

Decidability of multiset, set and numerically decipherable directed figure codes

Włodzimierz Moczurad*

Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

received 11th Apr. 2016, revised 11th Jan. 2017, accepted 12th Apr. 2017.

Codes with various kinds of decipherability, weaker than the usual unique decipherability, have been studied since multiset decipherability was introduced in mid-1980s. We consider decipherability of directed figure codes, where directed figures are defined as labelled polyominoes with designated start and end points, equipped with catenation operation that may use a merging function to resolve possible conflicts. This is one of possible extensions generalizing words and variable-length codes to planar structures. Here, verification whether a given set is a code is no longer decidable in general. We study the decidability status of figure codes depending on catenation type (with or without a merging function), decipherability kind (unique, multiset, set or numeric) and code geometry (several classes determined by relative positions of start and end points of figures). We give decidability or undecidability proofs in all but two cases that remain open.

Keywords: directed figures, uniquely decipherable codes, multiset decipherable codes, decipherability verification

1 Introduction

The classical notion of a code requires that an encoded message should be decoded uniquely, *i.e.* the exact sequence of codewords must be recovered. In some situations, however, it might be sufficient to recover only the multiset, the set or just the number of codewords. This leads to three kinds of decipherability, known as *multiset* (MSD), *set* (SD) and *numeric decipherability* (ND), respectively. The original exact decipherability is called *unique decipherability* (UD).

Multiset decipherability was introduced by Lempel (1986), whilst numeric decipherability originates in Head and Weber (1994). The same authors in Head and Weber (1995) develop what they call “domino graphs” providing a useful technique for decipherability verification. Guzmán (1999) defined set decipherability and presented a unifying approach to different decipherability notions using varieties of monoids. Contributions by Restivo (1989) and Blanchet-Sadri and Morgan (2001) settle Lempel’s conjectures for some MSD and SD codes. Blanchet-Sadri (2001) characterizes decipherability of three-word codes, whilst Burderi and Restivo (2007a,b) relate decipherability to the Kraft inequality and to coding partitions. A paper by Salomaa et al. (2009), although not directly concerned with decipherability, uses ND codes (dubbed *length codes*) to study prime decompositions of languages.

*Supported by National Science Centre (NCN) grant no. 2011/03/B/ST6/00418.

Extensions of classical words and variable-length word codes have also been widely studied. For instance, Aigrain and Beauquier (1995) introduced polyomino codes; two-dimensional rectangular pictures were studied by Giammarresi and Restivo (1996), whilst Mantaci and Restivo (2001) described an algorithm to verify tree codes. Recent results on picture codes include *e.g.* Anselmo et al. (2013a,b). The interest in picture-like structures is not surprising, given the huge amounts of pictorial data in use. Unfortunately, properties related to decipherability are often lost when moving to a two-dimensional plane. In particular, decipherability testing (*i.e.* testing whether a given set is a code) is undecidable for polyominoes and similar structures, *cf.* Beauquier and Nivat (2003); Moczurad (2000).

In Kolarz and Moczurad (2009) we introduced directed figures defined as labelled polyominoes with designated start and end points, equipped with catenation operation that uses a merging function to resolve possible conflicts. This setting is similar to symbolic pixel pictures, described by Costagliola et al. (2005), and admits a natural definition of catenation. The attribute “directed” is used to emphasize the way figures are catenated; this should not be confused with the meaning of “directed” in *e.g.* directed polyominoes. We proved that verification whether a given finite set of directed figures is a UD code is decidable. This still holds true in a slightly more general setting of codes with weak equality (see Moczurad (2010)) and is a significant change in comparison to previously mentioned picture models, facilitating the use of directed figures in, for instance, encoding and indexing of pictures in databases. On the other hand, a directed figure model with no merging function, where catenation of figures is only possible when they do not overlap, has again undecidable UD testing; *cf.* Kolarz (2010a,b). See also Moczurad (2013) for a short description of decipherability characterization with domino graphs.

In the present paper we extend the previous results by considering not just UD codes, but also MSD, SD and ND codes over directed figures. We prove decidability or undecidability for each combination of the following orthogonal criteria: catenation type (with or without a merging function), decipherability kind (UD, MSD, SD, ND) and code geometry (several classes determined by relative positions of start and end points of figures). Two combinations remain open, however.

We begin, in Section 2, with definitions of directed figures and their catenations. Section 3 defines decipherability kinds and shows the relationship between codes of those kinds. In Section 4 main decidability results for decipherability verification are given. Preliminary, short version of this paper appeared as Kolarz and Moczurad (2012).

2 Preliminaries

Let Σ be a finite, non-empty alphabet. A *translation* by vector $u = (u_x, u_y) \in \mathbb{Z}^2$ is denoted by tr_u , $\text{tr}_u : \mathbb{Z}^2 \ni (x, y) \mapsto (x + u_x, y + u_y) \in \mathbb{Z}^2$. By extension, for a set $V \subseteq \mathbb{Z}^2$ and an arbitrary function $f : V \rightarrow \Sigma$ define $\text{tr}_u : \mathcal{P}(\mathbb{Z}^2) \ni V \mapsto \{\text{tr}_u(v) \mid v \in V\} \in \mathcal{P}(\mathbb{Z}^2)$ and $\text{tr}_u : \Sigma^V \ni f \mapsto f \circ \text{tr}_{-u} \in \Sigma^{\text{tr}_u(V)}$.

Definition 1 (Directed figure, *cf.* Kolarz and Moczurad (2009)). *Let $D \subseteq \mathbb{Z}^2$ be finite and non-empty, $b, e \in \mathbb{Z}^2$ and $\ell : D \rightarrow \Sigma$. A quadruple $f = (D, b, e, \ell)$ is a directed figure (over Σ) with*

$$\begin{array}{lll} \text{domain} & \text{dom}(f) & = D, \\ \text{start point} & \text{begin}(f) & = b, \\ \text{end point} & \text{end}(f) & = e, \\ \text{labelling function} & \text{label}(f) & = \ell. \end{array}$$

Translation vector of f is defined as $\text{tran}(f) = \text{end}(f) - \text{begin}(f)$. Additionally, the empty directed figure ε is defined as $(\emptyset, (0, 0), (0, 0), \{\})$, where $\{\}$ denotes a function with an empty domain. Note that the

start and end points need not be in the domain.

The set of all directed figures over Σ is denoted by Σ° . Two directed figures x, y are *equal* (denoted by $x = y$) if there exists $u \in \mathbb{Z}^2$ such that

$$y = (\text{tr}_u(\text{dom}(x)), \text{tr}_u(\text{begin}(x)), \text{tr}_u(\text{end}(x)), \text{tr}_u(\text{label}(x))).$$

Thus, we actually consider figures up to translation.

Example 1. A directed figure and its graphical representation. Each point of the domain, (x, y) , is represented by a unit square in \mathbb{R}^2 with bottom left corner in (x, y) . A circle marks the start point and a diamond marks the end point of the figure. Figures are considered up to translation, hence we do not mark the coordinates.

$$(\{(0, 0), (1, 0), (2, 0), (1, 1)\}, (0, 1), (2, 1), \{(0, 0) \mapsto a, (1, 0) \mapsto b, (1, 1) \mapsto a, (2, 0) \mapsto a\})$$



Definition 2 (Catenation, cf. Kolarz and Moczurad (2009)). Let $x = (D_x, b_x, e_x, \ell_x)$ and $y = (D_y, b_y, e_y, \ell_y)$ be directed figures. If $D_x \cap \text{tr}_{e_x - b_y}(D_y) = \emptyset$, a catenation of x and y is defined as

$$x \circ y = (D_x \cup \text{tr}_{e_x - b_y}(D_y), b_x, \text{tr}_{e_x - b_y}(e_y), \ell),$$

where

$$\ell(z) = \begin{cases} \ell_x(z) & \text{for } z \in D_x, \\ \text{tr}_{e_x - b_y}(\ell_y)(z) & \text{for } z \in \text{tr}_{e_x - b_y}(D_y). \end{cases}$$

If $D_x \cap \text{tr}_{e_x - b_y}(D_y) \neq \emptyset$, catenation of x and y is not defined.

Definition 3 (m -catenation, cf. Kolarz and Moczurad (2009)). Let $x = (D_x, b_x, e_x, \ell_x)$ and $y = (D_y, b_y, e_y, \ell_y)$ be directed figures. An m -catenation of x and y with respect to a merging function $m : \Sigma \times \Sigma \rightarrow \Sigma$ is defined as

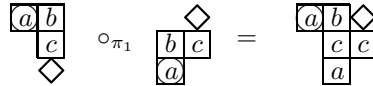
$$x \circ_m y = (D_x \cup \text{tr}_{e_x - b_y}(D_y), b_x, \text{tr}_{e_x - b_y}(e_y), \ell),$$

where

$$\ell(z) = \begin{cases} \ell_x(z) & \text{for } z \in D_x \setminus \text{tr}_{e_x - b_y}(D_y), \\ \text{tr}_{e_x - b_y}(\ell_y)(z) & \text{for } z \in \text{tr}_{e_x - b_y}(D_y) \setminus D_x, \\ m(\ell_x(z), \text{tr}_{e_x - b_y}(\ell_y)(z)) & \text{for } z \in D_x \cap \text{tr}_{e_x - b_y}(D_y). \end{cases}$$

Notice that when $x \circ y$ is defined, it is equal to $x \circ_m y$, regardless of the merging function m .

