The Alphabetic Complexity in Homomorphic Definitions of Word, Tree and Picture Languages

Stefano Crespi Reghizzi(✉)

DEIB and CNR - IEIIT, Politecnico di Milano, Milan, Italy
stefano.crespireghizzi@polimi.it

Abstract. The Medvedev's Theorem (MT) characterizes a regular language as the projection of a local, i.e., a strictly-locally-testable language of order $k = 2$ (2-slt), over an alphabet larger than the terminal one by a factor depending on the state complexity of the finite automaton (FA). MT was later generalized to other language domains that instead of words contain trees or rectangular pictures, namely the regular tree languages and the tiling-system recognizable (TS-rec) languages. For trees and pictures the notion of local testability based on the occurrence of digram (2-factors) in a word, is changed into a suitable neighborhood of size 2, resp. a tree of height one, or a two-by-two tile, to be generically called 2-grams- A more recent MT extension goes in the direction of enlarging the neighborhood using as generators the languages, characterized by the k-grams, $k > 2$, and called k-slt; of course the k-gram types are different for words, trees and pictures. For all three domains a remarkably similar Extended Medvedev's Theorem (EMT) holds: a word/tree/picture language R over a terminal alphabet Σ is regular/regular/TS-rec if there exists a k-slt language L over an alphabet Λ of size double of Σ, and a letter-to-letter homomorphism from Σ to Λ such that R is the image of L; the value of k is in $\mathcal{O}(\lg(n))$, n being the state set size of an automaton recognizing the word/tree language R (the tiling system alphabet size for pictures). The alphabetic ratio $|\Lambda|/|\Sigma|$ of EMT is thus two, against the value n of MT. For some languages the value two of the ratio is the minimal possible. We present a new simplified proof of EMT for words and hints to the similar proofs for trees and for pictures. The central idea, say for words, is to sample a run of the FA so that the states traversed every k transition steps are evidenced; then each state is encoded by means of a comma-free binary code-word of length k; such an encoding is known to be a $2k$-slt language. For trees the same idea requires to lay the code-word bits over the root-to-leaf tree paths. For pictures 2D comma-free codes are needed. The possibility of further generalizations of EMT is raised at the end.

1 Introduction

Of the several approaches available to define formal language families, the homomorphic characterization of interest here essentially says that a language belongs to a certain family, if, and only if, it is the homomorphic image of a language belonging to a simpler sub-family. Since it would be difficult to be more precise without fixing a language family, we start with a classic case: Medvedev's theorem [15, 18] (MT) for regular word languages. The theorem says that a language R over a terminal alphabet Σ is regular if,

© IFIP International Federation for Information Processing 2022
Published by Springer Nature Switzerland AG 2022
Y.-S. Han and G. Vaszil (Eds.): DCFS 2022, LNCS 13439, pp. 1–14, 2022.
https://doi.org/10.1007/978-3-031-13257-5_1

and only if, there exists a *local* language L over another alphabet Λ and a letter-to-letter homomorphism $\vartheta : \Sigma^* \to \Lambda^*$ such that $R = \vartheta(L)$. Therefore each word $x \in R$ is the image $\vartheta(y)$ of a word y in Λ, also called a pre-image of x.

We recall that a local language is a *strictly locally testable* [14] (slt) language of testability order $k = 2$, in short a 2-slt language; the k-slt families form a strict hierarchy with respect to inclusion. The notion of local testability is the pivot of all the developments reported here, also for tree languages and for picture languages. For an slt language the validity of a word can be decided by a series of local tests: by moving on the word a sliding window of width k and collecting the window contents, i.e., the k-factors present in the word, and then by checking that they are included in a given set called a *test set*, where also the prefixes and suffixes are specified as such.

For the other two families, the k-factors, become respectively certain finite pieces of trees and of pictures, that we may call k-grams (k-tiles for pictures).

The local alphabet in the proof [18] of MT is $\Lambda = \Sigma \times Q$, where Q is the state set of a finite-state automaton (FA) recognizing language R. We prefer to say that the *alphabetic ratio* $\frac{|\Lambda|}{|\Sigma|}$ of MT is $|Q|$, to evidence how the local alphabet size depends on the state complexity of the language.

It is known that homomorphic characterizations *à la* Medvedev exist also for languages comprising entities more complex than words, when a suitable definition of locality and strict local testability is possible. This is the case of the *regular tree languages* (e.g. in [3,9]) and of the picture languages defined by the *tiling systems* [10,11] (TS-rec), for brevity simply called tree languages and picture languages.

The question addressed in a series of studies concerns the reduction of the alphabetic ratio that can be achieved in a Medvedev's theorem if the local language is replaced by a k-slt language, where $k \geq 2$. The series started with a result [5] for regular word languages, further developed in [6], then similar results where obtained for the regular tree languages [7], and lastly for the picture languages defined by the tiling systems [4]. In all such cases a similar *Extended Medvedev's Theorem* (EMT) with alphabetic ratio 2 was proved.

The EMT instance for words says that a word language is regular if, and only if, it is the image of a k-slt language of alphabetic ratio 2. Moreover the value of k is in $\mathcal{O}(\lg(|Q|))$ where Q is the state set of a finite-state automaton (FA) recognizing the language. The value two is the minimal possible for some regular languages, no matter how large a k is taken. Such an EMT is a new property that carries over from regular word languages to tree languages and to picture languages.

Since it would be too long to cover the word, tree and picture cases at the same level of detail, we concentrate on the basic case of words for which we provide a new simpler proof. The other two cases cannot be discussed here and their specific aspects are intuitively presented in a later section.

EMT for Words. It may be surprising at first that the EMT property holds for three very different language domains, but a deep justification comes from the common approach used in all the three proofs. We explain it, referring to the regular word languages over an alphabet Σ. The chief aspects present in the proof are:

(1) The focus in the MT proof [18] on the recognizing runs performed by an FA, represented as state-labeled paths over the local alphabet $\Lambda = \Sigma \times Q$. The projection on Σ are the recognized words.

