteq \Sigma^*$, there exists an slt language $L \subseteq \Lambda^*$, where Λ is a finite alphabet of size $|\Lambda| = 2|\Sigma|$, and a letter-to letter homomorphism $\vartheta : \Lambda^* \to \Sigma^*$, such that $R = \vartheta(L)$.*

Proof. For the sake of simplicity, we prove a looser bound, namely $|\Lambda| = 2(|\Sigma| + 1)$. The tighter bound is proved in [3]. Let $\Delta = \{0, 1\}$, the homomorphism $\xi : \langle P_{\leq k}\rangle^* \to \Delta^*$ and the comma-free code $X \subset \Delta^+$ be defined as in the proof of Th. 4. Let $\Lambda = \Delta \times (\Sigma \cup \varepsilon)$. Let $Z \subset \Lambda^k$ be a comma-free code of length k, such that the encoding of each $\alpha \in \langle P_{\leq k}\rangle$ is defined as $[\![\alpha]\!]_Z = \xi(\alpha) \otimes lab(\alpha)$.
Referring to Lemma 1, we consider Γ to be the alphabet $\langle P_{\leq k}\rangle$ and the homomorphism $\pi : \Gamma^* \to \Sigma^*$ to be the projection $\pi(\alpha) = lab(\alpha)$ for every $\alpha \in \langle P_{\leq k}\rangle$. Therefore, there exists a local function $f : \Lambda^* \to \Sigma^*$ whose source language is a $2k$-slt language $L \subseteq \Lambda^*$ and whose target language is R.
Define a letter-to-letter homomorphism $\vartheta : \Lambda^* \to \Sigma^*$ as the projection to the alphabet Σ, i.e., $\vartheta(z) = [z]_\Sigma$ for every $z \in \Lambda$. Let $z \in L, \alpha \in \langle P_{\leq k}\rangle^+$ be such that $z = [\![\alpha]\!]_Z$. It is clear that $\vartheta(z) = lab(\alpha)$, and $f(z) = lab(\alpha)$ as well. Therefore, $R = \vartheta(L)$. □

In comparison, the proof in [3] used an *ad hoc* encoding paying the price of computing its size; moreover, it did not take advantage of the properties in Lemma 1 about comma-free codes, slt languages and local relations, that have permitted to shorten and simplify all the proofs in this paper.

5 Conclusion

We sum up the known results about characterizations of regular languages through local mappings (local function, local relation, homomorphic image of strictly locally testable language) in the following diagram:



We add that the lower limit $2|\Sigma|$ for the case of homomorphism is tight [3]. On the other hand, it is likely but not proved that the $|\Sigma|+1$ limit for length-preserving local functions is tight.

Acknowledgements D. Perrin directed us to comma-free codes. We thank the anonymous referees for their helpful suggestions.

References

1. J. Berstel. Transductions and Context-Free Languages. Teubner, Stuttgart, 1979.
2. J. Berstel, D. Perrin, and C. Reutenauer. Codes and automata. CUP, 2015.
3. S. Crespi Reghizzi and P. San Pietro. From regular to strictly locally testable languages. Int. J. Found. Comput. Sci., 23(8):1711–1728, 2012.
4. A. de Luca and A. Restivo. A characterization of strictly locally testable languages and its applications to subsemigroups of a free semigroup. Infor. and Cont., 44(3):300–319, 1980.
5. R. McNaughton and S. Papert. Counter-free Automata. MIT Press, 1971.
6. Y. T. Medvedev. On the class of events representable in a finite automaton. In E. F. Moore, editor, Sequential machines – Selected papers, pages 215–227. Addison-Wesley, 1964.
7. D. Perrin and C. Reutenauer. Hall sets, Lazard sets and comma-free codes. Discrete Mathematics, 341(1):232–243, 2018.
8. S. Eilenberg. Automata, Languages, and Machines, volume A. Academic Press, 1974.
9. J. Sakarovitch. Elements of Automata Theory. Cambridge University Press, 2009.
10. A. Szilard, S. Yu, K. Zhang, and J. Shallit. Characterizing regular languages with polynomial densities. In MFCS 1992, pages 494–503. Springer, 1992.