Theoretical Computer Science 1002 (2024) 114598

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

From words to pictures: Row-column combinations and Chomsky-Schützenberger theorem ☆

Stefano Crespi Reghizzi [a], Antonio Restivo [b], Pierluigi San Pietro [a,*]

[a] Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB) and CNR - IEIIT Politecnico di Milano, Milano, Italy
[b] Dipartimento di Matematica e Informatica, Università di Palermo, Palermo, Italy

ARTICLE INFO

Keywords:
Picture languages
Dyck language
Chomsky-Schützenberger theorem
Crosswords

ABSTRACT

The row-column combination RCC maps two (word) languages over the same alphabet onto the set of rectangular arrays, i.e., pictures, such that each row/column is a word of the first/second language. The resulting array is thus a crossword of the component words. Depending on the family of the components, different picture (2D) language families are obtained: e.g., the well-known tiling-system recognizable languages are the alphabetic projection of the crossword of local (regular) languages. We investigate the effect of the RCC operation especially when the components are context-free, also with application of an alphabetic projection. The resulting 2D families are compared with others defined in the past. The classical characterization of context-free languages, known as Chomsky-Schützenberger theorem, is extended to the crosswords in this way: the projection of a context-free crossword is equivalent to the projection of the intersection of a 2D Dyck language and the crossword of strictly locally testable language. The definition of 2D Dyck language relies on a new more flexible so-called Cartesian RCC operation on Dyck languages. The proof involves the version of the Chomsky-Schützenberger theorem that is non-erasing and uses a grammar-independent alphabet.

1. Introduction

Many efforts in the past have been made to extend the formal language theory from words to bidimensional (2D) symbolic arrays named pictures. Different approaches have been used including 2D automata, regular expressions, grammars and tiling systems; for a historical view, the reader may look at old and recent surveys [1–4]. Our approach is based on the operation of *row-column composition* (RCC) that takes two (word) languages on the same alphabet and creates the rectangular arrays such that each row and each column is respectively in the first and second component language. Such pictures may be viewed as *crosswords* [5] of the component words. RCC offers a very simple and natural way of mapping word languages on 2D languages, but surprisingly it has received little attention in the past, almost exclusively in connection with the characterization of the tiling recognizable languages [6,3] (denoted as REC) as the projection of the RCC of the local languages [7]. Here we focus on the combination of context-free (CF) languages, denoted by $\mathbb{C}(CF)$, an idea already suggested in [3] as a natural extension of CF languages in 2D.

☆ This article belongs to Section C: Theory of natural computing, Edited by Lila Kari.
* Corresponding author.
E-mail addresses: stefano.crespireghizzi@polimi.it (S. Crespi Reghizzi), antonio.restivo@unipa.it (A. Restivo), pierluigi.sanpietro@polimi.it (P. San Pietro).

https://doi.org/10.1016/j.tcs.2024.114598
Received 19 December 2023; Received in revised form 16 April 2024; Accepted 22 April 2024

Available online 25 April 2024
0304-3975/

Other researchers have studied the relationship between word and picture languages from a very different viewpoint than ours, and we mention a few examples. In a series of studies, the frontier (or row-projection) of a picture is defined as the word in the top row, while the frontier of a picture language is the word language containing the frontiers of all pictures. In [8] it is proved that the frontiers of REC coincide with the family of context-sensitive languages. Other results for certain subclasses of context-sensitive languages are in [9], while the recent study [10] examines the frontiers of languages recognized by various two-dimensional automaton classes.

Second, we mention the researches concerning the generalization to two dimensions of some important results in combinatorics on words as, for instance, the Fine-Wilf Theorem [11] and the Lyndon-Schutzenbeger Theorem [12].

In our opinion, the interest of a 2D language model depends on two aspects: (i) its formal properties, and (ii) its expressiveness in the sense of the richness of the pictures generated. For instance, the REC family enjoys many of the properties of the regular language family, starting from its very definition as the homomorphic image of a 2D local language [6]; this extends in 2D the well-known homomorphic definition of regular languages (also known as Medvedev's theorem). The existence of equivalent definitions of REC in terms of 2D automata [13] and logical predicates [14] adds to the robustness of such a family. Criteria (i) and (ii) should be followed, in perspective, also for the new families here investigated.

In this work, we go some way towards the study of two families, first the pure RCC of CF languages, denoted $\mathbb{C}(CF)$; and ultimately the family obtained by its projection, denoted by $\mathbb{PC}(CF)$. Of course the latter family strictly includes REC because of the strict inclusion of the corresponding word language families. Since little was known about RCC, we had to start from the basic properties of this operation by studying the 2D families produced by its application to the simpler families of local (LOC) and regular (REG) word languages, before entering the context-free domain. We prove that $\mathbb{C}(CF)$ is incomparable with REC, whereas $\mathbb{PC}(CF)$ strictly includes REC.

Our main objective was to find whether the Chomsky-Schützenberger theorem (C-S Theorem) characterizing the context-free languages can be extended in 2D. In the original C-S Theorem [15] for words, stating that every context-free language is the homomorphic image of the intersection of a Dyck and local languages, the homomorphism may erase some parentheses of the Dyck language and maps the others onto the terminal symbols of L. Since in a picture it is impossible to erase a cell without creating holes, the suitable form of the C-S Theorem is a non-erasing variant in [16,17] and especially in [18].

Whereas the notions of homomorphism and local language have a natural extension in 2D, in order to state a C-S Theorem in 2D, a suitable definition of the concept of 2D Dyck language is needed. The first naive idea is to define a 2D Dyck as the RCC of two Dyck word languages. One then obtains a very interesting family of 2D languages that has been investigated in the previous study [19], but it does not work for a C-S Theorem in 2D.

The actual successful reformulation of the C-S Theorem is built on a new row-column combination operation, to be called *Cartesian RCC*. It takes as arguments two word languages L_1, L_2 over possibly distinct alphabets, and defines the alphabet of the symbols (pairs) belonging to their Cartesian product. The result of the Cartesian RCC is the set of pictures such that, for each row, the projection of each symbol on the first component is a word in L_1; and similarly for the columns, by taking the projection on the second component, and L_2. We name *2D Dyck* the language resulting from the *Cartesian* RCC of Dyck languages.

With such a 2D Dyck language, we obtained our main result, the C-S Theorem (Theorem 6), which says that the projective crosswords of context-free languages $\mathbb{PC}(CF)$ coincide with the projection of the intersection of 2D Dyck languages with strictly-locally-testable pictures. Notice that the same statement in the case of projection-less RCC holds only in one direction, as a representation theorem –not as a characterization.

Of course, other relevant technical devices for obtaining such a result are presented in the main sections, e.g., the technique for extending the non-erasing C-S Theorem to languages whose pictures may have an odd number of rows or columns.

We have summarily exposed our main contribution but other results of interest are present in the paper that it would be long to describe at this stage. We just mention the comparison of the $\mathbb{C}(CF)$ family with some known 2D families defined by 2D grammars of various types (e.g., Kolam and Tiling grammars), all having been proposed as 2D extensions of context-free grammars. The generative capacity of CF crosswords is witnessed by some examples, in particular the suggestive combinatorial structures of HV-palindromes [20].

Paper organization. Section 2 contains the basic definitions and properties. Section 3 defines RCC and states some general properties. Section 4 focuses on the RCC of local and regular languages. Section 5 studies the RCC of context-free languages and compares with some known families of grammar-based 2D extensions of CF languages. Section 6 introduces the Cartesian RCC operation and applies it to the Dyck languages. Section 7 proves the Chomsky-Schützenberger theorem for the projective context-free crosswords. Section 8 concludes.

2. Basic definitions and properties

All the alphabets considered are finite.

2.1. Word languages

We use the traditional notation and terminology of formal language theory, e.g., in [21,22]. The empty word is denoted by ε. The reversal of a word w is denoted by w^R. The number of times a letter a occurs in a word w is denoted by $\#_a(w)$.

Definition 1 *(strictly locally testable languages)*. Let $k \geq 2$. A language L over alphabet Σ is *k-strictly locally testable* (k-SLT) if there exist finite sets $W \subseteq \Sigma \cup \Sigma^2 \cup \ldots \cup \Sigma^{k-1}$, I_{k-1}, $T_{k-1} \subseteq \Sigma^{k-1}$, and $F_k \subseteq \Sigma^k$ such that, for every $x \in \Sigma^*$, $x \in L$ if, and only if:

- if $|x| \leq k-1$, then $x \in W$.
- If $|x| > k-1$, then the $k-1$-prefix and $k-1$-suffix of x are resp. in I_{k-1} and in T_{k-1}, and
- each factor of x with length k is in F_k.

A language is *strictly locally testable* (SLT) if it is k-SLT for some k; it is *local* (LOC) if it is 2-SLT.

Dyck languages The definition and properties of Dyck word languages are basic concepts in formal language theory.

An alphabet Δ is Dyck if it is associated with a bijection, defined by a *coupling relation* $\widehat{\Delta} \subseteq \Delta_{\langle} \times \Delta_{\rangle}$ where $\Delta_{\langle}, \Delta_{\rangle}$ are two sets of the same cardinality forming a partition of Δ. Each element of the coupling relation is called a *coupled pair* and its elements are also called *open/closed* parentheses.
The Dyck language D over Δ is the set of words congruent to ε, via the *cancellation rule* $aa' \to \varepsilon$ where $(a, a') \in \widehat{\Delta}$, which erases two adjacent letters. Given a word over Δ, two occurrences of the coupled letters a and a' are *matching* if they are erased by the same cancellation rule application. Notice that in a Dyck word the two letters of a matching pair are separated by an even or null number of letters.