Example 2. Let π_1 be the projection onto the first argument.



The “non-merging” catenation is not defined for the above figures. Note that the result of (m -)catenation does not depend on the original position of the second argument.

Observe that \circ is associative, whilst \circ_m is associative if and only if m is associative. Thus for associative m , $\Sigma_m^\circ = (\Sigma^\circ, \circ_m)$ is a monoid (which is never free). From now on let m be an arbitrary associative merging function.

Abusing this notation, we also write X° (resp. X_m°) to denote the set of all figures that can be composed by \circ catenation (resp. \circ_m m -catenation) from figures in $X \subseteq \Sigma^\circ$. When some statements are formulated for both \circ and \circ_m , we use the symbol \bullet and “ $x \bullet y$ ” should then be read as “ $x \circ y$ (resp. $x \circ_m y$)”. Similarly, “ $x \in X_\bullet^\circ$ ” should be read as “ $x \in X^\circ$ (resp. $x \in X_m^\circ$)”.

For $u, v \in \mathbb{Z}^2$, $\text{HP}(u, v)$ denotes a half-plane $\{w \in \mathbb{Z}^2 \mid u \cdot (w - (v + u)) \leq 0\}$, where \cdot is the usual scalar product; see Figure 1. An angle between two vectors $u, v \in \mathbb{Z}^2$ is written as $\angle(u, v)$ and $\text{Rot}_\phi(u)$ denotes a rotation of u by an angle ϕ . For $u = (u_x, u_y) \in \mathbb{Z}^2$ and $n \in \mathbb{N}$, $\text{B}(u, n)$ denotes a ball on the integer grid with center u and radius n , i.e., $\text{B}(u, n) = \{(v_x, v_y) \in \mathbb{Z}^2 \mid |u_x - v_x| + |u_y - v_y| \leq n\}$.

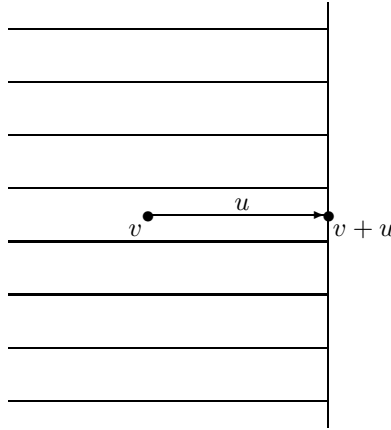


Fig. 1: $\text{HP}(u, v)$. The half-plane contains integer grid points lying on a vertical line and to the left side of that line (the region marked by horizontal lines).

3 Codes

In this section we define a total of eight kinds of directed figure codes, resulting from the use of four different notions of decipherability and two types of catenation. Note that by a *code* (over Σ , with no further attributes) we mean any finite non-empty subset of $\Sigma^\circ \setminus \{\varepsilon\}$.

Definition 4 (UD code). *Let X be a code over Σ . X is a uniquely decipherable code, if for any $x_1, \dots, x_k, y_1, \dots, y_l \in X$ the equality $x_1 \circ \dots \circ x_k = y_1 \circ \dots \circ y_l$ implies that (x_1, \dots, x_k) and (y_1, \dots, y_l) are equal as sequences, i.e. $k = l$ and $x_i = y_i$ for each $i \in \{1, \dots, k\}$.*

Definition 5 (UD m -code). *Let X be a code over Σ . X is a uniquely decipherable m -code, if for any $x_1, \dots, x_k, y_1, \dots, y_l \in X$ the equality $x_1 \circ_m \dots \circ_m x_k = y_1 \circ_m \dots \circ_m y_l$ implies that (x_1, \dots, x_k) and (y_1, \dots, y_l) are equal as sequences.*

In the remaining definitions, we use the obvious abbreviated notation.

Definition 6 (MSD code and m -code). Let X be a code over Σ . X is a multiset decipherable code (resp. m -code), if for any $x_1, \dots, x_k, y_1, \dots, y_l \in X$ the equality $x_1 \bullet \dots \bullet x_k = y_1 \bullet \dots \bullet y_l$ implies that $\{\{x_1, \dots, x_k\}\}$ and $\{\{y_1, \dots, y_l\}\}$ are equal as multisets.

Definition 7 (SD code and m -code). Let X be a code over Σ . X is a set decipherable code (resp. m -code), if for any $x_1, \dots, x_k, y_1, \dots, y_l \in X$ the equality $x_1 \bullet \dots \bullet x_k = y_1 \bullet \dots \bullet y_l$ implies that $\{x_1, \dots, x_k\}$ and $\{y_1, \dots, y_l\}$ are equal as sets.

Definition 8 (ND code and m -code). Let X be a code over Σ . X is a numerically decipherable code (resp. m -code), if for any $x_1, \dots, x_k, y_1, \dots, y_l \in X$ the equality $x_1 \bullet \dots \bullet x_k = y_1 \bullet \dots \bullet y_l$ implies $k = l$.

Proposition 1. If X is a UD (resp. MSD, SD, ND) m -code, then X is a UD (resp. MSD, SD, ND) code.

Proof: Assume X is not a UD (resp. MSD, SD, ND) code. Then for some $x_1, \dots, x_k, y_1, \dots, y_l \in X$ we have $x_1 \circ \dots \circ x_k = y_1 \circ \dots \circ y_l$ with (x_1, \dots, x_k) and (y_1, \dots, y_l) not satisfying the final condition of the respective definition. But then, irrespective of m , $x_1 \circ_m \dots \circ_m x_k = y_1 \circ_m \dots \circ_m y_l$ and X is not a UD (resp. MSD, SD, ND) m -code. \square

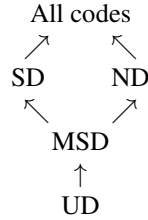
Note that the converse does not hold. A code may, for instance, fail to satisfy the UD m -code definition with $x_1 \circ_m \dots \circ_m x_k = y_1 \circ_m \dots \circ_m y_l$ and still be a UD code simply because some catenations in $x_1 \circ \dots \circ x_k$ and $y_1 \circ \dots \circ y_l$ are not defined.

Example 3. Take $X = \{x = \textcircled{a}\}$. X is not a UD m -code, since $x \circ_m x = x$. It is a trivial UD code, though, because $x \circ x$ is not defined.

Proposition 2. Every UD code is an MSD code; every MSD code is an SD code and an ND code. Every UD m -code is an MSD m -code; every MSD m -code is an SD m -code and an ND m -code.

Proof: Obvious. \square

The diagram illustrates inclusions between different families of codes. A similar diagram can be made for m -codes. Examples given below show that all those inclusions are strict.



Example 4. Four codes depicted below are, respectively, UD, MSD, SD and ND codes and m -codes. They are proper, in the sense that the MSD code is not a UD code (since $x_1 \circ x_2 \circ x_3 \circ x_4 = x_2 \circ x_4 \circ x_1 \circ x_3$), and the SD and ND codes are not MSD codes (since $y_1 \circ y_4 \circ y_4 \circ y_3 \circ y_2 = y_2 \circ y_3 \circ y_1 \circ y_3 \circ y_4 \circ y_1$ and $z_1 \circ z_3 = z_2 \circ z_1$). For the sake of simplicity, we show sets that could also serve as examples for corresponding properties of word codes. In fact, the MSD and SD examples come from Guzmán (1999).

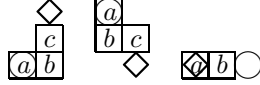
$$\begin{array}{ll}
UD & w_1 = \boxed{a} \diamond \quad w_2 = \boxed{b} \boxed{a} \diamond \quad w_3 = \boxed{b} \boxed{b} \diamond \\
MSD & x_1 = \boxed{a} \boxed{a} \boxed{b} \diamond \quad x_2 = \boxed{a} \boxed{a} \boxed{b} \boxed{a} \boxed{a} \diamond \quad x_3 = \boxed{a} \boxed{b} \boxed{a} \diamond \quad x_4 = \boxed{b} \boxed{a} \boxed{a} \boxed{a} \boxed{b} \boxed{a} \boxed{b} \boxed{a} \diamond \\
SD & y_1 = \boxed{a} \diamond \quad y_2 = \boxed{b} \boxed{a} \boxed{b} \diamond \quad y_3 = \boxed{a} \boxed{a} \boxed{b} \boxed{a} \boxed{a} \diamond \quad y_4 = \boxed{a} \boxed{b} \boxed{a} \boxed{a} \boxed{b} \boxed{a} \diamond \\
ND & z_1 = \boxed{a} \diamond \quad z_2 = \boxed{a} \boxed{b} \diamond \quad z_3 = \boxed{b} \boxed{a} \diamond
\end{array}$$

Before proceeding with the main decidability results, note that for UD, MSD and ND m -codes there is an “easy case” that can be verified quickly just by analyzing the translation vectors of figures. This is reflected in Theorem 1.