(2) The encoding of the states on the run by means of *comma-free* code-words [2,12]. Actually not all the states are encoded but only those that occur $k-1$ transition steps apart from each other. This means that the states are taken with a fixed *sampling rate* $k \geq 2$. The value of k determines the code-word length, hence the code dictionary numerosity that must suffice to encode the states.
(3) The self-synchronization property of comma-free codes permits to decode such an encoded run using a $2k$-slt machine, which is the same as saying that the successful encoded runs belong to an slt language of order $2k$. The known results on the numerosity of binary comma-free codes of length k permit to prove that k is in $\mathcal{O}(\lg(|Q|))$.

We raise a technical point that has to do with the words (and also with trees and pictures) which are (i) smaller than k or (ii) of a size that is not multiple of k. Case (i) is easily dismissed as in the classic theory of k-slt word languages, by stipulating that any words shorter than k belonging to the language are simply united to the language defined by the k-slt test. On the other hand, in case (ii) a language may comprise an infinite number of words that cannot be encoded (as in item (2) above) with a uniform comma-free code $X \subset \{0,1\}^k$ since X just contains code-words of fixed-length k. The solution we adopt here transforms the language into a *padded* version by appending to every word not in $(\Sigma^k)^+$ up to $k-1$ new symbols ($) so that all words have the required length for encoding. EMT is proved initially for the padded language, obtaining the k-slt test set. The test set is then pruned from the $'s and adjusted to construct the final test set.

Paper Organization. Section 2 contains the basic definitions for word languages and for comma-free codes, and some preliminary properties. Section 3 contains a result on the minimal alphabetic ratio, the proof of EMT for the regular word languages, and a complete example. Section 4 briefly describes the cases of tree languages and picture languages by comparing them with the case of words. The Conclusion Sect. 5 indicates a possible generalization.

2 Basic Definitions and Properties

2.1 Regular and Strictly Locally Testable Word Languages

For brevity, we omit the classical definitions for language and automata theory and just list our notations. The empty word is denoted ε. The Greek upper-case letters Γ, Δ, Θ, Λ and Σ denote finite terminal alphabets.

The i-th letter of a word x is $x(i)$, $1 \leq i \leq |x|$, i.e., $x = x(1)x(2)\ldots x(|x|)$. The character # is not present in the alphabets, and is used as word *delimiter* to mark the start and end of a word, but it is not counted as true input symbol. A *homomorphism* $\xi : \Lambda^* \to \Sigma^*$ is *letter-to-letter* if for every $b \in \Lambda$, $\xi(b)$ is in Σ; we only use letter-to-letter homomorphisms.

A *finite automaton* (FA) $\mathcal{A}$ is defined by a 5-tuple $(\Sigma, Q, \to, I, F)$ where Q is the set of states, $\to$ is the state-transition relation (or graph) $\to \subseteq Q \times \Sigma \times Q$; I and F are resp. the nonempty 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 $\rightarrow$ is defined as usual, e.g., we also write $q \xrightarrow{x} q'$ with $x \in \Sigma^+$ with obvious meaning. If $q \in I$ and $q' \in F$, then the word x is *recognized* by $\mathcal{A}$.

Strictly Locally Testable Languages. There are different yet asymptotically equivalent definitions of the family of strictly locally testable (*slt*) languages [14]; the following definition is based on delimited words. Notice also that in the definition and throughout the paper we disregard for simplicity a finite number of short words that may be present in the language. The next notation is useful: given an alphabet Λ and for all $k \geq 2$, let $\Lambda^k_\# = \#\Lambda^{k-1} \cup \Lambda^k \cup \Lambda^{k-1}\#$. Thus the set $\Lambda^k_\#$ includes all the words of length k over Λ and all the words of length $k-1$ bordered on the left or on the right by $\#$. For all words x, $|x| \geq k$, let $F_k(x) \subseteq \Lambda^k_\#$ be the *set of factors of length* k (*k-factors*) present in $\#x\#$. The definition of F_k is extended to languages as usual.

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

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

2.2 Comma-Free Codes

A finite set or *dictionary* $X \subset \Delta^+$ is a *code* [2] if every word in Δ^+ has at most one factorization (i.e., decoding) in words of X, also known as *code-words*. We assume that Δ is the binary alphabet $\{0, 1\}$. We use a code X to represent a finite alphabet Γ by means of a one-to-one homomorphism, denoted by $[\![\ ]\!]_X : \Gamma^* \rightarrow \Delta^*$, called *encoding*, such that $[\![\alpha]\!]_X \in X$ for every $\alpha \in \Gamma$.

The family of codes we use is named comma-free [2,12] because the code-words in a text are not separated by a reserved character (the "comma"). Let $k \geq 1$. A code dictionary $X \subset \Delta^k$ is *comma-free* [12], if, intuitively, no code-word overlaps the concatenation of two code-words, more precisely:

$$X^2 \cap \Delta^+ X \Delta^+ = \emptyset \text{ i.e., } \forall t, u, v, w \in \Delta^* \text{ if } tu, uv, vw \in X \text{ then } \begin{cases} u = w = \varepsilon \vee \\ t = v = \varepsilon \end{cases} . \tag{1}$$

An example of comma-free code dictionary is $X = \{0010, 0011, 1110\}$.

Numerosity of Comma-Free Code Dictionaries

Proposition 1. *For every $m \geq 2$, there exist $k \geq 2$ and a comma-free code $X \subset \Delta^k$ such that $|X| \geq m$, with $k \in O(\lg m)$.*

Proof. Assume without loss of generality that m is a power of 2, i.e., $m = 2^h$ for some $h \geq 1$. Let k be any prime number between $2h$ and $4h$, which always exists by Bertrand-Chebyshev theorem; hence, k is in $O(\lg m)$. From [8] (as a special case of Theorem 2 pag. 267) it is known that for every prime integer $k > 1$ there is a comma-free code of length k with $\frac{2^k-2}{k}$ code-words, whence the following inequality: $\frac{2^k-2}{k} \geq \frac{2^{2h}-2}{2h} \geq \frac{2^{2h-1}}{2h}$. Then, this value is at least as large as m if $\frac{2^{2h-1}}{2h} \geq 2^h$, i.e., if $2^{h-1} \geq 2h$ which is true for every $h \geq 2$. □

Comma-Free Codes and slt Languages. To prepare for later proofs, we state a fundamental relation between comma-free codes and slt languages. Given an alphabet Λ and a comma-free dictionary X that encodes the symbols of Λ, let $L \subseteq \Lambda^*$ be a local language. Then the encoding of L, i.e., the language $[\![L]\!]_X \subseteq (\Delta^k)^*$ is slt. Such result is known and derives from early studies on local parsability [13, 17].