Chomsky-Schützenberger theorem In the original form (e.g., [22,15]), this well-known theorem states that a language L is context-free (CF) if, and only if, it is the alphabetic homomorphism of the intersection of a Dyck language D and a local language R, i.e., $L = h(D \cap R)$. The homomorphism $h : \Delta^* \to \Sigma^*$ maps each letter of the Dyck alphabet Δ to a letter of the terminal alphabet Σ or to the empty word. Since for pictures, the erasure of a letter would create a hole, we have to choose a later formulation of the theorem where the homomorphism is non-erasing, i.e., it is a letter-to-letter projection [16,17]. The recent non-erasing variant in [18] is more convenient, because the Dyck alphabet size is independent of the language complexity (in terms of grammar size) and depends only on the size of the terminal alphabet.

An obvious limitation of any non-erasing version is that it only applies to words of even length. An approach to cope also with odd length words, followed in [17], is to allow a homomorphism that may map a Dyck letter to two symbols rather than one, with the regular language allowing only at most one occurrence of such a letter in a Dyck word (e.g., at the beginning of a word). The other approach also in [17]) slightly generalizes the notion of Dyck language, by extending the Dyck alphabet with a finite set of *neutral symbols*. Following [18], to make a word of odd length, we append one neutral symbol the Dyck words: a Dyck language on a Dyck alphabet Δ with neutral symbols is the set of words over Δ of either the form $w\delta$ or w, where w is a Dyck word without neutral symbols and $\delta \in \Delta$ is neutral. For instance, let δ_0, δ_1 be two neutral symbols and let $\Delta = \{a, a', b, b'\} \cup \{\delta_0, \delta_1\}$ with coupling $\{(a, a'), (b, b')\}$. Then, two examples of words of the Dyck language over Δ are $aabb'a'a'\delta_0$ and $abb'a'aa'bb'aa'\delta_1$.

The version of the C-S theorem used in the proof of our main theorem (in Sect. 7) is the following:

Theorem 1 *(Theorem 4 of [18]). For every finite alphabet Σ, there exist a number $q > 0$, a Dyck alphabet Δ with q pairs of parentheses and $|\Sigma|$ neutral symbols, a Dyck language D over Δ and a projection $h : \Delta \to \Sigma$ such that, for every context-free language $L \subseteq \Sigma^*$, there exists an SLT language R satisfying $L = h(D \cap R)$.*

When all the words in L have even length, then in the above theorem we can replace the alphabet Δ with a Dyck alphabet without neutral symbols (Theorem 5 of [18]). The inverse of the theorem, i.e., that $h(D \cap R)$ is CF, is not reported here since it follows immediately from the closure properties of the CF family.

2.2. Picture languages

The concepts and notations for picture languages follow mostly [3]. We assume some familiarity with the basic theory of the family REC of tiling system languages, defined as the projection of a local 2D language; the relevant properties of REC will be reminded when needed. A *picture* is a rectangular array of letters over an alphabet. The set of all non-empty pictures over Σ is denoted by Σ^{++}. A letter at a given position in the array is called a pixel. Given a picture p, the pixel at coordinates (i, j) is denoted as $p_{i,j}$. Given a picture p, $|p|_{row}$ and $|p|_{col}$ denote the number of rows and columns, respectively.
Let $p, q \in \Sigma^{++}$. The *row concatenation* of p and q is denoted as $p \oplus q$ and it is defined when $|p|_{row} = |q|_{row}$. Similarly, the *column concatenation* $p \ominus q$ is defined when $|p|_{col} = |q|_{col}$. We also use the power operations $p^{\ominus k}$ and $p^{\oplus k}$, $k \geq 1$, their closures $p^{\ominus +}$, $p^{\oplus +}$ and the closure under both concatenations $p^{\ominus +, \oplus +}$; concatenations and closures are extended to languages in the obvious way.

Let $P \subseteq \Sigma^{++}$ be a picture language. Define the *row language* of P as $\text{ROW}(P) = \{w \in \Sigma^+ \mid \text{there exist } p \in P, p', p'' \in \Sigma^{++} \text{ such that } p = w \ominus p'' \text{ or } p = p' \ominus w \ominus p'' \text{ or } p = p' \ominus w\}$. The *column language* of P, $\text{COL}(P)$, is defined symmetrically.

We recall the definition of picture homomorphism from [23].

Definition 2 *(picture homomorphism)*. Given two alphabets Γ, Σ, a (*picture*) *homomorphism* is a mapping $\varphi : \Sigma^{++} \to \Gamma^{++}$ such that, for all $p, q \in \Sigma^{++}$:

$$\begin{cases} i) & \varphi(p \oplus q) = \varphi(p) \oplus \varphi(q) \\ ii) & \varphi(p \ominus q) = \varphi(p) \ominus \varphi(q) \end{cases}$$

The definition implies that the images by φ of the symbols of a picture are isometric, i.e., for any $x, y \in \Sigma$, $|\varphi(x)|_{row} = |\varphi(y)|_{row}$ and $|\varphi(x)|_{col} = |\varphi(y)|_{col}$.

3. Row-column combination and its general properties

Following [3,7] we introduce the *row-column combination* (RCC) operation that takes two word languages over the same alphabet Σ and produces a picture language by making a (rectangular) crossword having the rows in the first language and the columns in the second. Then, given a projection from Σ to another alphabet, we introduce the RCC with projection.

Definition 3 *(row-column combination a.k.a. crossword).* Let L_1, L_2 be two word languages over Σ. Their *row-column combination* (RCC) or *crossword* is the picture language over Σ denoted by $L_1 \boxplus L_2$ and defined as

$$L_1 \boxplus L_2 = L_1^{\ominus *} \cap L_2^{\oplus *}.$$

A picture language that can be defined by a row-column combination is called an RCC. Given a projection $h : \Sigma \to \Delta$ over another alphabet Δ, the *projective RCC* of L_1, L_2 with projection h is the picture language $h(L_1 \boxplus L_2) \subseteq \Delta^{++}$.
Let $\mathcal{F}$ be a family of word languages. The row-column combination $\mathbb{C}(\mathcal{F})$ and the projective row-column combination $\mathbb{PC}(\mathcal{F})$ of family $\mathcal{F}$ are the families of picture languages of the form $L_1 \boxplus L_2$ and, respectively, of the form $h(L_1 \boxplus L_2)$, where the languages L_1 and L_2 of $\mathcal{F}$ are over the same alphabet, and h is a projection.

We illustrate with an example.

Example 1 *(square with a single 1 in every row/column).* Let $L = 0^*10^*$. Then $L \boxplus L \in \mathbb{C}(REG)$ is the set of square pictures such that in every row and column exactly one "1" occurs. An example is the picture:

0	1	0	0
0	0	0	1
0	0	1	0
1	0	0	0

Notice that the language L_{sq} of square pictures over a *unary* alphabet is not in $\mathbb{C}(REG)$, but it is well-known to be in $\mathbb{PC}(REG)$ [3]. Intuitively, L_{sq} is not $\mathbb{C}(REG)$ since its row and column languages of L_{sq} are 0^+ and thus they cannot "control" the size of a picture (see also Proposition 1 below). To see that L_{sq} is in $\mathbb{PC}(REG)$, consider the crossword $L_1 \boxplus L_2$ of two local languages $L_1 = 0^*12^*$ and $L_2 = 2^*10^*$: it defines the language of square pictures over the alphabet $\{0, 1, 2\}$; a projection of $L_1 \boxplus L_2$ to a unary alphabet defines L_{sq}. See also Example 2.

3.1. Basic RCC properties

The RCC operation has not received much attention in the past, and we start from several immediate properties of RCC languages. The operator $\boxplus$ distributes over intersection, i.e.,

$$(L_1 \boxplus L_2) \cap (L_3 \boxplus L_4) = (L_1 \cap L_3) \boxplus (L_2 \cap L_4). \tag{1}$$

As a consequence, if a family $\mathcal{F}$ is closed under intersection, then also $\mathbb{C}(\mathcal{F})$ is closed under intersection. Moreover, if $\mathcal{F}$ is closed under intersection with another family $\mathcal{F}'$, then $\mathbb{C}(\mathcal{F})$ is closed under intersection with $\mathbb{C}(\mathcal{F}')$.

On the other hand, the operator $\boxplus$ may not distribute over projection. Let L_1 and L_2 be word languages over an alphabet Δ and let $h : \Delta \to \Sigma$ be a projection. Then,

$$h(L_1 \boxplus L_2) \subseteq h(L_1) \boxplus h(L_2). \tag{2}$$

The inclusion can be strict, e.g. let $L_1 = \alpha^*, L_2 = \beta^*$ and let $h(\alpha) = h(\beta) = a$. Then, $h(L_1 \boxplus L_2) = \emptyset$, while $h(L_1) \boxplus h(L_2) = a^{++}$.

The next property of the row and column languages will be used later to prove that a language is not an RCC.

Proposition 1. *Let P be a picture language.*
$P = ROW(P) \boxplus COL(P)$ if, and only if, P is the RCC of two word languages.