Definition 9 (Two-sided and one-sided codes). *Let $X = \{x_1, \dots, x_n\}$ be a code. If there exist non-negative integers $\alpha_1, \dots, \alpha_n$, not all equal to zero, such that $\sum_{i=1}^n \alpha_i \text{tran}(x_i) = (0, 0)$, then X is called two-sided. Otherwise, X is called one-sided.*

This condition can be interpreted geometrically as follows: Translation vectors of a two-sided code do not fit in an open half-plane. For a one-sided code, there exists a line passing through $(0, 0)$ such that all translation vectors are on one side of it. Equivalently, there exists $\tau \in \mathbb{Z}^2$ such that the scalar products $\tau \cdot \text{tran}(x_i)$ are all positive.

Example 5. *The following set of figures is a two-sided code, with translation vectors $(1, 2)$, $(1, -2)$ and $(-2, 0)$:*



It is a one-sided code, if the rightmost figure is removed.

Theorem 1 (Necessary condition). *A two-sided code is not an ND m -code (and consequently neither an MSD nor UD m -code).*

Proof: Assume $X = \{x_1, \dots, x_n\}$ is two-sided, hence there exist non-negative integers $\alpha_1, \dots, \alpha_n$, not all equal to zero, such that $\sum_{i=1}^n \alpha_i \text{tran}(x_i) = (0, 0)$. Let

$$x = \underbrace{x_1 \circ_m \dots \circ_m x_1}_{\alpha_1 \text{ times}} \circ_m \underbrace{x_2 \circ_m \dots \circ_m x_2}_{\alpha_2 \text{ times}} \circ_m \dots \circ_m \underbrace{x_n \circ_m \dots \circ_m x_n}_{\alpha_n \text{ times}}.$$

Now consider the powers of x (with respect to \circ_m), x^i for $i \geq 1$. Since $\text{tran}(x) = (0, 0)$, each of the powers has the same domain. There is only a finite number of possible labellings of this domain, which implies that regardless of the merging function and labelling of x , there exist $p, q \in \mathbb{N}$, $p \neq q$ such that $x^p = x^q$. Hence X is not an ND m -code. \square

Corollary 1. *An ND m -code is one-sided.*

4 Decidability of verification

In this section we summarize all non-trivial decidability results for the decipherability verification. We aim to prove the decidability status for each combination of the following orthogonal criteria: catenation type (with or without a merging function), decipherability kind (UD, MSD, SD, ND) and code geometry (one-sided, two-sided, two-sided with parallel translation vectors). Two combinations remain open, however.

Proofs that have already appeared in our previous work and algorithms are omitted; references to respective papers are given. Note, however, that in all decidable non-trivial cases there exist algorithms to test the decipherability in question; the algorithms effectively find a double factorization of a figure if the answer is negative.

4.1 Positive decidability results

Proposition 3 (see Kolarz and Moczurad (2009), Section 4). *Let X be a one-sided code over Σ . It is decidable whether X is a UD m -code.*

Proposition 4 (see Kolarz (2010b), Section 3). *Let X be a one-sided code over Σ . It is decidable whether X is a UD code.*

Generalizing Propositions 3 and 4, we obtain a similar result for one-sided MSD, SD and ND codes and m -codes.

Theorem 2. *Let X be a one-sided code over Σ . It is decidable whether X is a $\{UD, MSD, SD \text{ or } ND\}$ $\{code \text{ or } m\text{-code}\}$.*

Proof: Starting with observations that allow us to construct a “bounding area” for figures, we proceed with properties that imply finiteness of possible configuration sets and, consequently, decidability of the problem in question.

Let $X = \{x_1, \dots, x_n\} \subseteq \Sigma^\circ$ and let $\text{begin}(x) = (0, 0)$ for each $x \in X$. Since X is one-sided, there exists a vector τ such that for all $x \in X$,

$$\tau \cdot \text{tran}(x) > 0.$$

We can assume that figures are sorted with respect to the angle of their translation vectors in the following way:

$$\angle(\text{Rot}_{-\frac{\pi}{2}}(\tau), \text{tran}(x_1)) \leq \angle(\text{Rot}_{-\frac{\pi}{2}}(\tau), \text{tran}(x_2)) \leq \dots \leq \angle(\text{Rot}_{-\frac{\pi}{2}}(\tau), \text{tran}(x_n)).$$

We choose constants $r_E, r_N, r_W, r_S > 0$ such that the vectors

$$\begin{aligned} \tau_E &= r_E \tau, \\ \tau_N &= r_N \text{Rot}_{\frac{\pi}{2}}(\text{tran}(x_n)), \\ \tau_W &= -r_W \tau, \\ \tau_S &= r_S \text{Rot}_{-\frac{\pi}{2}}(\text{tran}(x_1)) \end{aligned}$$

define a “bounding area” for figures in X , *i.e.*, for all $x \in X$,

$$\text{dom}(x) \cup \{\text{end}(x)\} \subseteq \bigcap_{u \in \{\tau_E, \tau_N, \tau_W, \tau_S\}} \{\text{HP}(u, \text{begin}(x))\}.$$

The choice of τ determines a “central axis” along which figures will be catenated. This is the line that bisects the half-plane containing all translation vectors of figures in X . Note that in all examples, τ and τ_E are drawn as horizontal pointing eastwards, giving the natural meaning to the subscripts of $\tau_E, \tau_N,$

τ_W and τ_S vectors. The ordering of translation vectors of figures in X is thus from the “southernmost” to “northernmost”.

For $x \in X_\bullet^\diamond$ define

$$\begin{aligned} \text{CE}^+(x) &= \text{HP}(\tau_S, \text{end}(x)) \cap \text{HP}(\tau_N, \text{end}(x)) \cap \text{HP}(\tau_W, \text{end}(x)), \\ \text{CE}^-(x) &= \mathbb{Z}^2 \setminus \text{CE}^+(x), \\ \text{CW}^+(x) &= \bigcup_v \{v + (\text{CE}^+(x) \cap \text{HP}(\tau_E, \text{end}(x)))\}, \\ \text{CW}^-(x) &= \mathbb{Z}^2 \setminus \text{CW}^+(x), \end{aligned}$$

where the union in the definition of $\text{CW}^+(x)$ is taken over $v \in \mathbb{Z}^2$ lying within an angle spanned by vectors $-\text{tran}(x_1)$ and $-\text{tran}(x_n)$. Note that each term of the union is a trapezoid, resulting from the intersection of four half-planes; see Figure 2 and Figure 3.

Immediately from the definition we have following properties, for $x, y \in X_\bullet^\diamond$:

$$\begin{aligned} u \in \text{CE}^-(x) \cap \text{dom}(x) &\Rightarrow \text{label}(x)(u) = \text{label}(x \bullet y)(u), \\ u \in \text{CE}^-(x) \setminus \text{dom}(x) &\Rightarrow u \notin \text{dom}(x \bullet y), \\ u \in \text{CW}^-(x) &\Rightarrow u \notin \text{dom}(x), \\ \text{CE}^+(x \bullet y) &\subseteq \text{CE}^+(x), \\ \text{CW}^+(x) &\subseteq \text{CW}^+(x \bullet y). \end{aligned}$$

For $x_1, \dots, x_k, y_1, \dots, y_l \in X_\bullet^\diamond$ we define a *configuration* as a pair of sequences $((x_1, \dots, x_k), (y_1, \dots, y_l))$. A *successor* of such a configuration is either $((x_1, \dots, x_k, z), (y_1, \dots, y_l))$ or $((x_1, \dots, x_k), (y_1, \dots, y_l, z))$ for some $z \in X$. If a configuration C_2 is a successor of C_1 , we write $C_1 \prec C_2$. By \prec^* we denote the transitive closure of \prec .

For a configuration $C = ((x_1, \dots, x_k), (y_1, \dots, y_l))$ let us denote:

$$\begin{aligned} L(C) &= \{x_1, \dots, x_k\}, \\ L_\bullet(C) &= x_1 \bullet \dots \bullet x_k, \\ R(C) &= \{y_1, \dots, y_l\}, \\ R_\bullet(C) &= y_1 \bullet \dots \bullet y_l. \end{aligned}$$

Now consider a starting configuration $((x), (y))$, for $x, y \in X, x \neq y$. Assume that there exists a configuration C such that $L_\bullet(C) = R_\bullet(C)$ and $((x), (y)) \prec^* C$. Now we have:

- X is not a UD code (resp. UD m -code),
- if $L(C) = R(C)$ as multisets then X is not an MSD code (resp. MSD m -code),
- if $L(C) = R(C)$ as sets then X is not an SD code (resp. SD m -code),
- if $|L(C)| = |R(C)|$ then X is not an ND code (resp. ND m -code).