Theorem 1 (preservation of slt by encoding). *Let $L \subseteq \Gamma^+$ be the 2-slt language $L(M_2)$ defined by a test set $M_2 \subseteq \Gamma_{\#}^2$. The encoding of L by means of a comma-free code X of length k, i.e., the language $[\![L]\!]_X \subseteq (\Delta^k)^*$, is a $2k$-slt language.*

In the special case when $L = \Gamma\,\Gamma^+$ the encoding XX^+ is the $(2k)$-slt language defined by the set $F_{2k}(XX^+)$ of the factors of length $2k$ of $\#X\,X^+\#$.

3 The Extended Medvedev's Theorem for Words

Theorem 2 (Medvedev's Theor. for words (see e.g. [18])). *A language $R \subseteq \Sigma^*$ is regular if, and only if, there exist a local language $L \subseteq \Lambda^*$ and a letter-to-letter homomorphism $\vartheta : \Lambda^* \to \Sigma^*$ such that $R = \vartheta(L)$. If R is recognized by an FA with state set Q the alphabet is $\Lambda = \Sigma \times Q$.*

Thus, each element of Λ can be written as $\xrightarrow{a} q$, intuitively meaning that from some state an a labeled arc goes to state q. We call *alphabetic ratio* the quotient $\frac{|\Lambda|}{|\Sigma|}$. Thus the alphabetic ratio of the MT statement is $|Q|$. A natural question is whether, by relaxing the condition that language L is local and permitting it to be k-slt with $k > 2$, the alphabetic ratio of such an extended Medvedev's statement may be reduced and how much. First, we prove with a simple witness that in general, no matter how large k, the alphabetic ratio cannot be smaller than two.

Theorem 3 (minimality of alphabetic ratio two *[5]***).** *For every alphabet Σ, there exists a regular language $R \subseteq \Sigma^+$ such that for every $k \geq 2$, R is not the homomorphic image of a k-slt language $L \subseteq \Lambda$, with $|\Lambda| = (2 \cdot |\Sigma| - 1)$.*

Proof. Let $R = \bigcup_{a\in\Sigma} (aa)^*$. By contradiction, assume that there exist $k \geq 2$ and a local alphabet Λ of cardinality $2|\Sigma| - 1$, a mapping $\pi : \Lambda \to \Sigma$ and a k-slt language $L \subseteq \Lambda^+$ such that $\pi(L) = R$. Since $|\Lambda| = 2 \cdot |\Sigma| - 1$, there exists at least one element of Σ, say, a, such that there is only one symbol $b \in \Lambda$ with $\pi(b) = a$. Since $a^{2k} \in R$, there exists $x \in L$ such that $\pi(x) = a^{2k}$. By definition of π and of Λ, $x = b^{2k}$. Consider the word $xb = b^{2k+1}$. Clearly, $\pi(xb) = a^{2k+1}$, which is not in R, since all words in R have even length. Hence, $xb \notin L$. But the k-factors of $\#x\#$ coincide with the k-factors of $\#xb\#$ therefore xb is in L, a contradiction. □

In other words, some regular languages cannot be generated as images of an slt language, if the alphabetic ratio is too small. The remaining question whether an alphabetic ratio of two is sufficient was positively settled in [5].[1] The proof presented here is simpler than the original one and is based on the use of comma-free codes (as already in [6] where local functions are used instead of homomrphisms), combined with a convenient padding technique, already in [4] for picture languages.

Sampled Runs. Referring to Theorem 2, given $k \geq 2$, reorganize each computation as follows. Starting in the initial state, group together every k consecutive steps, until the computation ends in a final state or $h < k$ residual steps are left before the end; in the second case, group together all the h steps. Such a representation is called a *run with sampling rate* k schematized as:

$$\xrightarrow{y_1} q_{i_1} \xrightarrow{y_2} q_{i_2} \dots \xrightarrow{y_{n-1}} q_{i_{n-1}} \xrightarrow{z} q_{i_n}, y_i \in \Sigma^k,\ z \in \Sigma^h, (1 \leq h \leq k), q \in Q, q_{i_n} \in F.$$

Now, interpret each symbol such as $\xrightarrow{y_2} q_{i_2}$ in the run as an element of a finite set called *sampling alphabet*, $\Lambda_k = \Sigma^k \times Q \cup \Sigma^h \times F \ \ (1 \leq h \leq k)$ where $F \subseteq Q$ are the final states. Thus, a sampled run is a word over Λ_k. To illustrate, consider for the FA in Example 1 the sampled run recognizing a^5b: $\xrightarrow{aaaa} q_0 \cdot \xrightarrow{ab} q_2$

From Theorem 2 the following is obvious: (i) the projection on alphabet Σ of a sampled run is exactly the word recognized by the corresponding run of FA $\mathcal{A}$; and (ii) the language of sampled runs is local.

Padding to a Multiple of the Sampling Rate. To prove EMT we will encode the states visible in the sampled run, using binary comma-free code-words of length k. In Example 1 see at item 3. the encoding of the states by code-words of length 4. Thus the concatenation of the code-words $[\![q_1]\!]\,[\![q_0]\!]$ is a an 8 bit string, against an input word $aaaaab$ of length just 6, too short to assign one bit per input character when encoding the states visible in the run. Since it would be complicated to use code-words of variable length, we prefer to stretch the last symbol of a sampled run, in the example $\xrightarrow{ab} q_2$. We append to it as many symbols \$ (assumed not in Σ) as needed to obtain a length equal to the sampling rate. We call *padded* such modified sampled runs and from now on we only deal with them. In our case the padded sampled run is $\xrightarrow{aaaa} q_0 \ \xrightarrow{ab\$\$} q_2$.