Proof. It is obvious that if $P = \mathrm{ROW}(P) \boxplus \mathrm{COL}(P)$ then P is an RCC. Assume now that $P = L_1 \boxplus L_2$ for two word languages L_1, L_2. By definition of RCC, $\mathrm{ROW}(P) \subseteq L_1$ and $\mathrm{COL}(P) \subseteq L_2$. Therefore, $\mathrm{ROW}(P) \boxplus \mathrm{COL}(P) \subseteq L_1 \boxplus L_2 = P$. Since $P \subseteq \mathrm{ROW}(P) \boxplus \mathrm{COL}(P)$, the statement follows. □

Notice that when P is not an RCC, then $P \subsetneq \mathrm{ROW}(P) \boxplus \mathrm{COL}(P)$. For instance, let $P = \left\{ \begin{array}{|c|}\hline aa \\ aa \\ \hline\end{array}, \begin{array}{|c|}\hline ab \\ ab \\ ab \\ \hline\end{array} \right\}$: the picture $\begin{array}{|c|}\hline aa \\ aa \\ aa \\ \hline\end{array}$ is in $\mathrm{ROW}(P) \boxplus \mathrm{COL}(P)$ but it is not in P.

Notice also that there are crosswords $P = L_1 \boxplus L_2$ that do not *saturate* their components, i.e., such that $ROW(P) \subsetneq L_1$ and/or $COL(P) \subsetneq L_2$. As an example, let $L_1 = \{0, 2\}, L_2 = \{0, 1\}$: then, $L_1 \boxplus L_2 = \{0\}$.

Another useful proposition follows.

Proposition 2. *Let $\mathcal{F}$ and $\mathcal{G}$ be two families of languages over an alphabet Σ, such that $\mathcal{F}$ is closed under intersection with languages of the form Σ'^*, for every $\Sigma' \subseteq \Sigma$, and $\mathcal{G}$ includes the finite language Σ.*
If $\mathcal{F}$ is a proper subfamily of $\mathcal{G}$ then $\mathbb{C}(\mathcal{F})$ is a proper subfamily of $\mathbb{C}(\mathcal{G})$.

Proof. Let $L \in \mathcal{G} \setminus \mathcal{F}$, with $L \subseteq \Sigma^+$. To prove that the containment of $\mathbb{C}(\mathcal{F}) \subseteq \mathbb{C}(\mathcal{G})$ is proper, consider the picture language in $\mathbb{C}(\mathcal{G})$: $L \boxplus \Sigma$ (i.e., a set of pictures with just one row). By contradiction, let $L \boxplus \Sigma$ be in $\mathbb{C}(\mathcal{F})$: there exist $L', L'' \in \mathcal{F}$ such that $L' \boxplus L'' = L \boxplus \Sigma$. Since $L'' \subseteq \Sigma$ (to have one-row pictures), then $L' \boxplus L''$ is the language $L' \cap (L'')^*$, which is equal to L and thus it is in $\mathcal{F}$, a contradiction with the assumption that $L \in \mathcal{G} \setminus \mathcal{F}$. □

3.2. Closure properties

In general, the crossword operation, without projection, over a family of word languages does not preserve closure over concatenations or union. It is easy to prove, using some elementary examples, the following proposition.

Proposition 3. *If a family $\mathcal{F}$ of languages includes the finite languages composed by words of length 1 and 2, then $\mathbb{C}(\mathcal{F})$ is not closed under union and row/column concatenations.*

Proof. We first prove the case for union. Let $L_1 = \{00, 01, 10, 11\}, L_2 = \{01\}$, $L_3 = \{01, 10\}$ be languages in $\mathcal{F}$. Define $L' = L_1 \boxplus L_2$, and $L'' = L_3 \boxplus L_3$. By contradiction, assume that $L' \cup L''$ is in $\mathbb{C}(\mathcal{F})$. The picture $p' = \begin{array}{|cc|}\hline 0 & 0 \\ 1 & 1 \\ \hline\end{array}$ is the only picture in L' and the picture $p'' = \begin{array}{|cc|}\hline 1 & 0 \\ 0 & 1 \\ \hline\end{array}$ is one of the two pictures in L'', the other one being $\begin{array}{|cc|}\hline 0 & 1 \\ 1 & 0 \\ \hline\end{array}$. Let $R = \mathrm{ROW}(p') \cup \mathrm{ROW}(p'') = \{00, 11, 10, 01\} = \mathrm{ROW}(L' \cup L'')$ and let $C = \mathrm{COL}(p') \cup \mathrm{COL}(p'') = \{01, 10\} = \mathrm{COL}(L' \cup L'')$. Consider the picture $p = \begin{array}{|cc|}\hline 1 & 1 \\ 0 & 0 \\ \hline\end{array}$, which is neither in L' nor in L'': $\mathrm{ROW}(p) = \{00, 11\} \subseteq R$ and $\mathrm{COL}(p) = \{10\} \subseteq C$. Then, p is in every RCC including both p' and p'', a contradiction.
We now consider row concatenation (the column concatenation case is symmetrical). Let $L_1 = \{0, 1\}, L_2 = \{01\}$ and $L_3 = \{10\}$ be in $\mathcal{F}$. Define $L' = L_1 \boxplus L_2$, and $L'' = L_1 \boxplus L_3$. By contradiction, assume that $L' \obar L''$ is a RCC. The picture $\begin{array}{|c|}\hline 0 \\ 1 \\ \hline\end{array}$ is in L', but not in L'', while the picture $\begin{array}{|c|}\hline 1 \\ 0 \\ \hline\end{array}$ is in L'', but not in L'. Hence, the row concatenation $L' \obar L''$ contains the picture $p_1 = \begin{array}{|cc|}\hline 0 & 1 \\ 1 & 0 \\ \hline\end{array}$ but not $p_2 = \begin{array}{|cc|}\hline 1 & 0 \\ 0 & 1 \\ \hline\end{array}$. However, $\mathrm{ROW}(p_1) = \mathrm{ROW}(p_2) = \{01, 10\}$ and $\mathrm{COL}(p_1) = \mathrm{COL}(p_2) = \{01, 10\}$, therefore the two pictures are indistinguishable by the crossword operation, i.e., also p_2 is in the RCC of $L' \obar L''$, a contradiction. □

On the other hand, the "disjoint" versions of the closures are preserved.

Disjoint closure operations The *disjoint union* and the *disjoint concatenation* operators are defined as the union and, respectively, the concatenation, of two languages over disjoint alphabets. Given a language $L \subseteq \Delta^*$, the *disjoint concatenation closure* of L is the language $(L \cdot L')^* \cdot L \cup (L \cdot L')^+$, where $L' \subseteq \Delta'^*$, where Δ' is a disjoint copy of Δ, and L' is the language isomorphic to L. These definitions can be immediately extended to 2D languages as well, by distinguishing column and row concatenations and their closure.

Proposition 4. *Let $\mathcal{F}$ be closed under disjoint union. The following closure properties of $\mathbb{C}(\mathcal{F})$ hold:*

1. *$\mathbb{C}(\mathcal{F})$ is closed under disjoint union;*
2. *if $\mathcal{F}$ is closed under disjoint concatenation then $\mathbb{C}(\mathcal{F})$ is closed under both row and column disjoint concatenations;*
3. *if $\mathcal{F}$ is closed under disjoint concatenation closure, then $\mathbb{C}(\mathcal{F})$ is closed under both column and row disjoint concatenation closures.*

Proof. Let $L \in \mathbb{C}(\mathcal{F})$ over an alphabet Δ; by definition, there exist two languages $L_H, L_V \subseteq \Delta^+$ in $\mathcal{F}$, such that $L = L_H \boxplus L_V$.

1. Let $L' \in \mathbb{C}(\mathcal{F})$ be a language over an alphabet Δ' with $\Delta \cap \Delta' = \emptyset$. Then, there exist $L'_H, L'_V \subseteq \Delta'^+$ such that $L' = L'_H \boxplus L'_V$. Since L, L' are over disjoint alphabets, $L \cup L' = (L_H \cup L'_H) \boxplus (L_V \cup L'_V)$.

2. Let L' be defined as in the previous case. Then,

$$L \oplus L' = (L_H \oplus L'_H) \boxplus (L_V \cup L'_V),$$

since Δ, Δ' are disjoint. The case $\ominus$ is symmetrical.
3. Let the languages H, V be the disjoint concatenation closure of L_H, L_V, respectively. Let the alphabet Δ' and the languages $L'_H \subseteq \Delta'^*, L'_V \subseteq \Delta'^*$ be as in the definition of disjoint concatenation closure. It is immediate to see that $L^{\oplus+} = H \boxplus (L_V \cup L'_V)$ and $L^{\ominus+} = (L_H \cup L'_H) \boxplus V$. $\square$

Closure properties of projective RCC The application of projection after the RCC operation, thus going from the family $\mathbb{C}(\mathcal{F})$ to the family $\mathbb{PC}(\mathcal{F})$, preserves many more closure properties of the family $\mathcal{F}$. For instance, as shown above, if $\mathcal{F}$ is closed under disjoint union, then $\mathbb{C}(F)$ is closed under disjoint union (Proposition 4), but $\mathbb{PC}(\mathcal{F})$ is closed also under non-disjoint union.

Another disjoint operation that is useful is the *disjoint finite substitution*. Given two alphabets Δ, Λ, it is a mapping $\rho : \Delta \to 2^{(\Lambda^+)}$ such that:

- for every $\delta \in \Delta$, $\rho(\delta)$ is a finite language, and
- for all $\delta_1, \delta_2 \in \Delta$, with $\delta_1 \neq \delta_2$, there exist two subsets Λ_1, Λ_2 of Λ such that $\Lambda_1 \cap \Lambda_2 = \emptyset$ and $\rho(\delta_1) \in 2^{(\Lambda_1^+)}$, $\rho(\delta_2) \in 2^{(\Lambda_2^+)}$.

For instance, strictly locally testable languages are closed under this disjoint operation, although if a language is k-SLT, the resulting language may be k'-SLT only for $k' > k$.