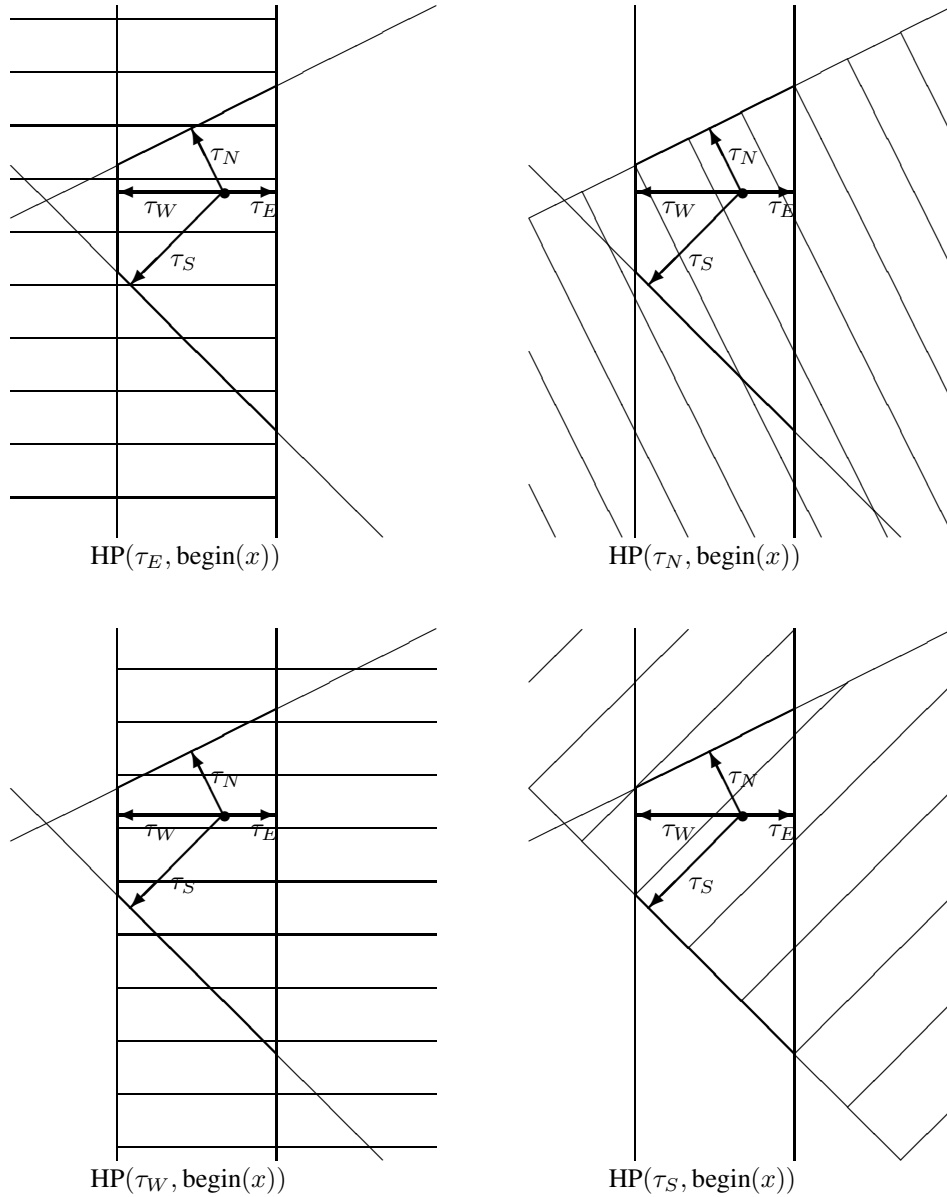


Fig. 2: Half-planes $HP(\tau, \text{begin}(x))$ for $\tau \in \{\tau_E, \tau_N, \tau_W, \tau_S\}$ are marked with parallel lines; the black dot denotes the start point of x .

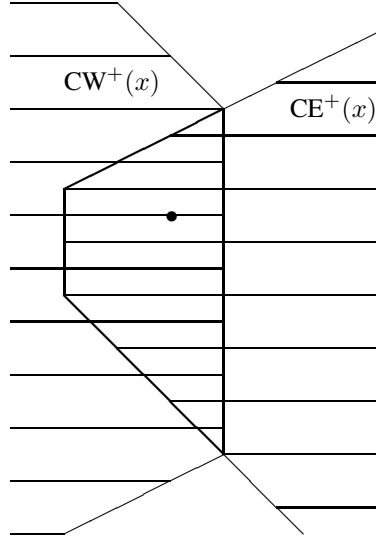


Fig. 3: $CW^+(x)$ and $CE^+(x)$ regions; the black dot denotes the end point of x .

A configuration C' such that $C' \prec^* C$ and $L_\bullet(C) = R_\bullet(C)$ for some C , is called a *proper configuration*.

Our goal is either to show that there exists no proper configuration, or to find such configuration(s). In the former case, X is a code (resp. m -code) of each kind. In the latter case, if we find one of such configurations, X is already not a UD code (resp. UD m -code). To verify whether X is an MSD, SD or ND code (resp. m -code), we have to check the above conditions for *all* possible proper configurations.

Let

$$\rho = \max_{x \in X} \min\{n \in \mathbb{N} \mid B(\text{begin}(x), n) \cap \text{dom}(x) \neq \emptyset\}.$$

This number determines a distance within which both parts of a configuration, L and R , can be found. The following properties of a proper configuration C are now easily verified:

$$B(\text{end}(L_\bullet(C)), \rho) \cap (CW^+(R_\bullet(C)) \cup CE^+(R_\bullet(C))) \neq \emptyset, \quad (1)$$

$$B(\text{end}(R_\bullet(C)), \rho) \cap (CW^+(L_\bullet(C)) \cup CE^+(L_\bullet(C))) \neq \emptyset, \quad (2)$$

and for the common domain $D = CE^-(L_\bullet(C)) \cap CE^-(R_\bullet(C))$:

$$\text{label}(L_\bullet(C)) \upharpoonright_D \equiv \text{label}(R_\bullet(C)) \upharpoonright_D. \quad (3)$$

Notice that we do not need all of the information contained in configurations, just those labellings that can be changed by future catenations. By (3), instead of a configuration C we can consider a *reduced configuration* defined as a pair $(\pi_{RC}(L_\bullet(C), R_\bullet(C)), \pi_{RC}(R_\bullet(C), L_\bullet(C)))$ where

$$\pi_{RC}(z, z') = (\text{end}(z), \text{label}(z) \upharpoonright_{\text{dom}(z) \setminus (CE^-(z) \cap CE^-(z'))}).$$

Obviously we need only consider configurations where the span along τ_E is bounded by $|\tau_E|$, *i.e.*,

$$|\tau_E \cdot (\text{end}(L_\bullet(C)) - \text{end}(R_\bullet(C)))| \leq |\tau_E|^2,$$

since no single figure advances $\text{end}(L_\bullet(C))$ or $\text{end}(R_\bullet(C))$ by more than $|\tau_E|$. Moreover, (1) and (2) restrict the perpendicular span (in the direction of $\text{Rot}_{-\frac{\pi}{2}}(\tau_E)$). Hence the number of reduced configurations, up to translation, is finite and there is a finite number of proper configurations to check. Consequently, we can verify whether X is a UD, MSD, SD or ND code (resp. m -code). \square

Combined with Theorem 1, this proves the decidability for all UD, MSD and ND m -codes. The case of two-sided SD m -codes remains unsolved, however.

Two-sided codes with parallel translation vectors constitute an interesting special case.

Definition 10 (Two-sided codes with parallel translation vectors). *Let $X = \{x_1, \dots, x_n\}$ be a two-sided code. If there exists a vector $\tau \in \mathbb{Z}^2$ and numbers $\alpha_1, \dots, \alpha_n \in \mathbb{Z}$, not all positive and not all negative, such that $\text{tran}(x_i) = \alpha_i \tau$ for $i = 1, \dots, n$, then X is called two-sided with parallel translation vectors.*

Proposition 5 (see Kolarz (2010a), Section 4). *Let X be a two-sided code with parallel translation vectors. It is decidable whether X is a UD code.*

This can again be generalized to two-sided MSD, SD and ND codes with parallel translation vectors:

Theorem 3. *Let X be a two-sided code with parallel translation vectors. It is decidable whether X is a UD, MSD, SD or ND code.*

Proof: Even though the problem is one-dimensional, it cannot be easily transformed to any known word problem. Hence, a setting similar to that of Theorem 2 is used: we define bounding areas and use them to show that the number of possible configurations is finite. This is accomplished by trying to find a figure that has two different factorizations and observing that the configurations are indeed bounded.