Definition 2 (sampled runs). *The* sampling alphabet *(with padding) is* $\Lambda_{k\$} \subseteq \Sigma^k \times Q \cup \left(\Sigma^h \cdot \$^{k-h}\right) \times F$ *where* $1 \leq h < k$. *A sampled run is*

$$\xrightarrow{y_1} q_{i_1} \xrightarrow{y_2} q_{i_2} \dots \xrightarrow{y_{n-1}} q_{i_{n-1}} \xrightarrow{z} q_{i_n} \text{ where } \begin{cases} q_0 \xrightarrow{y_1} q_{i_1}, \\ y_i \in \Sigma^k, z \in \Sigma^h \cdot \$^{k-h}, \\ \text{all } q \in Q \text{ and } q_{i_n} \in F. \end{cases} \qquad (2)$$

The following proposition is immediate.

[1] A similar construction in [19] (proof of Theorem 5.2) is used to logically characterize regular languages; it may provide an alternative proof that the alphabetic ratio of EMT is two, though with the $k \in \mathcal{O}(|Q|)$ bound.

Proposition 2 (language of sampled runs). *Let $R \subseteq \Sigma^*$ be the language recognized by an FA $\mathcal{A}$. Let $k \geq 2$ be the sampling rate and $\Lambda_{k\$}$ the sampling alphabet with padding. A word x is in R if, and only if, $\mathcal{A}$ has a sampled run $y \in {\Lambda_{k\$}}^+$ such that the projection of y on Σ is equal to x. The language of sampled runs $L \subseteq (\Lambda_{k\$})^+$ is local.*

Merged Comma-Free Code-Words. It is convenient to introduce a binary operator that merges two words of identical length into one of the same length on the Cartesian product of the alphabets. Define the operator $\otimes : \Delta^+ \times \Sigma^+ \to (\Delta \times \Sigma)^+$ for any two words $y \in \Sigma^+, u \in \Delta^+$, with $|y| = |u|$, as: $y \otimes u = (y(1), u(1),) \ldots (y(|y|, u(|y|)))$. E.g., if $y = aabb$ and $u = 0101$ then $y \otimes u = (a, 0)(a, 1)(b, 0)(b, 1)$. The operator can be extended in the usual way to a pair of languages. We also need the projections, resp. denoted by $[\,]_\Sigma$ and $[\,]_\Delta$ onto the alphabets Σ and Δ i.e., $[u \otimes y]_\Sigma = y$, $[u \otimes y]_\Delta = u$.

Proposition 3 (merged comma-free code). *If $X \subset \Delta^k$ is a comma-free code of length k, then every subset $Z \subseteq \Sigma^k \otimes X$ is also a comma-free code of length k.*

Proof. We prove that Z satisfies the definition of comma-free code in Sect. 2.2, Eq. (1). Let $t, u, v, w \in (\Sigma \times \Delta)^*$ be such that $tu, uv, vw \in Z$. By definition of Z, $[tu]_\Delta$, $[uv]_\Delta$, $[vw]_\Delta \in X$, with $[u]_\Delta = [w]_\Delta = \varepsilon$ or $[t]_\Delta = [v]_\Delta = \varepsilon$ since X is a comma-free code; by definition of $\otimes$, it must be that also $u = w = \varepsilon$ or $t = v = \varepsilon$. □

The EMT for words (Theorem 8 of [5]) is now straightforward to prove.

Theorem 4 (Extended Medvedev's theorem for words). *For any regular language $R \subseteq \Sigma^*$, there exist an slt language $L \subseteq \Lambda^*$, where Λ is an alphabet of size $|\Lambda| = 2|\Sigma|$, and a letter-to letter homomorphism $\vartheta : \Lambda^* \to \Sigma^*$, such that $R = \vartheta(L)$.*
If R is recognized by an FA with $|Q|$ states, the language L is k-slt with k in $\mathcal{O}(\lg(|Q|))$.

Proof. Let $R = L(\mathcal{A})$ where $\mathcal{A} = (\Sigma, Q, \to, q_0, F)$. Let $k \in \mathcal{O}(\lg |Q|)$ be a value such that by Proposition 1 there is a comma-free dictionary X with $|X| = |Q|$. With reference to Definition 2, let $L \subseteq \Lambda_{k\$}^+$ be the 2-slt language of the sampled (padded) runs of $\mathcal{A}$. Define the comma-free code $Z = \Lambda_{k\$} \otimes X$, and apply this encoding to L, obtaining the language $[\![L]\!]_Z$. Notice that it exclusively contains (padded) words of length multiple of k. The language $[\![L]\!]_Z$ is $2k$-slt by Theorem 1. Denote with M_{2k} the test set such that $[\![L]\!]_Z = L(M_{2k})$.
The next transformation of M_{2k} eliminates or modifies the $2k$-factors containing one or more \$'s in order to clean $[\![L]\!]_Z$ from the padding symbols.

(1) Remove from set M_{2k} any $2k$-factor that contains as substring $(\$, \beta)(\$, \gamma)$ or $(\$, \beta)\#$, $\beta, \gamma \in \{0, 1\}$.
(2) At last, replace any occurrence of $(\$, \beta)$ with $\#$ (dropping the bit β).

Let M'_{2k} be the resulting set. Clearly, $L(M'_{2k}) \subseteq L(M_{2k})$. Since it is obvious that $[L(M'_{2k})]_\Sigma \subseteq R$, it remains to prove $[L(M'_{2k})]_\Sigma \supseteq R$.

By contradiction, assume that a sampled run α of $\mathcal{A}$ is such that $[\![\alpha]\!]_Z$ is not in $L(M'_{2k})$.