Theorem 2 *(disjoint operations). Let $\mathcal{F}$ be closed under disjoint union. The following closure properties hold:*

1. *$\mathbb{PC}(\mathcal{F})$ is closed under union;*
2. *if $\mathcal{F}$ is closed under disjoint concatenation then $\mathbb{PC}(\mathcal{F})$ is closed under both row and column concatenations;*
3. *if $\mathcal{F}$ is closed under disjoint concatenation closure, then $\mathbb{PC}(\mathcal{F})$ is closed under both column and row concatenation closures;*
4. *if $\mathcal{F}$ is closed under inverse alphabetic homomorphism and under intersection with local languages, then $\mathbb{PC}(\mathcal{F})$ is closed under intersection with the REC family;*
5. *if $\mathcal{F}$ is closed under disjoint finite substitution, then $\mathbb{PC}(\mathcal{F})$ is closed under picture homomorphism.*

Proof. The items 1, 2 and 3 follow immediately from Proposition 4, by applying suitable projections.
Let $L \in \mathbb{PC}(\mathcal{F})$ over an alphabet Σ; therefore, there exist an alphabet Δ, two languages $L_H, L_V \subseteq \Delta^+$ in $\mathcal{F}$ and a projection $h : \Delta \to \Sigma$ such that $L = h(L_H \boxplus L_V)$. We now prove separately the remaining items of the statement.

4. Consider $L \cap R$, where R is in REC. Therefore, there is an alphabet Θ such that R is defined by a projection $\pi : \Theta \to \Sigma$ of the RCC of two local languages LOC_H, LOC_V over Θ. Let Γ be the alphabet:

$$\{\langle \theta, \delta \rangle \mid \theta \in \Theta, \delta \in \Delta, \pi(\theta) = h(\delta)\},$$

i.e., Γ is the subset of the Cartesian product $\Theta \times \Delta$ such that the components of each element $\langle \theta, \delta \rangle$ have the same projection on Σ. Define $\upsilon_\Theta : \Gamma \to \Theta$ and $\upsilon_\Delta : \Gamma \to \Delta$ as the projection maps to Θ and Δ respectively.
Let $\widetilde{L}_H = \upsilon_\Delta^{-1}(L_H) \cap \upsilon_\Theta^{-1}(LOC_H) \cap \Gamma^+$ and let $\widetilde{L}_V = \upsilon_\Delta^{-1}(L_V) \cap \upsilon_\Theta^{-1}(LOC_V) \cap \Gamma^+$. The languages $\widetilde{L}_H$ and $\widetilde{L}_V$ are still in $\mathcal{F}$, since both $\mathcal{F}$ and the family of local languages are closed under inverse alphabetic homomorphism. Define the projection $\widetilde{h} : \Gamma \to \Sigma$ as $\widetilde{h}(\langle \theta, \delta \rangle) = h(\delta)$. Then, $L \cap R = \widetilde{h}\left(\widetilde{L}_H \boxplus \widetilde{L}_V\right)$.
5. Let $\varphi : \Sigma^{++} \to \Gamma^{++}$, where Γ is an alphabet, be a picture homomorphism. Let $m, n \geq 1$ be such that $\varphi(a) \in \Gamma^{n,m}$ for all $a \in \Sigma$: as already noticed, by Definition 2 the images by φ of the symbols of a picture must be isometric. Define $\Delta' = \Delta \times \{1, \dots, n\} \times \{1, \dots, m\}$. For every $1 \leq i \leq n$, define the word homomorphism:

$$\rho^i_H : \Delta^* \to (\Delta \times \{i\} \times \{1, \dots, m\})^* \text{ as}$$

$$\rho^i_H(\delta) = \langle \delta, i, 1 \rangle \langle \delta, i, 2 \rangle \dots \langle \delta, i, m \rangle, \text{ for all } \delta \in \Delta.$$

Let $\rho_H : \Delta \to 2^{(\Delta'^+)}$ be the substitution defined by $\rho_H(\delta) = \bigcup_{1 \leq i \leq n} \rho^i_H(\delta)$. It is a disjoint finite substitution, since each $\rho^i_H(\delta)$ returns a word over a different subset $\Delta \times \{i\} \times \{1, \dots, m\} \subset \Delta'$.
Symmetrically, for every $1 \leq j \leq m$ define the word homomorphism $\rho^j_V : \Delta^* \to (\Delta \times \{1, \dots, n\} \times \{j\})^*$ as $\rho^j_V(\delta) = \langle \delta, 1, j \rangle \dots \langle \delta, n, j \rangle$ and let the disjoint finite substitution $\rho_V : \Delta \to 2^{(\Delta'^+)}$ be the union of all ρ^j_V.
Let $L'_H = \rho_H(L_H), L'_V = \rho_V(L_V)$ and consider the picture language $L'_H \boxplus L'_V$. Given $\delta \in \Delta$, the language $\rho_H(\delta) \boxplus \rho_V(\delta)$ thus includes only the single picture shown below:

$$\rho_H(\delta) \boxplus \rho_V(\delta) = \left\{ \begin{array}{|cccc|} \langle \delta,1,1\rangle & \langle \delta,1,2\rangle & \dots & \langle \delta,1,m\rangle \\ \dots & \dots & \dots & \dots \\ \langle \delta,n,1\rangle & \langle \delta,n,2\rangle & \dots & \langle \delta,n,m\rangle \end{array} \right\}. \quad (3)$$

Therefore, every picture p' in $L'_H \boxplus L'_V$ is obtained from a picture p in $L_H \boxplus L_V$ by replacing every pixel $\delta \in \Delta$ of p with the subpicture in (3).
Define the projection $h' : \Delta' \to \Gamma$, for every $\langle \delta, i, j\rangle \in \Delta'$, as

$$h'(\langle \delta, i, j\rangle) = \varphi(h(\delta))_{(i,j)},$$

i.e., $h'(\langle \delta, i, j\rangle)$ is the pixel (i, j) of the picture $\varphi(h(\delta))$. For instance, if $h(\delta) = a$, the above subpicture (3) is projected by h' to the subpicture:

$$\begin{array}{|cccc|} \varphi(a)_{(1,1)} & \varphi(a)_{(1,2)} & \dots & \varphi(a)_{(1,m)} \\ \dots & \dots & \dots & \dots \\ \varphi(a)_{(n,1)} & \varphi(a)_{(n,2)} & \dots & \varphi(a)_{(n,m)} \end{array} = \varphi(a),$$

where $\varphi(a)_{(i,j)}$ is pixel (i, j) of the picture $\varphi(a)$. Therefore, $\varphi(L) = h'(L'_H \boxplus L'_V)$. □

4. Row-column combination of local and regular languages

The RCC of regular languages has received attention in the past since its alphabetic projection coincides with the family of recognizable languages, REC [3]. Some complexity issues for this case have been recently addressed in [5] where the RCCs of a regular language are called "regex crosswords".

We start from the subfamily of *local* languages. The RCC of local word languages is defined by 1×2 and 2×1 tiles, called *domino* tiles. It is folklore and easily proved that such a family is strictly included in the family defined by 2×2 tiles, a.k.a. local picture languages.

It is interesting to remark that even a "weak" class such as $\mathbb{C}(LOC)$ is already quite expressive, as shown by the following example.

Example 2. Let $\Sigma = \{0, 1, 2, 0', 1', 2'\}$, $L_1 = 0^*12^*2'^*1'0'^*$, $L_2 = 2^*10^* \cup 2'^*1'0'^*$. It is easy to see that both languages are local. Then, the crossword $L = L_1 \boxplus L_2$ consists of the row concatenation of two square pictures over Σ, with the left square having the letter 1 in the main diagonal, all 0's under the diagonal and all 2's above the diagonal Each row of the right square is the apostrophed mirror image of the same row of the left square, i.e., with $0', 1', 2'$ replacing $0, 1, 2$. Therefore, $ROW(L) = \{0^n 1 2^m 2'^m 1' 0'^n \mid n, m \geq 0\}$, which is CF. An example is the picture below, with the square subpictures divided by the dashed line:

1	2	2	2	2′	2′	2′	1′
0	1	2	2	2′	2′	1′	0′
0	0	1	2	2′	1′	0′	0′
0	0	0	1	1′	0′	0′	0′

The example can be generalized to the row concatenation of (any multiple of) four square pictures, by introducing a marked copy of Σ, denoted by $\hat{\Sigma} = \{\hat{0}, \hat{1}, \hat{2}, \hat{0}', \hat{1}', \hat{2}'\}$, and defining $\hat{L}_1, \hat{L}_2$ as the $\hat{}$-image of L_1, L_2. Then, define the picture language $L_{sq*} = (L_1 \cdot \hat{L}_1)^+ \boxplus (L_2 \cup \hat{L}_2)$. Both $(L_1 \cdot \hat{L}_1)^+$ and $L_2 \cup \hat{L}_2$ are local, but

$$ROW(L_{sq*}) = \{ \left(0^n 1 2^m 2'^m 1' 0'^n \hat{0}^n \hat{1} \hat{2}^m \hat{2}'^m \hat{1}' \hat{0}'^n\right)^+ \mid n, m \geq 0\}$$

is a context-sensitive but not CF, language whose Parikh image is non-semilinear.

The following statement appears to be new.

Corollary 1. *The family REC is closed under picture homomorphism.*

Proof. The thesis follows from Theorem 2, since REC is closed under disjoint finite substitutions. □

The following immediate properties describe the relations of REC with the RCC of well known subclasses of regular languages.