Let $X \subseteq \Sigma^\circ$ be finite and non-empty and let $\text{begin}(x) = (0, 0)$ for each $x \in X$. Since translation vectors of elements of X are parallel, there exists a shortest vector $\tau \in \mathbb{Z}^2$ such that for all $x \in X$,

$$\text{tran}(x) \in \mathbb{Z}\tau = \{j\tau \mid j \in \mathbb{Z}\}.$$

In particular, if $(t_1, t_2) = \text{tran}(x)$ for some $x \in X$ with $\text{tran}(x) \neq (0, 0)$, then τ is one of the following vectors:

$$(t_1 / \gcd(|t_1|, |t_2|) \quad , \quad t_2 / \gcd(|t_1|, |t_2|)), \quad (4)$$

$$(-t_1 / \gcd(|t_1|, |t_2|) \quad , \quad -t_2 / \gcd(|t_1|, |t_2|)), \quad (5)$$

where gcd denotes greatest common divisor. If all translation vectors of elements of X are $(0, 0)$, then the decidability problem is trivial: X is an MSD, SD and ND code (since each element can be used at most once) and X is a UD code if and only if no two elements can be concatenated, *i.e.* no two elements $x, y \in X$ have $\text{dom}(x) \cap \text{dom}(y) \neq \emptyset$ (otherwise $xy = yx$); this case is obviously decidable.

We define the following *bounding areas*:

$$B_L = \{u \in \mathbb{Z}^2 \mid 0 > u \cdot \tau\},$$

$$B_0 = \{u \in \mathbb{Z}^2 \mid 0 \leq u \cdot \tau < \tau \cdot \tau\},$$

$$B_R = \{u \in \mathbb{Z}^2 \mid \tau \cdot \tau \leq u \cdot \tau\}.$$

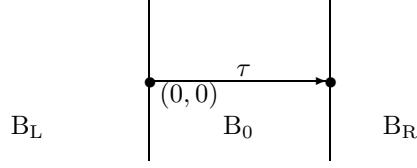


Fig. 4: Bounding areas B_L , B_0 and B_R .

For a non-empty figure $x \in \Sigma^\diamond$, *bounding hulls* of x are sets:

$$\begin{aligned} \text{hull}(x) &= \bigcup_{n=m \dots M} \text{tr}_{n\tau}(\mathbf{B}_0), \\ \text{hull}^*(x) &= \bigcup_{n=-M \dots -m} \text{tr}_{n\tau}(\mathbf{B}_0), \end{aligned}$$

where

$$\begin{aligned} m &= \min\{n \in \mathbb{Z} \mid \text{tr}_{n\tau}(\mathbf{B}_0) \cap (\text{dom}(x) \cup \{\text{begin}(x), \text{end}(x)\}) \neq \emptyset\}, \\ M &= \max\{n \in \mathbb{Z} \mid \text{tr}_{n\tau}(\mathbf{B}_0) \cap (\text{dom}(x) \cup \{\text{begin}(x), \text{end}(x)\}) \neq \emptyset\}. \end{aligned}$$

In addition, for the empty figure, $\text{hull}(\varepsilon) = \emptyset$ and $\text{hull}^*(\varepsilon) = \emptyset$.

The area B_0 is a vertical stripe of width equal to the length of τ . For a figure $x \in \Sigma^\diamond$, $\text{hull}(x)$ is a union of translated stripes such that the whole figure, including its start and end points, lies inside it. The $\text{hull}^*(x)$ variant is a mirror image of $\text{hull}(x)$.

Starting Configurations: Our goal is either to find a figure $x \in X^\diamond$ that has two different factorizations over elements of X , or to show that such a figure does not exist. If it exists, without loss of generality we can assume it has the following two different x - and y -factorizations:

$$x = \dot{x}_1 \ddot{x}_1 \cdots \ddot{x}_{k-1} \dot{x}_k \ddot{x}_k = \dot{y}_1 \ddot{y}_1 \cdots \ddot{y}_{l-1} \dot{y}_l \ddot{y}_l$$

where $\dot{x}_1 \neq \dot{y}_1$, $\text{begin}(\dot{x}_1) = \text{begin}(\dot{y}_1) = (0, 0)$ and for $i \in \{1, \dots, k\}$ and $j \in \{1, \dots, l\}$ we have:

$$\begin{array}{ll} \dot{x}_i \in X & \text{and } \text{hull}(\dot{x}_i) \cap B_0 \neq \emptyset, \\ \ddot{x}_i \in X^\diamond \cup \{\varepsilon\} & \text{and } \text{hull}(\ddot{x}_i) \cap B_0 = \emptyset, \\ \dot{y}_j \in X & \text{and } \text{hull}(\dot{y}_j) \cap B_0 \neq \emptyset, \\ \ddot{y}_j \in X^\diamond \cup \{\varepsilon\} & \text{and } \text{hull}(\ddot{y}_j) \cap B_0 = \emptyset. \end{array}$$

Observe that the following conditions for the x -factorization are satisfied for $i \in \{1, \dots, k-1\}$:

- if $\text{end}(\dot{x}_i) \in B_L$, then $\text{begin}(\dot{x}_{i+1}) \in B_L$,
- if $\text{end}(\dot{x}_i) = (0, 0)$, then $\ddot{x}_i = \varepsilon$ and $\text{begin}(\dot{x}_{i+1}) = (0, 0)$,
- if $\text{end}(\dot{x}_i) \in B_R$, then $\text{begin}(\dot{x}_{i+1}) \in B_R$.

These are trivial implications of the assumption that $\text{hull}(\ddot{x}_i) \cap B_0 = \emptyset$ and the fact that \dot{x}_i must be somehow linked with \dot{x}_{i+1} . Similar conditions are satisfied for the y -factorization. In addition, the x -factorization must match the y -factorization, *i.e.*:

- if $\text{end}(\dot{x}_k) \in B_L$, then $\text{end}(\dot{y}_l) \in B_L$,
- if $\text{end}(\dot{x}_k) = (0, 0)$, then $\text{end}(\dot{y}_l) = (0, 0)$,
- if $\text{end}(\dot{x}_k) \in B_R$, then $\text{end}(\dot{y}_l) \in B_R$.

Also, it is clear that

$$\bigcup_{i=1\dots k} \text{dom}(\dot{x}_i) \cap B_0 = \bigcup_{i=1\dots l} \text{dom}(\dot{y}_i) \cap B_0, \quad (6)$$

$$\bigcup_{i=1\dots k} \text{label}(\dot{x}_i) |_{B_0} = \bigcup_{i=1\dots l} \text{label}(\dot{y}_i) |_{B_0}. \quad (7)$$

Now we consider all possible pairs of sequences $((\dot{x}_i)_i, (\dot{y}_j)_j)$ satisfying the above conditions. Note that equality of such sequences is considered not up to translation: relative position of sequence elements is important. Such a pair will be called a *starting configuration*. Observe that there can be only a finite number of such configurations, since

$$\begin{aligned} \bigcup_{i=1\dots k} \text{dom}(\dot{x}_i) &\subseteq \bigcup_{x \in X} (\text{hull}(x) \cup \text{hull}^*(x)), \\ \bigcup_{i=1\dots l} \text{dom}(\dot{y}_i) &\subseteq \bigcup_{x \in X} (\text{hull}(x) \cup \text{hull}^*(x)) \end{aligned}$$

and the set on the right hand side is bounded in the direction of τ . Also note that if there is no starting configuration for X , then obviously X is a UD code and consequently an MSD, SD and ND code.

Left and Right Configurations: We consider independently all starting configurations constructed for X . By (6) and (7), we can now forget the labelling of B_0 . From a starting configuration $((\dot{x}_i)_{i=1}^k, (\dot{y}_j)_{j=1}^l)$ we construct *L-* and *R-configurations* (left and right configurations)

$$\begin{aligned} C_R &= ((D_R^x, l_R^x, EB_R^x), (D_R^y, l_R^y, EB_R^y)), \\ C_L &= ((D_L^x, l_L^x, EB_L^x), (D_L^y, l_L^y, EB_L^y)). \end{aligned}$$

First we show a construction for the x -part of a configuration:

$$\begin{aligned} D_R^x &= \bigcup_{i=1\dots k} \text{dom}(\dot{x}_i) \cap B_R & \text{and} & & l_R^x &= \bigcup_{i=1\dots k} \text{label}(\dot{x}_i) |_{B_R}, \\ D_L^x &= \bigcup_{i=1\dots k} \text{dom}(\dot{x}_i) \cap B_L & \text{and} & & l_L^x &= \bigcup_{i=1\dots k} \text{label}(\dot{x}_i) |_{B_L} \end{aligned}$$

and multisets EB_L^x, EB_R^x are obtained in the following way: for each $i \in \{1, \dots, k-1\}$:

- if $\text{end}(\dot{x}_i) \in B_L$, then $(\text{end}(\dot{x}_i), \text{begin}(\dot{x}_{i+1}))$ is added to EB_L^x ,
- if $\text{end}(\dot{x}_i) = (0, 0)$, then no pair is added to EB_L^x or EB_R^x ,
- if $\text{end}(\dot{x}_i) \in B_R$, then $(\text{end}(\dot{x}_i), \text{begin}(\dot{x}_{i+1}))$ is added to EB_R^x

and

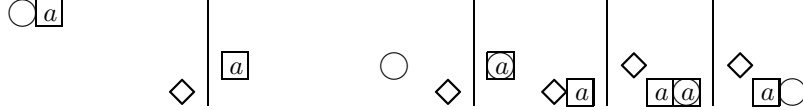
- if $\text{end}(\dot{x}_k) \in B_L$, then $(\text{end}(\dot{x}_k), \odot)$ is added to EB_L^x ,

- if $\text{end}(\dot{x}_k) = (0, 0)$, then no pair is added to EB_L^x or EB_R^x ,
- if $\text{end}(\dot{x}_k) \in \text{B}_R$, then $(\text{end}(\dot{x}_1), \odot)$ is added to EB_R^x .