Let x be the projection of α on Σ. If $|x|$ is a multiple of k, all the $2k$-factors of $[\![\alpha]\!]_Z$ are free from \$ symbols, therefore they are all preserved in M'_{2k} since they are untouched by the steps 1. and 2. above. If $|x|$ is not a multiple of k, α terminates with a symbol of $\Lambda_{k\$}$ having the form $\overset{u(\$)^{k-h}}{\longrightarrow} q$ with $q \in F$ and $u \in \Sigma^h$, $1 \le h < k$. For brevity we discuss the case $\alpha = \ldots \overset{y}{\rightarrow} q' \ \overset{c(\$)^{k-1}}{\longrightarrow} q$ where $y = a_1 a_2 \ldots a_k \in \Sigma^k$ and $c \in \Sigma$. The encoded run $[\![\alpha]\!]_Z$ has therefore the form $\ldots a_1\beta_1\, a_2\beta_2 \ \ldots a_k\beta_k \ \ c\gamma_1\, \$\gamma_2 \ \ldots \ \$\gamma_{k-1}$ where each Greek letter stands for a bit. The $2k$-factor $a_2\beta_3 \ \ldots a_k\beta_k \ \ c\gamma_1\, \$\gamma_2$ occurs in $[\![\alpha]\!]_Z$, it is in M_{2k} and after step (2) above becomes $a_2\beta_3 \ \ldots a_k\beta_k \ \ c\gamma_1\, \# \in M'_{2k}$. Since all dollar-less $2k$-factors are untouched, $[\![\alpha]\!]_Z \in L(M'_{2k})$. $\square$

Example 1. The example illustrates the constructions used in the proof of EMT applied to a case simple enough to fit in the paper.

1. Finite automaton:

$$\begin{tikzcd}\mathcal{A} = \arrow[r] & q_0 \arrow[r, bend left, "a"] & q_1 \arrow[l, bend left, "a"] \arrow[r, "b"] & q_2 \arrow[r] & {}\end{tikzcd} \qquad R = a(aa)^*b$$

2. Sampled padded runs with sampling rate $k = 4$ (see Definition 2)
   – Alphabet $\Lambda_{4\$} = \left\{ \overset{aaaa}{\rightarrow} q_0,\ \overset{aaab}{\rightarrow} q_2,\ \overset{ab\$\$}{\rightarrow} q_2 \right\}$ and two sampled runs:

word	*sampled run*
a^5b	$\overset{aaaa}{\rightarrow} q_0 \cdot \overset{ab\$\$}{\rightarrow} q_2$
$a^{11}b$	$\overset{aaaa}{\rightarrow} q_0 \cdot \overset{aaaa}{\rightarrow} q_0 \cdot \overset{ab\$\$}{\rightarrow} q_2$

(3)

   – The language L of sampled runs is local (Proposition 2) and defined by the test set:

$$M_2 = \left\{ \begin{array}{l} \# \overset{aaaa}{\rightarrow} q_0,\ \# \overset{aaab}{\rightarrow} q_2,\ \# \overset{ab\$\$}{\rightarrow} q_2, \\ \overset{aaaa}{\rightarrow} q_0 \ \overset{aaaa}{\rightarrow} q_0,\ \overset{aaaa}{\rightarrow} q_0 \ \overset{aaab}{\rightarrow} q_2,\ \overset{aaaa}{\rightarrow} q_0 \ \overset{ab\$\$}{\rightarrow} q_2, \\ \overset{aaab}{\rightarrow} q_2 \,\#,\ \overset{ab\$\$}{\rightarrow} q_2 \,\# \end{array} \right\}$$

3. Comma-free dictionary X and encoding of FA states:

$[\![q_0]\!]_X$	$[\![q_1]\!]_X$	$[\![q_2]\!]_X$
0001	0011	0111

4. The merged comma-free dictionary $Z = \Lambda^4_{4\$} \otimes X$ of Proposition 3 is:

$aaaa \otimes [\![q_0]\!]_X$	$aaab \otimes [\![q_2]\!]_X$	$ab\$\$ \otimes [\![q_2]\!]_X$
$a0 \cdot a0 \cdot a0 \cdot 1a$	$a0 \cdot a1 \cdot 1a \cdot b1$	$a0 \cdot b1 \cdot \$1 \cdot \$1$

5. Apply the encoding $[\![\ldots]\!]_Z : (\Lambda_{4\$})^+ \rightarrow Z^+$ to the sampled runs, obtaining the language $[\![L]\!]_Z$, which is 8-slt by Theorem 1. It is defined by a test set M_8 that contains the 8-factors occurring in the runs at line (3):

$$\begin{array}{l} \#\ a0\ a0\ a0\ a1\ \ a0\ a0\ a0\ a1\ \ a0\ a1\ a1\ b1\ \# \\ \#\ a0\ a0\ a0\ a1\ \ a0\ b1\ \$1\ \$1\ \ \# \end{array}$$

6. To obtain the final test set M'_8, transform the set M_8 as follows:
   – The 8-factors $a0\ a0\ a0\ a1\ \ a0\ b1\ \$1\ \$1$ and $a0\ a0\ a1\ \ a0\ b1\ \$1\ \$1\ \#$ contain two \$ and are canceled.
   – The 8-factor $a1\ \ a0\ a0\ a0\ a1\ \ a0\ b1\ \$1$ is replaced by $a1\ \ a0\ a0\ a0\ a1\ \ a0\ b1\ \#$.

The test set is $M'_8 = \left\{ \begin{array}{l} \#\ a0\ a0\ a0\ a1\ \ a0\ a0\ a0,\ \ a0\ a0\ a0\ a1\ \ a0\ a0\ a0\ a1, \\ a0\ a0\ a1\ \ a0\ a0\ a0\ a1\ \ a0,\ a0\ a1\ \ a0\ a0\ a0\ a1\ \ a0\ a1, \\ a1\ \ a0\ a0\ a0\ a1\ \ a0\ a1\ a1,\ a0\ a0\ a0\ a1\ \ a0\ a1\ a1\ b1, \\ a0\ a0\ a1\ \ a0\ a1\ a1\ b1\ \#,\ \ \#\ a0\ a0\ a0\ a1\ \ a0\ b1\ \# \end{array} \right\}.$

7. The projection of language $L(M'_8)$ on Σ is $R - \{ab,\ aaab\}$.

Notice that this language admits a more economical *ad hoc* EMT statement with alphabetic ratio $\frac{3}{2}$, represented by the projection on $\{a, b\}$ of the 2-slt language $a(a'\ a)^*b$.