Lemma 1. *The picture language* $L_{\neg RCC} = (a^{++} \oplus b^{++}) \ominus b^{++}$ *is in* REC*, but it is not an RCC.*

Proof. $L_{\neg RCC}$ is obviously in REC. Both the row and column languages of $L_{\neg RCC}$ are a^*b^+. Therefore, by Proposition 1, if the language were an RCC then also a picture of the form b^{++} would be in $L_{\neg RCC}$, a contradiction. □

Proposition 5. $\mathbb{C}(LOC) \subsetneq \mathbb{C}(SLT) \subsetneq \mathbb{C}(REG) \subsetneq REC$.

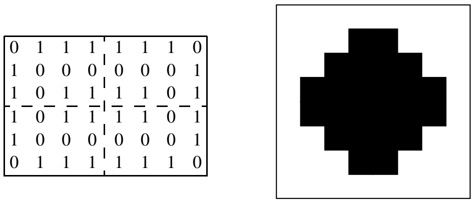

Fig. 1. Two pictures for Example 3: left, a HV-palindrome, instance of language P_{SP}; right, a HV-convex polyomino, instance of language P_{HV}.

Proof. All inclusions, apart from the rightmost, derive from Proposition 2. The strict inclusion $\mathbb{C}(REG) \subsetneq REC$ derives from Lemma 1. □

As an example of Proposition 5, consider the RCC $L = 0^*10^* \boxplus (0 \cup 1)^*$, which is in $\mathbb{C}(REG)$: the two (single row) pictures 010 and 01010 have the same domino tiles, but the former is in L and the latter is not, therefore L is not in $\mathbb{C}(LOC)$.

As a last remark on the relationship between families, we notice that the strict inclusions holding for the non-projective crosswords of certain families included in REG become now identities because the projection of their RCC families coincides with the REC languages [3,7], in formulae:

$$\mathbb{P}\mathbb{C}(LOC) = \mathbb{P}\mathbb{C}(REG) = REC.$$

5. RCC of context-free languages

To our knowledge, the crosswords of CF languages have never been studied after their definition in [3]. We show two suggestive examples of CD crosswords and we compare them with other families of picture languages.

Proposition 6. *Let $L_{pal} = \{ww^R \mid w \in \{0,1\}^+\}$. The picture language $P = L_{pal} \boxplus \{0,1\}^*$ is in $\mathbb{C}(CF)$ but it is not in REC.*

Proof. By contradiction, if P is in REC, then $P' = P \cap \left((0 \cup 1)^{*\obar}\right)$ is in REC as well, by closure of REC under intersection. P' is just the one-row picture language $L_{pal} \boxplus \{0,1\}$, which cannot be in REC since one-row languages in REC must be regular (they are the image under projection of a local language). □

Example 3 *(palindromic symmetries).* We show that the so-called 2D *horizontal-vertical (HV) palindromes* studied in [20] are in $\mathbb{C}(CF)$. Let $L_{pal} = \{ww^R \mid w \in \{0,1\}^+\}$. We define the language of HV-palindromes as $P_{SP} = L_{pal} \boxplus L_{pal}$. This language is not in REC, otherwise also the two-row language $P_{SP} \cap (\{0,1\} \ominus \{0,1\})^{*\obar}$ would be in REC, contradicting (essentially) Proposition 6.

To illustrate, every picture in P_{SP} can be subdivided in four subpictures P_i, $1 \le i \le 4$, of the same size. Fig. 1, left, shows an example where the four subpictures P_i are separated by dashed lines. The language P_{SP} can also be defined as:

$$\begin{aligned} P_{SP} = \; & (P_1 \obar P_2) \ominus (P_3 \obar P_4) \text{ such that } ROW(P_1 \obar P_2) \in L_{pal} \\ & \text{and } COL(P_1 \ominus P_3) \in L_{pal} \text{ and } ROW(P_3 \obar P_4) \in L_{pal}. \end{aligned}$$

Notice that necessarily $COL(P_2 \ominus P_4) \in L_{pal}$.
Another interesting palindromic language consists of the pictures containing a HV-convex polyomino [24], symmetrical both horizontally and vertically, drawn in color black, and exactly centered in a white background. Encoding black or white pixels by 1 or 0, respectively, such a language, denoted by P_{HV}, is defined as the RCC of the language $L_{HV} = \{0^n 1^+ 0^n \mid n \ge 0\}$ with itself, namely $L_{HV} \boxplus L_{HV}$. An example is in Fig. 1, right.

5.1. Basic $\mathbb{P}\mathbb{C}(CF)$ properties

The next result is immediate from Theorem 2.

Corollary 2. *The family $\mathbb{P}\mathbb{C}(CF)$ is closed under union, row/column concatenations and their closures, under inverse alphabetic homomorphism, under intersection with REC languages and under picture homomorphism.*

The following properties are also immediate.

Corollary 3 *(inclusion relations).*

1. $\mathbb{C}(REG) \subsetneq \mathbb{C}(CF)$.
2. *$\mathbb{C}(CF)$ is incomparable with REC.*
3. $REC = \mathbb{P}\mathbb{C}(REG) \subsetneq \mathbb{P}\mathbb{C}(CF)$.

Proof. Following Proposition 6, we just prove that REC is not included in $\mathbb{C}(CF)$. By Lemma 1, the language $L_{\neg RCC} = (a^{++} \oplus b^{++}) \ominus b^{++}$ is not an RCC, hence it cannot be in $\mathbb{C}(CF)$, but it is in REC. Then, since $REC = \mathbb{PC}(REG)$ it is obvious that $REC \subsetneq \mathbb{PC}(CF)$; the inclusion is strict by considering any picture language with just one row where a CF, but not regular, language, is placed. □

Theorem 3. *The membership problem for the family* $\mathbb{PC}(CF)$ *is NP-complete.*

Proof. To prove the problem is in NP, consider an algorithm that, given two CF languages L_1, L_2, a projection h and a picture p, guesses the correct inverse projection $h^{-1}(p)$ (i.e., the pre-image) and then verifies if it is in $L_1 \boxplus L_2$ (which obviously takes polynomial time). The problem is NP-hard, since $\mathbb{PC}(CF)$ includes the class REC for which NP-hardness of membership testing is known. □

Example 4. We show an example of a language in $\mathbb{PC}(CF)$ but not in $\mathbb{C}(CF)$. Let R_{sq} be the language of square pictures over alphabet $\{0, 1\}$, which is a classical REC language. Let $P_{4sq} = P_{SP} \cap R_{sq}$, where P_{SP} is the HV-palindrome language of Example 3. By closure under intersection with REC, as stated in Corollary 2, it follows that P_{4sq} is in $\mathbb{PC}(CF)$. We observe that P_{4sq} is derived from the language P_{SP} of Example 3, with the constraint that the four rectangles in which each picture is divided are indeed squares. To see that the language P_{4sq} is not in $\mathbb{C}(CF)$, assume that P_{4sq} is an RCC. Consider a picture in P_{4sq} where each pixel is the letter 0: thus all words in $(00)^+$ are both in $ROW(P_{4sq}), COL(P_{4sq})$. Therefore, by Proposition 1, also non-square pictures, with even height and width and composed only of symbol 0, must be in P_{4sq}, a contradiction.

5.2. *Comparison with families of grammar-based picture languages*

Various definitions of *context-free picture languages* have been proposed in the past (for a survey see [4]) that extend the formalism of context-free grammar productions into some sort of picture generating productions. Of course, the RCC operation is yet another approach to move from CF words to pictures, but unlike the past ones it does not construct any sort of 2D grammar productions.

Here we compare the $\mathbb{C}(CF)$ family with three notable grammar-based cases, namely Siromoney [2] context-free Kolam Grammars (KG) equivalent to Matz [25] grammars, Průša [26] grammars (PG), and Tiling Grammars [27] (TG). We denote with $\mathcal{L}(KG)$, $\mathcal{L}(PG)$, $\mathcal{L}(TG)$ the corresponding language families. The following inclusions are known [4]:

$$\mathcal{L}(KG) \subsetneq \mathcal{L}(PG) \subsetneq \mathcal{L}(TG), \quad REC \subsetneq \mathcal{L}(TG)$$

while $\mathcal{L}(KG)$ and $\mathcal{L}(PG)$ are incomparable with REC.

Theorem 4 *($\mathbb{C}(CF)$ vs grammar-based picture families).*

1. *$\mathbb{C}(CF)$ is incomparable with both $\mathcal{L}(KG)$ and $\mathcal{L}(PG)$.*
2. *$\mathcal{L}(TG)$ is not included in $\mathbb{C}(CF)$.*

Proof. We prove Part (1). To prove $\mathcal{L}(KG) \not\subseteq \mathbb{C}(CF)$, hence also $\mathcal{L}(PG) \not\subseteq \mathbb{C}(CF)$, consider the subset of language $L_{\neg RCC} = (a^{++} \oplus b^{++}) \ominus b^{++}$ of Lemma 1, which is not in $\mathbb{C}(CF)$, such that it only contains two-row pictures. It is immediate to see that also such a language, denoted by $L'_{\neg RCC}$, is not in $\mathbb{C}(CF)$ and it is generated by the following Kolam grammar (using the Matz simplified notation for column and row concatenation):

$$S \to \begin{pmatrix} X \\ B \end{pmatrix}, \; X \to aX \mid aB, \; B \to bB \mid b.$$

To prove $\mathbb{C}(CF) \not\subseteq \mathcal{L}(KG)$, consider the two-row picture language defined as $P = L_1 \boxplus L_2$, where

$$L_1 = \left(\{c^+ww^R \mid w \in \{a,b\}^+\}\right) \cup \left(\{ww^Rc^+ \mid w \in \{a,b\}^+\}\right)$$

and L_2 is the set of length two words over $\{a, b, c\}$ excluding cc. Hence, in P there is no occurrence of the tile $c \ominus c$. P is known not to be in $\mathcal{L}(KG)$ [4].
Part (2) follows from $\mathcal{L}(PG) \subsetneq \mathcal{L}(TG)$. □

The incomparability results of Theorem 4 change if we consider the projective RCC of CF languages. For instance, the language $L'_{\neg RCC}$ in the proof of that theorem is in $\mathbb{PC}(CF)$ indeed. But a precise comparison of $\mathbb{PC}(CF)$ with the grammar-based families would require a fine analysis of the possibility to simulate quite different picture generating devices, and is out-of-scope.