These multisets keep information on how figures \dot{x}_i and \dot{x}_{i+1} should be linked by \ddot{x}_i factors. The \odot symbol denotes the end of the whole figure.

The y -part is created in a similar way.

Example 6. Consider a set containing the following figures (vertical lines separate the figures):



Taking $\tau = (2, -1)$, we construct one of possible starting configurations (x -part only). We also show the construction of the x -part of L- and R-configurations.

Figure 5 shows the construction. Each image presents a current figure (with bold lines) and its translation vector. Domain and labelling of all of the previous figures are also presented, together with the end point of the previous figure (which is important for the construction). B_0 lies between the slanted lines. Domains and labellings of L- and R-configurations are presented in Fig. 6.

Now let us consider the R-configuration only (the L-configuration is handled in a similar way). We say that an R-configuration $((D_R^x, l_R^x, \text{EB}_R^x), (D_R^y, l_R^y, \text{EB}_R^y))$ is *terminating* if it satisfies the following conditions:

- the domain and labelling of the x -part of the R-configuration match the domain and labelling of its y -part, *i.e.*,

$$D_R^x = D_R^y \text{ and } l_R^x = l_R^y,$$

- if a location of the end point of the whole figure is encoded in the R-configuration, then its location is the same in both x - and y -parts, *i.e.*, for all $e \in \mathbb{Z}$,

$$(e, \odot) \in \text{EB}_R^x \Leftrightarrow (e, \odot) \in \text{EB}_R^y,$$

- all points that should be linked together are trivially linked, since they are the same points, *i.e.*, for all $(e, b) \in \text{EB}_R^x \cup \text{EB}_R^y$,

$$e = b \text{ or } b = \odot.$$

Note that if for some starting configuration we obtain a pair of terminating L- and R- configurations, then X is not a UD code (it can still be an MSD, SD or ND code, though). On the other hand, if we show that for all starting configurations such pair of terminating L- and R-configurations cannot be reached, then X is a UD code (and hence an MSD, SD and ND code).

Similarly as in Theorem 2, to verify whether X is an MSD, SD or ND code, we have to check the following conditions for all possible pairs C of terminating L- and R- configurations:

- if $\pi_x(C) = \pi_y(C)$ as multisets then X is not an MSD code,
- if $\pi_x(C) = \pi_y(C)$ as sets then X is not an SD code,

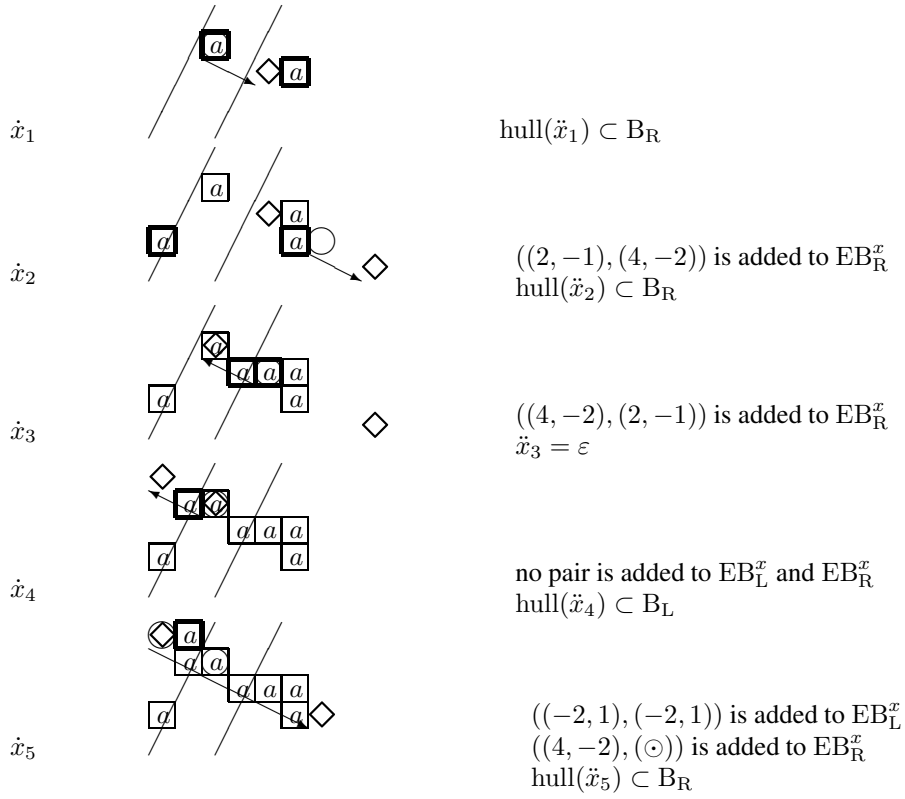


Fig. 5: Construction of a sample starting configuration and its L- and R-configurations (figures added at each step are marked with thick lines).

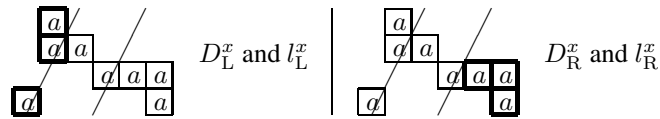


Fig. 6: Domains and labellings of sample L- and R-configurations (respective objects are marked with thick lines).

- if $|\pi_x(C)| = |\pi_y(C)|$ then X is not an ND code,

where $\pi_x(C)$ and $\pi_y(C)$ denote respective multisets of elements used in the construction of C . Note that computation of $\pi_x(C)$ and $\pi_y(C)$ requires the history of C to be kept; this does not spoil the finiteness of the part of C that has to be kept.

Obtaining New R-Configurations: When an R-configuration derived from a starting configuration is terminating, we can proceed to the analysis of the L-configuration. If the R-configuration is not terminating, we must check whether adding new figures may create a terminating configuration.

Initially such a derived configuration lies in B_R . For simplicity of notation, we can translate such a configuration by a vector $-\tau$ (translating all its elements).

Now from the given R-configuration we want to obtain a new R-configuration by adding new figures from X . In order to obtain a new R-configuration from a given R-configuration, we create the new R-configuration as a copy of the old one. Then zero or more of the following operations must be performed (note that they need not be admissible for an arbitrary R-configuration or we may not need such operations to be performed):

- an x -part operation: add any $x \in X$ for which

$$\text{hull}(x) \cap B_L = \emptyset, \quad (8)$$

$$\text{hull}(x) \cap B_0 \neq \emptyset, \quad (9)$$

$$\text{dom}(x) \cap D_R^x = \emptyset \quad (10)$$

to the new configuration, adding its domain and labelling function to the domain and labelling function of the R -configuration, and replacing any pair (e, b) from EB_R^x in the old configuration with two pairs $(e, \text{begin}(x))$ and $(\text{end}(x), b)$ in the new one,

- an y -part operation: similarly.

In each step of creating the new generation of an R-configuration, we add only figures that change the given R-configuration within B_0 ; hence (9). We add such figures to an R-configuration only at that step. In consecutive steps adding such figures is forbidden; hence (8). At the first step this is a consequence of restrictions for \ddot{x}_i and \ddot{y}_i . Condition (10) is obvious. Of course it is possible that a given R-configuration is not extendable at all.

After these operations we want the x -part of the R-configuration obtained to match its y -part on B_0 , *i.e.*,

$$D_R^x \cap B_0 = D_R^y \cap B_0 \text{ and } l_R^x|_{B_0} = l_R^y|_{B_0}.$$

In addition, for the x -part (and similarly for the y -part):

- if $((0, 0), b) \in EB_R^x$, then $b = (0, 0)$ or $b = \odot$,
- if $(e, (0, 0)) \in EB_R^x$, then $e = (0, 0)$,

and for both parts

- $((0, 0), \odot) \in EB_R^x$ if and only if $((0, 0), \odot) \in EB_R^y$.

These conditions are trivial consequences of (8), (9) and (10) on new figures added to R-configuration. Of course it is possible that one cannot obtain any R-configuration from the old one.

Here, since the x -part and y -part of each newly created R-configuration are the same, we now do not have to remember the labelling of B_0 . When we forget this information, configurations created lie in B_R , so we can translate them by $-\tau$ as previously.