4 The Extended Medvedev's Theorem for Trees and Pictures

For comparability sake, the same presentation scheme is used in both cases: (i) the definition of the language family, (ii) the notion of k-gram, (iii) the Medvedev's theorem, (iv) the extended Medvedev's theorem.

4.1 Tree Languages

The ranked tree languages are recognized by nondeterministic root-to-leaves *tree automata* [3,9] (TA), assumed to be familiar to the reader. Given a tree with internal nodes labeled over the ranked alphabet Σ, and leaves labeled over $\Sigma_0 \subset \Sigma$, the machine starts in an initial state in the root, then it computes in one step the states of the sibling nodes. Then recursively it does the same for each sibling subtree, until the computation reaches a leaf. The state must be a final one in all leaves for the tree to be recognized. Thus the effect of the computation run is to label each node with a state from the state set Q.

The analogy with the runs of an FA on words is manifest; the difference is that an FA run traverses a linear graph whereas a TA run traverses a tree graph along all the root-to-leaf paths. The result is a *state-labeled* tree, isomorphic to the original one, where nodes are labeled over the alphabet $\Sigma \times Q$.

The notion of k-gram, $k \geq 2$, requires some preliminary concepts. A *tree domain* D is a finite, non empty, prefix-closed subset of $\mathbb{N}^*_{>0}$ satisfying the following condition: if $xi \in D$ for $x \in \mathbb{N}^*_{>0}$ and $i \in \mathbb{N}_{>0}$, then $xj \in D$ for all j with $1 \leq j \leq i$. A *tree* t over a finite alphabet Σ is a mapping $t : dom_t \to \Sigma$, where dom_t is a tree domain. A node of t is an element x of dom_t. The *root* of t is the node ε. The *successors* of a node x are all the nodes of the form xi, with $xi \in dom_t, i \in \mathbb{N}_{>0}$. The *yield* or frontier of a tree is the word of leaf labels read from left to right. A *path* is a sequence of nodes $x_1 \ldots x_m$, such that x_{i+1} is a successor of x_i. The *label of a path* $x_1 \ldots x_m$ is the word $t(x_1) \cdots t(x_m)$, i.e., the concatenation of the labels of nodes $x_1, \ldots, x_m$.

Given a node $x \in dom_t$, the portion of the tree rooted at x is denoted as $t_{|x}$ and called the *subtree* of t at node x, i.e., $t_{|x}$ is the subset of nodes of dom_t having the form xy, $y \in \mathbb{N}^*_{>0}$. A subtree $t_{|x}$ is not formally a tree but it can be made into a tree t' by removing the prefix x from every node of $t_{|x}$, by positing $t'(y) = t_{|x}(xy)$ for all $y \in \mathbb{N}^*_{>0}$; in this case, the subtree is said to be *normalized*.

Tree Languages Defined by Local Tests

Definition 3 (k-gram). *Let $k \geq 2$, let $t \in \mathcal{T}_\Delta$ and x a node of t. The* k-gram[2] *of t at node $x \in dom_t$ is the subset of nodes of the normalized subtree $t_{|x}$ at downwards*

[2] The meaning of digram is a sequence of 2 letters or symbols; or also of patterns such as the colors in a flag. For non-textual languages the term k-gram is preferable to k-factor. Other terms are in use, e.g., our definition of k-gram corresponds to the $(k-1)$-type of [16].

distance less than k from x. When $x = \varepsilon$ (i.e., the root of t) the k-gram is called a root *k-gram. The set of k-grams of t is denoted by $\langle\!\langle t \rangle\!\rangle_h$.*

Note: the yield of a k-gram, unlike the one of a tree, may include symbols in $\Sigma - \Sigma_0$.

Definition 4 (strictly locally testable tree language). *Let $k \geq 2$. A language $T \subseteq \mathcal{T}_\Sigma$ is k-*strictly locally testable *(k-slt) if there exist two sets Γ_k and Θ_k of k-grams, called the* test sets*, with $\Theta_k \subseteq \Gamma_k$, such that the membership of a tree in T can be decided just by considering its k-grams, namely, for all $t \in \mathcal{T}_\Sigma$: $t \in T$ if, and only if, $\langle\!\langle t \rangle\!\rangle_k \subseteq \Gamma_k$ and the root k-gram of t is in Θ_k. Then we write $T = T(\Gamma_k, \Theta_k)$.The value k is called the* order *of T. A language is* strictly locally testable *(slt) if it is k-slt for some $k \geq 2$; if $k = 2$ it is also called* local.

Two examples of local language are: the state-labeled trees, denoted by $\widehat{T(\mathcal{M})}$, of a language $T(M) \subseteq \mathcal{T}_\Sigma$, and the syntax trees of a context-free grammar.

Medvedev's Theorem and Its Extension The next well-known proposition (e.g., in [9], Sect. 2.8) is a Medvedev-like characterization of tree languages.

Theorem 5 (MT for trees). *A tree language $T \subseteq \mathcal{T}_\Sigma$ is regular if, and only if, there exists a ranked alphabet Λ, a local (i.e. 2-slt) tree language $T' \subseteq \mathcal{T}_\Lambda$ and a projection $\eta : \Lambda \to \Sigma$ such that $T = \eta(T')$. Moreover, if a tree automaton recognizing T has the state set Q, then the* alphabetic ratio *is $\frac{|\Lambda|}{|\Sigma|} \leq |Q|$.*

The proof of the theorem is based on the observation that the set of all state-labeled trees of a TA $\mathcal{M}$ is a local tree language, since a transition of $\mathcal{M}$ operates on the neighborhood of a node consisting of its children.

The EMT (Theorem 4) for words and the minimality of the alphabetic ratio 2 (Theorem 3) have been recently proved for tree languages [7]; the mimality result simply comes from the fact that a word is also a linear tree. therefore the same witness holds in both cases.