6. 2D Dyck languages

The family of context-free word languages is characterized by the Chomsky-Schützenberger Theorem that was briefly recalled in Sect. 2.1, where we anticipated our use of the non-erasing variant [18] whose alphabet is independent of the language.

An interesting question is whether a similar characterization for the families $\mathbb{C}(CF)$ and $\mathbb{PC}(CF)$ is possible, saying that any CF crossword is the projection of a "2D Dyck language" intersected with a strictly locally testable picture language. This of course requires a suitable definition of 2D-Dyck languages. But the naive choice of "2D Dyck" as the RCC of two Dyck languages does not work. For instance, the RCC of two Dyck languages D_1, D_2 over the alphabet $\{a, a'\}$ is empty [19] (e.g., if the first row of the RCC is aa', it is impossible to place a Dyck word in both the first and the second column), hence a CF crossword cannot be the homomorphic image of this RCC.

This motivates the introduction of a less constrained crossword operation.

6.1. Cartesian crosswords

We introduce a new type of row-column combination, called *Cartesian crossword* (or Cartesian RCC), that defines a non-empty language also for two languages with different alphabets; it creates a picture on a common alphabet that is the Cartesian product of the row and column alphabets. Therefore, each symbol is a pair (row-symbol, column-symbol) and it is natural to define the resulting picture in such a way that its pixel by pixel projection on the row-symbols produces a picture whose rows belong to the first language; similarly for the projection on the column symbols.

Definition 4 *(Cartesian crossword).* Let Σ_1, Σ_2 be two (possibly not distinct) alphabets and let $h_1 : \Sigma_1 \times \Sigma_2 \to \Sigma_1$ and $h_2 : \Sigma_1 \times \Sigma_2 \to \Sigma_2$ be the projection maps from the set $\Sigma_1 \times \Sigma_2$, called the *Cartesian alphabet*, to the first and to the second component, respectively. Given two word languages $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$, their *Cartesian crossword* (CCW), denoted as $L_1 \boxtimes L_2$, is the picture language $P \subseteq (\Sigma_1 \times \Sigma_2)^{++}$ defined as:

$$P = h_1^{-1}(L_1) \boxplus h_2^{-1}(L_2).$$

A picture language is called a CCW if it is the Cartesian crossword of two word languages. Let $\mathcal{F}$ be a family of word languages. The *Cartesian crossword* of $\mathcal{F}$, denoted as $\mathbb{C}^C(\mathcal{F})$, is the family of picture languages of the form $L_1 \boxtimes L_2$, for all languages L_1, L_2 of $\mathcal{F}$.

Properties of Cartesian crosswords From the definition we have:

$$h_1(ROW(L_1 \boxtimes L_2)) = L_1 \quad \text{and} \quad h_2(COL(L_1 \boxtimes L_2)) = L_2. \tag{4}$$

Hence, the Cartesian crossword of two non-empty languages is not empty and every word of L_1 and L_2 contributes to define some pictures. The following statement is immediate from Definition 4.

Proposition 7. *If a family $\mathcal{F}$ of word languages is closed under inverse alphabetic homomorphism, then $\mathbb{C}^C(\mathcal{F}) \subseteq \mathbb{C}(\mathcal{F})$.*

This inclusion clearly holds for the language families LOC, SLT, REG and CF.

The following Lemma, whose proof is immediate from the definition of Cartesian RCC, is needed to prove Theorem 5.

Lemma 2. *Let $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$.*

1. *If $L_3 \subseteq \Sigma_1^*$ and $L_4 \subseteq \Sigma_2^*$, then*

$$(L_1 \boxtimes L_2) \cap (L_3 \boxtimes L_4) = (L_1 \cap L_3) \boxtimes (L_2 \cap L_4).$$

2. *If $\Sigma_1 = \Sigma_2 = \Sigma$, then $L_1 \boxplus L_2 = h((L_1 \boxtimes L_2) \cap G^{++})$, where G is the alphabet $\{(a, a) \mid a \in \Sigma\}$ and $h : G \to \Sigma$ is the projection defined, for all $a \in \Sigma$, as $h((a, a)) = a$.*

6.2. Cartesian crossword of Dyck languages

We now apply the new definition to Dyck languages.

Definition 5 *(2D-Dyck language).* Let D_1, D_2 be two Dyck languages defined respectively over the Dyck alphabets Δ_1 and Δ_2. The alphabet $\Gamma = \Delta_1 \times \Delta_2$ is called a *2D-Dyck alphabet*. The Cartesian crossword $D_1 \boxtimes D_2$ is called a *2D-Dyck language* over Γ.

Example 5. We define a 2D-Dyck language as $D_1 \boxtimes D_2$ where:

- $D_1 \subseteq \Delta_1^*$, $\Delta_1 = \{a, a', b, b'\}$ with coupling $\{(a, a'), (b, b')\}$;
- $D_2 \subseteq \Delta_2^*$, $\Delta_2 = \{a, a', b, b', c, c', d, d'\}$ with coupling $\{(a, a'), (b, b'), (c, c'), (d, d')\}$.

An example of a picture in $(D_1 \boxtimes D_2)$ is in Fig. 2, left. For clarity, the picture in Fig. 2, right, associates also a corner symbol to each pair of letters in the row-column alphabet, as shown in the following table:

ac	$a'b$	bd	$b'a$	aa	$a'b$
ac	aa	aa	$a'b$	$a'b$	$a'd$
ac'	ba'	ba'	$b'b'$	$b'b'$	$a'd'$
ac'	ab'	$a'd'$	$a'a'$	ba'	$b'b'$

┌ ac	┐ $\mathbf{a'b}$	┌ bd	┐ $b'a$	┌ aa	┐ $\mathbf{a'b}$
┌ ac	┌ aa	┌ aa	┐ $a'b$	┐ $a'b$	┐ $a'd$
└ ac'	└ ba'	└ ba'	┘ $b'b'$	┘ $b'b'$	┘ $a'd'$
└ ac'	└ $\mathbf{ab'}$	┘ $a'd'$	┘ $a'a'$	└ ba'	┘ $\mathbf{b'b'}$

Fig. 2. (Left) A picture of the 2D-Dyck language of Example 5. (Right) The same picture with corner symbols.

corner	row	col	example
┌	open	open	(a, a)
└	open	closed	(a, a')
┐	closed	open	(a', a)
┘	closed	closed	(a', a')

We highlight in bold an instance of two coupled pairs violating the Dyck condition over the alphabet Γ: (a', b) is matching vertically with both (a, b') and (b', b'). Thus, the column language over alphabet Γ is not Dyck, but it is a very simple case of the Input-Driven languages [28], later renamed Visibly Pushdown languages [29],

7. A Chomsky-Schützenberger theorem for RCC with projection

In this section we establish our main result: a homomorphic characterization of $\mathbb{PC}(CF)$ languages based on 2D-Dyck languages. As a first step, we prove a homomorphic representation theorem for the RCC of CF languages.

7.1. A representation theorem for $\mathbb{C}(CF)$

We show that any CF crossword can be represented as the projection of the intersection of a 2D-Dyck language and a $\mathbb{C}(SLT)$ language. We assume that all pictures have even height and width, until Sect. 7.2.1 where the restriction is lifted. To this goal, we denote with CF_{even} the class of context-free languages whose words may only have even length.

Theorem 5 *(representation theorem). Let $P \subseteq \Sigma^{++}$ be a picture language in $\mathbb{C}(CF_{even})$. Then there exist a 2D-Dyck alphabet Γ, a 2D-Dyck language $D_{\boxtimes}$, a $\mathbb{C}(SLT)$ language R, both over alphabet Γ, and a projection $h : \Gamma \to \Sigma$ such that $P = h(D_{\boxtimes} \cap R)$.*

Proof. Let $P = L_1 \boxplus L_2$, where $L_1, L_2 \subseteq \Sigma^*$ are context-free languages.
By applying Theorem 1 to L_1, L_2, we have that $L_1 = h_1(D_1 \cap R_1)$ and $L_2 = h_2(D_2 \cap R_2)$, where: Δ is a Dyck alphabet, $D_1, D_2 \subseteq \Delta^*$ are Dyck languages, R_1, $R_2 \subseteq \Delta^*$ are two SLT languages, and $h_1 : \Delta \to \Sigma$, $h_2 : \Delta \to \Sigma$ are two projections.
By Lemma 2, Part (2),

$$L_1 \boxplus L_2 = \pi\left((L_1 \boxtimes L_2) \cap G^{++}\right) \tag{5}$$

where $G = \{(a, a) \mid a \in \Sigma\}$ and $\pi : G \to \Sigma$ is such that $\pi((a, a)) = a$.

Let $\Gamma = \Delta \times \Delta$. Let ρ be the projection $\rho : \Gamma \to \Sigma \times \Sigma$ defined by $\rho(\langle\alpha, \beta\rangle) = \langle h_1(\alpha), h_2(\beta)\rangle$.