Now observe that all parts of an R-configuration are bounded: domains are contained in the area restricted by the widest hull of elements of X ; multisets EB_R^x and EB_R^y cannot be infinite, since eventually all points must be linked. There are only finitely many such configurations. Either we find a terminating R-configuration, or we consider all configurations that can be obtained from a given starting configuration performing one or more steps described. \square

Note that codes with parallel translation vectors are similar to classical word codes and two-sidedness does not make a significant difference in terms of decidability. This can be contrasted with the Post Correspondence Problem (PCP), which is also “linear” yet undecidable. The essential difference is that PCP configurations are extended with pre-defined pairs of words and there is no *a priori* bound on how much two parts of a configuration can differ. Code configurations are extended with individual words or figures and the respective bound can be determined by inspecting the size of words/figures.

4.2 Negative decidability results

Proposition 6 (see Kolarz (2010b), Section 2). *Let X be a two-sided code over Σ . It is undecidable whether X is a UD code.*

This result can again be extended to other decipherability kinds:

Theorem 4. *Let X be a two-sided code over Σ . It is undecidable whether X is a UD, MSD, SD or ND code.*

Proof: We prove Theorem 4 for UD codes first. The same reasoning is applied to MSD and SD codes, whilst for ND codes we use an additional technique, described at the end of this proof. The proof is a reduction from PCP to the decipherability problem. Given a PCP instance, we construct a two-sided code such that the PCP instance has a solution if and only if the code is not decipherable. Detailed explanation why this is indeed the case is given in the form of separate Lemmas 1 and 2, for the “only if” and “if” part, respectively. They are, however, part of the proof since they rely heavily on notations introduced here and would be impossible to formulate clearly outside this context.

First we define figures that will be used throughout the reduction. Let $\Sigma = \{a\}$. For positive integers h, h_N, h_E, h_S, h_W such that $h_N, h_E, h_S, h_W \leq h$ and $b, e \in \{N, E, S, W\}$ (with the usual geographical meaning) we define a *directed hooked square* $DHS_h(h_N, h_E, h_S, h_W)_e^b$ to be a directed figure $f \in \Sigma^\circ$ with:

$$\begin{aligned} \text{dom}(f) &= (B \setminus (H_N^- \cup H_E^- \cup H_S^- \cup H_W^-)) \cup (H_N^+ \cup H_E^+ \cup H_S^+ \cup H_W^+), \\ \text{begin}(f) &= \begin{cases} (0, h+2) & \text{if } b = N, \\ (h+2, 0) & \text{if } b = E, \\ (0, -h-2) & \text{if } b = S, \\ (-h-2, 0) & \text{if } b = W, \end{cases} \end{aligned}$$

$$\text{end}(f) = \begin{cases} (0, h+3) & \text{if } e = N, \\ (h+3, 0) & \text{if } e = E, \\ (0, -h-3) & \text{if } e = S, \\ (-h-3, 0) & \text{if } e = W, \end{cases}$$

where

$$\begin{aligned} B &= \{(x, y) \mid x, y \in \{-h-2, \dots, h+2\}\}, \\ H_N^- &= \{(-1, y) \mid y \in \{h+2-h_N, \dots, h+2\}\} \cup \{(0, h+2-h_N)\}, \\ H_E^- &= \{(x, 1) \mid x \in \{h+2-h_E, \dots, h+2\}\} \cup \{(h+2-h_E, 0)\}, \\ H_S^- &= \{(1, y) \mid y \in \{-h-2, \dots, -h-2+h_S\}\} \cup \{(0, -h-2+h_S)\}, \\ H_W^- &= \{(x, -1) \mid x \in \{-h-2, \dots, -h-2+h_W\}\} \cup \{(-h-2+h_W, 0)\}, \\ H_N^+ &= \{(1, y) \mid y \in \{h+3, \dots, h+3+h_N\}\} \cup \{(0, h+3+h_N)\}, \\ H_E^+ &= \{(x, -1) \mid x \in \{h+3, \dots, h+3+h_E\}\} \cup \{(h+3+h_E, 0)\}, \\ H_S^+ &= \{(-1, y) \mid y \in \{-h-3-h_S, \dots, -h-3\}\} \cup \{(0, -h-3-h_S)\}, \\ H_W^+ &= \{(x, 1) \mid x \in \{-h-3-h_W, \dots, -h-3\}\} \cup \{(-h-3-h_W, 0)\}, \end{aligned}$$

i.e. f is a square with hooks on each side (see e.g. Figure 7).

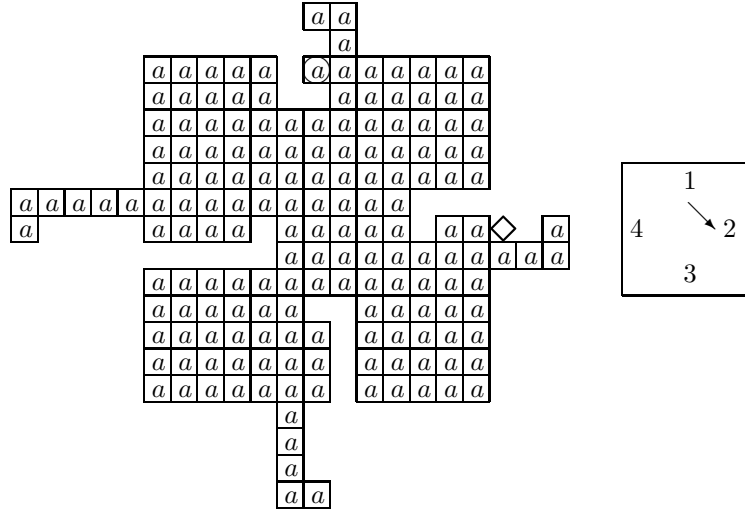


Fig. 7: $\text{DHS}_4(1, 2, 3, 4)_E^N$; full and reduced graphical representation.

Observe that for

$$x = \text{DHS}_h(h_N, h_E, h_S, h_W)_e^b \text{ and } x' = \text{DHS}_h(h'_N, h'_E, h'_S, h'_W)_{e'}^{b'}$$

catenation $x \circ x'$ is defined if and only if e matches b' , i.e.,

$$\begin{array}{lcl}
e = N & \text{and} & b' = S \quad \text{or} \\
e = E & \text{and} & b' = W \quad \text{or} \\
e = S & \text{and} & b' = N \quad \text{or} \\
e = W & \text{and} & b' = E
\end{array}$$

and $h_e = h'_b$.

Now we encode a PCP instance in a set of directed figures over $\Sigma = \{a\}$. The PCP can be stated as follows: Let $A = \{a_1, \dots, a_p\}$ be a finite alphabet, $x_1, \dots, x_k, y_1, \dots, y_k \in A^+$ such that $x_i \neq y_i$ for $i \in \{1, \dots, k\}$. Find a sequence $i_1, \dots, i_n \in \{1, \dots, k\}$, $n \geq 2$, such that $x_{i_1} \cdots x_{i_n} = y_{i_1} \cdots y_{i_n}$.

We describe a set of directed figures \mathbf{X} such that a given PCP instance has a solution if and only if \mathbf{X} is not a UD code. Consider the following set:

$$H = \bigcup_{i \in \{1, \dots, k\}} \{x_i, y_i, e_{x_i}, e_{y_i}, I_i\} \cup \{a_i \mid i \in \{1, \dots, p\}\} \cup \{x, y, x', y', b_x, b_y, e\},$$

where I_i are additional elements related to each pair (x_i, y_i) of the PCP instance. Set $h = |H| = 5k + p + 7$. We can define a bijection between H and $\{1, \dots, h\}$, so from now on, each element of H is identified with its image by this bijection. Since h is now fixed, we write $\text{DHS}(h_N, h_E, h_S, h_W)_e^b$ instead of $\text{DHS}_h(h_N, h_E, h_S, h_W)_e^b$.

For each x_i , $i \in \{1, \dots, k\}$, we define *basic-figures* $[x_i]_E^W$, $]x_i[_E^W$ and $]x_i]_S^W$ (Figure 8); these figures will be used to encode the word x_i standing at the beginning (we call it *begin solution figure*), in the middle (*middle solution figure*) and at the end (*end solution figure*) of the PCP instance solution, respectively.

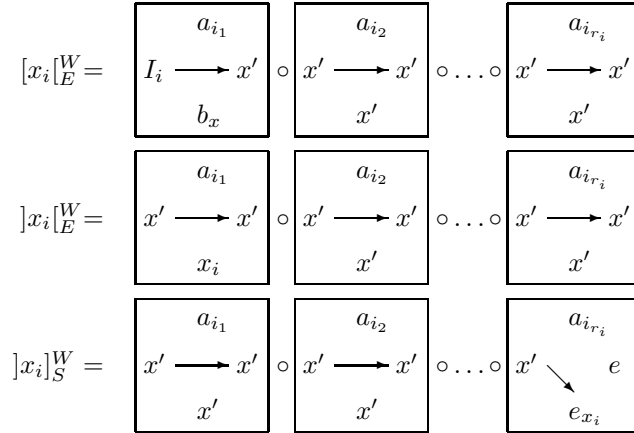


Fig. 8: Basic-figures for $x_i = a_{i_1} \cdots a_{i_{r_i}}$.