Theorem 6 (EMT for trees (theorem 1 of [7])). *For every ranked alphabet Σ, there exist a ranked alphabet Λ, with alphabetic ratio $\frac{|\Lambda|}{|\Sigma|} \leq 2$, and a projection $\eta : \mathcal{T}_\Lambda \to \mathcal{T}_\Sigma$, such that for every regular tree language $T \subseteq \mathcal{T}_\Sigma$ there exist $k \geq 2$ and a $2k$-slt tree language $\widetilde{T} \subseteq \mathcal{T}_\Lambda$ such that $T = \eta(\widetilde{T})$.*

To explain the proof of EMT for trees, we look at a state-labeled tree, and we consider each root-to-leaf path. On each path we encode, with binary comma-free code-words of length k, the states that are located at nodes at distance $0, k, 2k, \ldots$ from the root. Notice that, for any internal node at a distance multiple of k from the root, the same code-word is placed on all the downward paths originating from it. Paths too short to contain a whole code-word, will contain just a prefix. Having placed the binary code-words on the path, we cancel all state labels that where present in the state-labeled tree. The result is a tree isomorphic to the state-labeled one, aptly called an *encoding* tree. As said, its node labels are over the alphabet $\Sigma \times \{0, 1\}$. Analogously with Theorem 1 about the preservation of the slt property by encoding, it is possible to prove that the language of encoded trees is $2k$-slt, and its projection on Σ coincides with language T.

4.2 Picture Languages

The case of TS-rec pictures sets itself apart from the cases of words and trees since the primary definition of the tiling-system recognizable picture languages is not based on an automaton (or on a regular expression) but on Medvedev's theorem.

Assuming some familiarity with the subject, we list a few essential definitions (see [10,11]). A picture p is a rectangular array with $|p|_{row}$ rows and $|p|_{col}$ columns, each cell containing a symbol (or pixel) from an alphabet Σ. The set of all pictures of size m, n is $\Sigma^{m,n}$ and the set of all pictures is Σ^{++}. A picture contained in another one is a subpicture. A (square) picture in $\Sigma^{k,k}$, $k \geq 2$, is called k*-tile* and simply *tile* if $k = 2$: k-tiles play the role of k-factors for words and k-grams for trees. The *bordered* version $\widehat{p}$ of p is the picture of size $(|p|_{row} + 2,\ |p|_{col+2}$ that surrounds p with a rectangular frame containing the reserved symbol $\#$.

Definition 5 (strictly locally testable picture languages). *A picture language, $L \subseteq (\Sigma \cup \{\#\})^{++}$ is k-*strictly-locally testable *(k-slt) if there is a set $T_k \subseteq (\Sigma \cup \{\#\})^{k,k}$ of k-tiles, called the* test set *such that $p \in L$ if, and only if, the k-tiles occurring in $\widehat{p}$ as subpictures are included in T_k. Then we write $L = L(T_k)$. A pictures language is slt if it is k-slt for some k.*

The k-slt, $k \geq 2$, family is an infinite strict hierarchy, for every non-unary alphabet.

As said, the definition of TS-rec is an MT statement.

Theorem 7 (MT for pictures). *A picture language $R \subseteq \Sigma^{++}$ is* tiling-system recognizable *(TS-rec) if it is the* projection *of a 2-slt (i.e., local) language $L \subseteq \Lambda^{++}$, i.e., defined by a test set $T_2 \subseteq (\Sigma \cup \{\#\})^{2,2}$. In formula, $R = \pi(L(T_2))$ where $\pi : \Lambda \to \Sigma$. The quadruple $(\Sigma, \Lambda, T_2, \pi)$ is called a* tiling system.

Tradeoff Between Alphabet Cardinality and Tile Size. The definition of tiling system has been extended towards the *k-tiling system*. It uses a set $T_k \subseteq \Gamma^{k,k}$ of k-tiles, $k \geq 2$ instead of 2-tiles. The *alphabetic ratio* of a k-tiling system is the quotient $\frac{|\Gamma|}{|\Sigma|}$.

Theorem 8. *Given a k-slt language $L \subseteq \Sigma^{++}$ defined as $L = L(T_k)$ where $T_k \subseteq (\Sigma \cup \{\#\})^{k,k}$, there exists an alphabet Γ, a local language $L' \subseteq \Gamma^{++}$ and a projection $\pi : \Gamma \to \Sigma$ s.t. $L = \pi(L')$. Hence the family of k-TS recognizable languages coincides with TS-rec.*

The proof in [10] has the alphabet size $|\Gamma| = |\Sigma| \cdot |T_k|$.

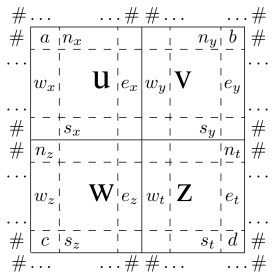
Extended Medvedev's Theorem. The EMT (Theorem 4) for words and the minimality of the alphabetic ratio 2 (Theorem 3) have been recently proved for TS-rec pictures [4]; the minimality simply follows from the fact that a word is also a one-row picture.

Theorem 9 (EMT for pictures). *For any $R \subseteq \Sigma^{++}$ in TS-rec there exist $k \geq 2$ and a $2k$-tiling system with alphabetic ratio 2, recognizing R. Moreover, if n is the size of the local alphabet of a tiling system recognizing R then the value of k is $\mathcal{O}(\lg(n))$.*

The proof follows the one for words in Sect. 3 with some important differences.

1. If the number of rows or columns is not multiple of k, the picture has to be *padded* (as we did for words in Sect. 3) on the east and south sides with \$ symbols, so that the padded picture can be tessellated with k-tiles, that fit in a mesh of a $k \times k$ grid.
2. A comma-free picture (i.e., 2D) code is, intuitively, a set of k-tiles (code-pictures) such that, for any picture tessellated with such tiles, none of the $k \times k$ subpictures at positions misaligned with respect to the grid, can be a code-picture. The slt properties of such 2D codes and for code-words are similar.
   The property of code-words that XX^+ is $2k$-slt (special case of Theorem 1) becomes: let $X \subseteq \Lambda^{k,k}$ be a comma-free code, then the language X^{++} is $2k$-slt.
   Instead of Theorem 1, the statement is: let $T \subseteq \Gamma^{2,2}$ be a set of tiles defining the local language $L(T)$ and let $X \subseteq \Delta^{k,k}$ be a comma-free 2D code with numerosity $|X| = |\Gamma|$. The encoding $[\![L(T)]\!]_X$ is a $2k$-slt language.
   Although a general formula for the numerosity of 2D comma-free codes is lacking, in [1] a useful lower bound for a family of codes that cannot overlap is given; the non-overlapping condition is stronger than the comma-free one.
3. The major difference is that for pictures we cannot rely on an automaton analogous to FA for words and TA for trees.