Therefore:

$$L_1 \boxtimes L_2 = h_1(D_1 \cap R_1) \boxtimes h_2(D_2 \cap R_2)) = \rho\left((D_1 \cap R_1) \boxtimes (D_2 \cap R_2))\right).$$

Thus, by Lemma 2, Part (1):

$$L_1 \boxtimes L_2 = \rho\left((D_1 \boxtimes D_2) \cap (R_1 \boxtimes R_2)\right). \tag{6}$$

From (5) and (6), it follows:

$$L_1 \boxplus L_2 = \pi\left(G^{++} \cap \rho\left((D_1 \boxtimes D_2) \cap (R_1 \boxtimes R_2)\right)\right). \tag{7}$$

Define the subset F of Γ as:

$$F = \{(\alpha, \beta) \mid h_1(\alpha) = h_2(\beta)\}. \tag{8}$$

The picture language F^{++} is in $\mathbb{C}(SLT)$, since $F^{++} = F^+ \boxplus F^+$.

For every picture language $W \subseteq \Gamma^{++}$, we have:

$$\rho(W) \cap G^{++} = \rho(W \cap F^{++}). \tag{9}$$

Therefore, combining (7) and (9), we have:

$$L_1 \boxplus L_2 = \pi(\rho\left((D_1 \boxtimes D_2) \cap (R_1 \boxtimes R_2) \cap F^{++}\right)). \tag{10}$$

We notice that, since the word language family SLT is closed under inverse alphabetic homomorphism, by Proposition 7, it follows that $R_1 \boxtimes R_2 \in \mathbb{C}(SLT)$.

Picture:

0	1	1	0	0	1
0	0	0	1	1	1
0	1	1	0	0	1
0	0	1	1	1	0

Pre-image:

⌜ ac	⌝ $a'b$	⌜ bd	⌝ $b'a$	⌜ aa	⌝ $a'b$
⌜ ac	⌜ aa	⌜ aa	⌝ $a'b$	⌝ $a'b$	⌝ $a'd$
⌞ ac'	⌞ ba'	⌞ ba'	⌟ $b'b'$	⌟ $b'b'$	⌟ $a'd'$
⌞ ac'	⌞ ab'	⌟ $a'd'$	⌟ $a'a'$	⌞ ba'	⌟ $b'b'$

Fig. 3. A picture and its pre-image for Example 6.

The statement of the theorem follows if we let $D_{\boxtimes}$ be the 2D-Dyck language $D_1 \boxtimes D_2$, we let h be the projection composition $\pi \circ \rho$, and we let $R = (R_1 \boxtimes R_2) \cap F^{++}$; hence, $L_1 \boxplus L_2 = h\left((D_1 \boxtimes D_2) \cap R\right)$. By closure of $\mathbb{C}(SLT)$ under intersection, also R is in $\mathbb{C}(SLT)$. $\square$

We illustrate the construction of the proof of Theorem 5.

Example 6. Let $\Sigma = \{0, 1\}$ and let $L_1 = \{x \in \Sigma^+ \mid \#_0(x) = \#_1(x)\}$, i.e., L_1 is the set of words with the same number of 0 and 1, defined for instance by the CF grammar: $S \to 0S1S \mid 1S0S \mid \varepsilon$.

Let $L_2 = (00)^+ \cup \{x \in (\Sigma^2)^+ \mid \#_1(x) \geq \#_0(x)\}$, defined for instance by the following grammar: $S \to A \mid B$, $A \to 0A0A \mid \varepsilon$, $B \to 0B1B \mid 1B0B \mid 1B1B \mid \varepsilon$.

The alphabets Δ_1, Δ_2, $\Gamma = \Delta_1 \times \Delta_2$, and the Dyck languages D_1, D_2 are the same of Example 5. Define:

- The projections $h_1, h_2 : \{a, b'\} \mapsto 0, \{a', b\} \mapsto 1$ and also $h_2 : \{c, c'\} \mapsto 0, \{d, d'\} \mapsto 1$.
- The language $R_1 = \{a, a', b, b'\}^*$ is the universal language; R_2 is the local language defined by the following forbidden factors of length 2:

  $$\mathbb{FF} = \left\{xy, yx \mid x \in \{a, a', b, b', d, d'\},\ y \in \{c, c'\}\right\},$$

  i.e., the letters c and c' may not occur next to one of the other letters in Δ_2.
  The equality $L_1 = h_1(D_1 \cap R_1)$ follows immediately since the Dyck language D_1 can be defined by the grammar $S \to aSa'S \mid bSb'S \mid \varepsilon$ and the projection h_1 correctly maps the Dyck letters to 0,1. The same reasoning can be used to prove the equality $L_2 = h_2(D_2 \cap R_2)$.
- The projection $h : \Gamma \to \Sigma$ is defined as 0, respectively 1, for the following sets Γ_0, Γ_1:

  $$\Gamma_0 = \left\{(x, y) \mid x \in \{a, b'\}, y \in \{a, b', c, c'\}\right\}$$
  $$\Gamma_1 = \left\{(x, y) \mid x \in \{a', b\}, y \in \{a', b, d, d'\}\right\}.$$
- The set F–defined in Eq. (8)–is just $\Gamma_0 \cup \Gamma_1$.

An example of a picture and its pre-image in $(D_1 \boxtimes D_2) \cap (R_1 \boxtimes R_2) \cap F^{++}$ is in Fig. 3. The pre-image was already shown in Fig. 2, right. We also add the corner symbols for clarity.

Notice that the inverse of Theorem 5 does not hold, i.e., a language of the form $h(D_{\boxtimes} \cap R)$ is not necessarily a CF crossword. For instance, the language $L_{\neg RCC} = (a^{++} \oplus b^{++}) \ominus b^{++}$ of Lemma 1 is not in $\mathbb{C}(CF)$, but it is in REC hence also in $\mathbb{PC}(REG) \subseteq \mathbb{PC}(CF)$; therefore, it can be defined as the projection of a $\mathbb{C}(CF)$ language. By Theorem 5 and composing two projections, it has the form $h(D_{\boxtimes} \cap R)$. The complete characterization *à la* Chomsky-Schützenberger requires the addition of the projection operation and is presented in the next section.

7.2. A Chomsky-Schützenberger Theorem for $\mathbb{PC}$(CF)

We move to the projective CF crosswords and state a complete C-S Theorem, that was not possible without projection. As for Theorem 5, we first prove the theorem under the restrictive hypothesis that the pictures height and width are even. Then in Theorem 7 we drop the restriction.

Theorem 6 *(even case). A picture language $P \subseteq \Sigma^{++}$, is in $\mathbb{PC}(CF_{even})$ if, and only if, there exist a 2D-Dyck alphabet Γ, a 2D-Dyck language $D_{\boxtimes}$ over Γ, a $\mathbb{C}(SLT)$ language $R \subseteq \Gamma^+$ and a projection $h : \Gamma \to \Sigma$ such that $P = h(D_{\boxtimes} \cap R)$.*

Proof. Since P can be obtained applying a projection to a language in $\mathbb{C}(CF)$ and the composition of two projections is still a projection, by Theorem 5 it follows that $P = h(D_{\boxtimes} \cap R)$ as in the statement of this theorem.

It remains to show that any language of the form $h(D_{\boxtimes} \cap R)$ is in $\mathbb{PC}(CF)$. The language $D_{\boxtimes}$ is in $\mathbb{C}(CF)$, since by definition it is equal to $D_1 \boxtimes D_2$ for some Dyck word languages D_1, D_2. Hence it is also in $\mathbb{PC}(CF)$. Since $R \in REC$ and $D_{\boxtimes} \in \mathbb{PC}(CF)$, by closure of $\mathbb{PC}(CF)$ under intersection with REC (Corollary 2) and under projection, we have that the language $h(D_{\boxtimes} \cap R)$ is also in $\mathbb{PC}(CF)$. $\square$

Picture:

0	1	1	0	0	1	**0**
0	0	0	1	1	1	**1**
0	1	1	0	0	1	**0**
0	0	1	1	1	0	**1**
0	**1**	**1**	**0**	**0**	**1**	**0**

Pre-image:

ac	$a'b$	bd	$b'a$	aa	$a'b$	$\delta_0 a$
ac	aa	aa	$a'b$	$a'b$	$a'd$	$\delta_1 b$
ac'	ba'	ba'	$b'b'$	$b'b'$	$a'd'$	$\delta_0 b'$
ac'	ab'	$a'd'$	$a'a'$	ba'	$b'b'$	$\delta_1 a'$
$a\delta_0$	$a'\delta_1$	$b\delta_1$	$b'\delta_0$	$a'\delta_0$	$a'\delta_1$	$\delta_0\delta_0$

Fig. 4. (Left) Example of a picture with odd height and width, in the language P' of Example 7; (right) its pre-image in the 2D-Dyck language.

7.2.1. Homomorphic characterization for pictures of arbitrary size

We drop the restriction that all pictures must have even height and width. It suffices to revisit Theorem 5 for $\mathbb{C}(CF)$, since projections do not change the picture size. As anticipated in Section 2, we extend the Dyck alphabet with new symbols, called neutral, that are not coupled.

Replace in Theorem 5 the Dyck word languages, used in the proof for the characterization of row/column languages, with the Dyck languages with neutral symbols of Theorem 1. A 2D-Dyck alphabet with n neutral symbols is $\Gamma = \Delta_1 \times \Delta_2$, where Δ_1, Δ_2 are Dyck alphabets with n neutral symbols. A 2D-Dyck language over Γ is the picture language $D_{\boxtimes} = D_1 \boxtimes D_2$ where D_1, D_2 are Dyck languages over Δ_1, Δ_2. Thus, we immediately have:

Theorem 7 *(main). A picture language $P \subseteq \Sigma^{++}$ is in $\mathbb{PC}(CF)$ if, and only if, there exist a 2D-Dyck alphabet Γ, with $|\Sigma|$ neutral symbols, a 2D-Dyck language $D_{\boxtimes}$ over Γ, a $\mathbb{C}(SLT)$ language $R \subseteq \Gamma^{+}$ and a projection $h : \Gamma \to \Sigma$ such that $P = h(D_{\boxtimes} \cap R)$.*

Example 7. We consider a language P' allowing also pictures with odd sizes. The row language is $L'_1 = \{x \in \{0,1\}^+ : |\#_0(x) - \#_1(x)| \le 1\}$ and the column language is $L'_2 = L_2 \cdot \{0,1\}$, where L_2 is the language of Example 6.