   Therefore the technique in Sect. 3 of sampled computations labeled with the states traversed has to be replaced by another approach. The *frame* $f(p)$ of a picture $p \in \Gamma^{k,k}$ is the square ring composed by the four sides (n_p, e_p, s_p, w_p) each one being a word of length k; each corner is shared by two words. A bordered picture of size $(2k, 2k)$ tessellated with four k-tiles each one with its frame, is shown.



4. A comma-free code-picture encodes each frame, i.e., a quadruple (n_p, e_p, s_p, w_p); code-pictures are schematized by u, v, w, z in the figure. The frame contains $4k$ symbols of Γ and can be encoded by a code-picture in X, which contains k^2 bits, since for k large enough, the numerosity of the family of non-overlapping 2D codes [1] suffices to encode all possible frames.
5. In each pixel the original terminal symbol from Σ is accompanied by a bit of the code-picture, so that the alphabetic ratio is $\frac{|\Sigma| \cdot |\{0,1\}|}{|\Sigma|} = 2$.
6. The language of encoded pictures is $2k$-slt, hence language R is $2k$-TS recognizable with alphabetic ratio 2. The proof is more combinatorial than for words.

5 Conclusion

For words, trees and pictures we have evidenced the similarity between the statements and proofs of the Extended Medvedev's Theorem. At closer reflection, we may attribute such similarity to the fact that in all cases a recognizing computation sweeps over a discrete structure, a directed graph whose nodes are labeled with terminal symbols and states. The graph is respectively a total order, a tree order, and an acyclic graph with square meshes. The computation never returns to an already visited node. The sampling technique with sampling rate k clusters the computation into k-grams. Such k-grams are then taken as symbols of a new alphabet and the correct adjacencies between k-grams are specified by a 2-slt language. Then the encoding of each k-gram symbol by means of a binary comma-free code transforms the 2-slt language into a $2k$-slt language over the doubled alphabet $\Sigma \times \{0, 1\}$.

An open question is whether the EMT property holds for other language domains beyond the three considered, as for instance the directed-acyclic-graph automata that from time to time have been proposed in the literature.

Acknowledgment. The present simpler proof of EMT for words has been worked out jointly with P. San Pietro and incorporates an original idea of A. Restivo about padding.

References

1. Anselmo, M., Giammarresi, D., Madonia, M.: Non-expandable non-overlapping sets of pictures. Theor. Comput. Sci. **657**, 127–136 (2017)
2. Berstel, J., Perrin, D., Reutenauer, C.: Codes and Automata. Encyclopedia of Mathematics and its Applications, vol. 129. CUP (2009)
3. Comon, H., et al.: Tree automata techniques and applications (2007). http://www.grappa.univ-lille3.fr/tata
4. Crespi Reghizzi, S., Restivo, A., San Pietro, P.: Reducing local alphabet size in recognizable picture languages. In: Moreira, N., Reis, R. (eds.) DLT 2021. LNCS, vol. 12811, pp. 103–116. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81508-0_9
5. Crespi-Reghizzi, S., San Pietro, P.: From regular to strictly locally testable languages. Int. J. Found. Comput. Sci. **23**(8), 1711–1728 (2012)
6. Crespi Reghizzi, S., San Pietro, P.: Regular languages as local functions with small alphabets. In: Ćirić, M., Droste, M., Pin, J.É. (eds.) CAI 2019. LNCS, vol. 11545, pp. 124–137. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21363-3_11
7. Crespi Reghizzi, S., San Pietro, P.: Homomorphic characterization of tree languages based on comma-free encoding. In: Leporati, A., Martín-Vide, C., Shapira, D., Zandron, C. (eds.) LATA 2021. LNCS, vol. 12638, pp. 241–254. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68195-1_19
8. Eastman, W.L.: On the construction of comma-free codes. IEEE Trans. Inf. Theory **11**(2), 263–267 (1965)
9. Gecseg, F., Steinby, M.: Tree Automata. arXiv (2015)
10. Giammarresi, D., Restivo, A.: Recognizable picture languages. Int. J. Pattern Recogn. Artif. Intell. **6**(2–3), 241–256 (1992)
11. Giammarresi, D., Restivo, A.: Two-dimensional languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, pp. 215–267. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-642-59126-6_4
12. Golomb, S.W., Gordon, B., Welch, L.: Comma-free codes. Can. J. Math. **10**, 202–209 (1958)
13. Hashiguchi, K., Honda, N.: Properties of code events and homomorphisms over regular events. J. Comput. Syst. Sci. **12**(3), 352–367 (1976)
14. McNaughton, R., Papert, S.: Counter-Free Automata. MIT Press, Cambridge (1971)
15. Medvedev, Y.T.: On the class of events representable in a finite automaton. In: Moore, E.F. (ed.) Sequential Machines - Selected Papers, pp. 215–227. Addison-Wesley (1964). Originally Published in Russian in Avtomaty, pp. 385–401 (1956)
16. Place, T., Segoufin, L.: A decidable characterization of locally testable tree languages. Log. Methods Comput. Sci. **7**(4) (2011)

17. Restivo, A.: On a question of McNaughton and Papert. Inf. Control **25**(1), 93–101 (1974)
18. Eilenberg, S.: Automata, Languages, and Machines, vol. A. Academic Press (1974)
19. Thomas, W.: Classifying regular events in symbolic logic. J. Comput. Syst. Sci. **25**(3), 360–376 (1982)