Then, for $i = 1,2$, $L'_i = h'_i(D'_i \cap R'_1)$, where D'_i is obtained from the Dyck language $D_i \subseteq \Delta_i^{++}$ of Example 6 by allowing also two neutral symbols δ_0, δ_1; the projection h'_i is equal to h_i over Δ_i, with $h'_i(\delta_0) = 0$, $h'_i(\delta_1) = 1$; the regular language R'_i is obtained from R_i by adding δ_0 or δ_1 at the end of words of even length. An example of picture in $P' = L'_1 \boxplus L'_2$, obtained by adding one row and one column (in bold) to the picture of Fig. 3, is in Fig. 4 (left), together with its pre-image (right).

8. Conclusion

In this paper we have advanced in the study of row-column combinations in two cases, the regular languages (with the strictly locally testable subfamily) and chiefly the context-free languages. The RCC operation is a very simple constructor of pictures and a popular one for crossword games. A commonality in both cases $\mathbb{C}(REG)$ and $\mathbb{C}(CF)$ is that the resulting family has such as poor closure properties, but it gains many more formal properties when the RCC is followed by projection. In fact $\mathbb{PC}(REG)$ is the family REC of tiling system recognizable languages, well-known for preserving quite a few properties of regular languages.

But an important conceptual difference separates the $\mathbb{PC}(REG)$ and $\mathbb{PC}(CF)$ families. The former is officially defined by a homomorphic characterization, i.e., as the projection of pictures tiled with two-by-two tiles. Such a definition is acknowledged as the transposition in 2D of the Medvedev's theorem that homomorphically characterizes the regular languages starting from local ones. Since the Chomsky-Schützenberger Theorem is widely viewed as the counterpart for CF languages, of the Medvedev's theorem, a natural idea is to investigate whether the RCC of CF languages can be characterized by a C-S Theorem extension in 2D, much as REC is defined by the 2D extension of Medvedev's theorem. Such a characterization of $\mathbb{PC}(CF)$ was unknown until now, and it is our major contribution.

All the three entities in C-S Theorem, namely a Dyck language, a 2-strictly locally testable language and a homomorphism, have been suitably redefined in 2D. The redefinition of the homomorphism and of the 2-SLT entities is quite natural: a letter-to-letter homomorphism and a strictly locally testable picture language, respectively. But what should be the essence of a Dyck language in 2D was unclear, and several possibilities having intuitive appeal were investigated in [19], none unfortunately being adequate for stating a C-S Theorem in 2D. By replacing the standard RCC operation with the Cartesian RCC, resulted in the Definition 5 of the 2D-Dyck language, that acts as generator in the homomorphic characterization of the $\mathbb{PC}(CF)$ family.

Now, this family possesses two equivalent definitions: one uses the row-column combination of CF words, and the other selects the relevant 2D-Dyck structures by means of the intersection with strictly locally testable picture languages, and then projects them on the terminal alphabet. In our opinion, such a new homomorphic definition may play the role of the grammar-based one for CF languages, with the accompanying theory of syntax structures. This may hopefully open a new direction of research on the syntactic structure of $\mathbb{PC}(CF)$ pictures.

CRediT authorship contribution statement

Stefano Crespi Reghizzi: Writing – review & editing, Investigation. **Antonio Restivo:** Writing – review & editing, Investigation. **Pierluigi San Pietro:** Writing – review & editing, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We thank the anonymous reviewers for their valuable suggestions.

References

[1] A. Rosenfeld, Picture Languages: Formal Models for Picture Recognition, Academic Press, 1979.
[2] R. Siromoney, Advances in array languages, in: Graph-Grammars and Their Application to Computer Science, 3rd Int, Workshop, Warrenton, Virginia, USA, 1986, in: LNCS, vol. 291, Springer, 1986, pp. 549–563.
[3] D. Giammarresi, A. Restivo, Two-dimensional languages, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, vol. 3, Springer, 1997, pp. 215–267.
[4] S. Crespi Reghizzi, D. Giammarresi, V. Lonati, Two-dimensional models, in: J. Pin (Ed.), Handbook of Automata Theory, European Mathematical Society Publishing House, 2021, pp. 303–333.
[5] S.A. Fenner, D. Padé, T. Thierauf, The complexity of regex crosswords, Inf. Comput. 286 (2022) 1–18.
[6] D. Giammarresi, A. Restivo, Recognizable picture languages, Int. J. Pattern Recognit. Artif. Intell. 6 (2–3) (1992) 241–256.
[7] M. Latteux, D. Simplot, Recognizable picture languages and domino tiling, Theor. Comput. Sci. 178 (1–2) (1997) 275–283.
[8] M. Latteux, D. Simplot, Context-sensitive string languages and recognizable picture languages, Inf. Comput. 138 (2) (1997) 160–169.
[9] M. Anselmo, D. Giammarresi, M. Madonia, Classification of string languages via tiling recognizable picture languages, in: A. Dediu, S. Inenaga, C. Martín-Vide (Eds.), LATA 2011, in: Lecture Notes in Computer Science, Springer, Berlin, 2011, pp. 105–116.
[10] T.J. Smith, K. Salomaa, Recognition and complexity results for projection languages of two-dimensional automata, J. Autom. Lang. Comb. 28 (1–3) (2023) 201–220.
[11] F. Mignosi, A. Restivo, P.V. Silva, On fine and Wilf's theorem for bidimensional words, Theor. Comput. Sci. 292 (1) (2003) 245–262.
[12] G. Gamard, G. Richomme, J. Shallit, T.J. Smith, Periodicity in rectangular arrays, Inf. Process. Lett. 118 (2017) 58–63.
[13] K. Inoue, I. Takanami, A characterization of recognizable picture languages, in: A. Nakamura, M. Nivat, A. Saoudi, P.S.P. Wang, K. Inoue (Eds.), Parallel Image Analysis, Springer, Berlin, 1992, pp. 133–143.
[14] D. Giammarresi, A. Restivo, S. Seibert, W. Thomas, Monadic second-order logic over rectangular pictures and recognizability by tiling systems, Inf. Comput. 125 (1) (1996) 32–45.
[15] N. Chomsky, M. Schützenberger, The algebraic theory of context-free languages, in: P. Brafford, D. Hirschenber (Eds.), Computer Programming and Formal Systems, North-Holland, Amsterdam, 1963, pp. 118–161.
[16] J. Berstel, L. Boasson, Towards an algebraic theory of context-free languages, Fundam. Inform. 25 (3) (1996) 217–239.
[17] A. Okhotin, Non-erasing variants of the Chomsky—Schützenberger theorem, in: Proc. of the 16th Intern. Conf. on Developments in Language Theory, DLT'12, Springer-Verlag, Berlin, 2012, pp. 121–129.
[18] S. Crespi Reghizzi, P. San Pietro, Non-erasing Chomsky-Schützenberger theorem with grammar-independent alphabet, Inf. Comput. 269 (2019) 1–17.
[19] S. Crespi Reghizzi, A. Restivo, P. San Pietro, Row-column combination of Dyck words, in: H. Fernau, S. Gaspers, R. Klasing (Eds.), SOFSEM 2024: Theory and Practice of Computer Science, Springer, Berlin, 2024, pp. 139–153.
[20] K. Mahalingam, P. Pandoh, HV-palindromes in two-dimensional words, Int. J. Found. Comput. Sci. 33 (5) (2022) 389–409.
[21] J. Hopcroft, J. Ullman, Formal Languages and Their Relation to Automata, Addison-Wesley, Wokingham, 1969.
[22] A. Salomaa, Formal Languages, Academic Press, New York, 1973.
[23] S. Crespi Reghizzi, A. Restivo, P. San Pietro, Reducing the local alphabet size in tiling systems by means of 2D comma-free codes, Theor. Comput. Sci. 933 (2022) 1–20.
[24] E. Barcucci, A. Del Lungo, M. Nivat, R. Pinzani, Reconstructing convex polyominoes from horizontal and vertical projections, Theor. Comput. Sci. 155 (2) (1996) 321–347.
[25] O. Matz, Regular expressions and context-free grammars for picture languages, in: 14th Annual Symposium on Theoretical Aspects of Computer Science, in: LNCS, vol. 1200, 1997, pp. 283–294.
[26] D. Průša, Non-recursive trade-offs between two-dimensional automata and grammars, Theor. Comput. Sci. 610 (2016) 121–132.
[27] S. Crespi Reghizzi, M. Pradella, Tile rewriting grammars and picture languages, Theor. Comput. Sci. 340 (1) (2005) 257–272.
[28] K. Mehlhorn, Pebbling mountain ranges and its application to dcfl-recognition, in: J. de Bakker, J. van Leeuwen (Eds.), Automata, Languages and Programming, Springer, Berlin, 1980, pp. 422–435.
[29] R. Alur, P. Madhusudan, Visibly pushdown languages, in: L. Babai (Ed.), Proceedings of the 36th Annual ACM Symposium on Theory of Computing, 2004, ACM, 2004, pp. 202–211.