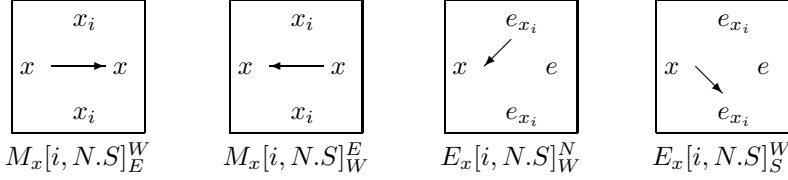
In addition we define *annex-figures* (Figure 9).

In the same way we define figures for the “ y -part” of the PCP instance, replacing the letter x with y .

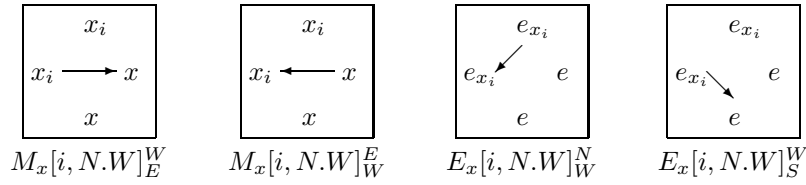
Let \mathbf{X} be the set of all figures defined ($6k$ basic-figures and $32k + 2$ annex-figures, $16k$ for each part: “ x -part” and “ y -part”). Observe that there exists no half-plane of integer values anchored in $(0, 0)$ (i.e. $\{v \in \mathbb{Z}^2 \mid u \cdot v > 0\}$ for some $u \in \mathbb{Z}^2$) containing all translation vectors of the figures we have defined.

The following two lemmas now complete the proof of Theorem 4 for UD, MSD and SD cases.

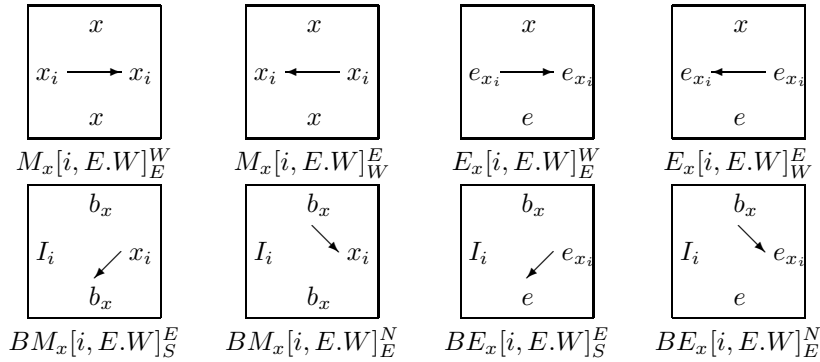
Annex-figures for passing information from north to south:



Annex-figures for passing information from north to west:



Annex-figures for passing information from east to west:



Annex-figures which pass no information:

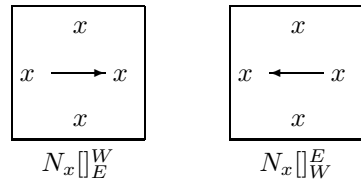


Fig. 9: Annex-figures.

Lemma 1. *If the PCP instance has a solution then \mathbf{X} is not a UD (MSD, SD) code.*

Proof of Lemma 1: Let $i_1, \dots, i_n \in \{1, \dots, k\}$ be a solution of the PCP instance, i.e. $x_{i_1} \cdots x_{i_n} = y_{i_1} \cdots y_{i_n}$. Consider the following directed figures:

$$wx_1 = [x_{i_1} \begin{smallmatrix} W \\ E \end{smallmatrix} \circ] x_{i_2} \begin{smallmatrix} W \\ E \end{smallmatrix} \circ \cdots \circ x_{i_{n-1}} \begin{smallmatrix} W \\ E \end{smallmatrix} \circ x_{i_n} \begin{smallmatrix} W \\ S \end{smallmatrix} \begin{smallmatrix} W \\ S \end{smallmatrix},$$

$$\begin{aligned} wx_j = & E_x[i_n, N.S] \begin{smallmatrix} N \\ W \end{smallmatrix} \circ \underbrace{N_x \begin{smallmatrix} E \\ W \end{smallmatrix} \circ \cdots \circ N_x \begin{smallmatrix} E \\ W \end{smallmatrix} \circ}_{|x_n|-2 \text{ times}} \\ & \underbrace{N_x \begin{smallmatrix} E \\ W \end{smallmatrix} \circ \cdots \circ N_x \begin{smallmatrix} E \\ W \end{smallmatrix} \circ}_{|x_{n-1}|-1 \text{ times}} \circ M_x[i_{n-1}, N.E] \begin{smallmatrix} E \\ W \end{smallmatrix} \circ \\ & \dots \\ & \underbrace{N_x \begin{smallmatrix} E \\ W \end{smallmatrix} \circ \cdots \circ N_x \begin{smallmatrix} E \\ W \end{smallmatrix} \circ}_{|x_{j+1}|-1 \text{ times}} \circ M_x[i_{j+1}, N.E] \begin{smallmatrix} E \\ W \end{smallmatrix} \circ \\ & \underbrace{N_x \begin{smallmatrix} E \\ W \end{smallmatrix} \circ \cdots \circ N_x \begin{smallmatrix} E \\ W \end{smallmatrix} \circ}_{|x_j|-1 \text{ times}} \circ M_x[i_j, N.E] \begin{smallmatrix} E \\ W \end{smallmatrix} \circ \\ & \underbrace{M_x[i_j, E.W] \begin{smallmatrix} E \\ W \end{smallmatrix} \circ \cdots \circ M_x[i_j, E.W] \begin{smallmatrix} E \\ W \end{smallmatrix} \circ}_{|x_{i_1} \cdots x_{i_{j-1}}|-1 \text{ times}} \circ BM_x[i_j, E.W] \begin{smallmatrix} E \\ S \end{smallmatrix} \\ & \text{(for even } j < n), \end{aligned}$$

$$\begin{aligned} wx_j = & BM_x[i_j, E.W] \begin{smallmatrix} N \\ E \end{smallmatrix} \circ \underbrace{M_x[i_j, E.W] \begin{smallmatrix} W \\ E \end{smallmatrix} \circ \cdots \circ M_x[i_j, E.W] \begin{smallmatrix} W \\ E \end{smallmatrix} \circ}_{|x_{i_1} \cdots x_{i_{j-1}}|-1 \text{ times}} \\ & M_x[i_j, N.E] \begin{smallmatrix} W \\ E \end{smallmatrix} \circ \underbrace{N_x \begin{smallmatrix} W \\ E \end{smallmatrix} \circ \cdots \circ N_x \begin{smallmatrix} W \\ E \end{smallmatrix} \circ}_{|x_j|-1 \text{ times}} \\ & M_x[i_{j+1}, N.E] \begin{smallmatrix} W \\ E \end{smallmatrix} \circ \underbrace{N_x \begin{smallmatrix} W \\ E \end{smallmatrix} \circ \cdots \circ N_x \begin{smallmatrix} W \\ E \end{smallmatrix} \circ}_{|x_{j+1}|-1 \text{ times}} \\ & \dots \\ & M_x[i_{n-1}, N.E] \begin{smallmatrix} W \\ E \end{smallmatrix} \circ \underbrace{N_x \begin{smallmatrix} W \\ E \end{smallmatrix} \circ \cdots \circ N_x \begin{smallmatrix} W \\ E \end{smallmatrix} \circ}_{|x_{n-1}|-1 \text{ times}} \\ & \underbrace{N_x \begin{smallmatrix} W \\ E \end{smallmatrix} \circ \cdots \circ N_x \begin{smallmatrix} W \\ E \end{smallmatrix} \circ}_{|x_n|-2 \text{ times}} \circ E_x[i_n, N.S] \begin{smallmatrix} W \\ S \end{smallmatrix} \\ & \text{(for odd } j < n), \end{aligned}$$

$$wx_n = E_x[i_n, N.W] \begin{smallmatrix} N \\ W \end{smallmatrix} \circ \underbrace{E_x[i_n, E.W] \begin{smallmatrix} E \\ W \end{smallmatrix} \circ \cdots \circ E_x[i_n, E.W] \begin{smallmatrix} E \\ W \end{smallmatrix} \circ}_{|x_{i_1} \cdots x_{i_n}|-2 \text{ times}} \circ BE_x[i_n, E.W] \begin{smallmatrix} E \\ S \end{smallmatrix}$$

$$\begin{aligned}
& \text{(if } n \text{ is even),} \\
wx_n &= BE_x[i_n, E.W]_E^N \circ \underbrace{E_x[i_n, E.W]_E^W \circ \dots \circ E_x[i_n, E.W]_E^W}_{|x_{i_1} \dots x_{i_n}| - 2 \text{ times}} \circ E_x[i_n, N.W]_S^W \\
& \text{(if } n \text{ is odd).}
\end{aligned}$$

In the same way we define figures wy_1, \dots, wy_n .

It is easy to see that $wx_1 \circ \dots \circ wx_n = wy_1 \circ \dots \circ wy_n \subseteq X^\circ$. Hence \mathbf{X} is not a UD code.
(End of proof of Lemma 1.) □

Example 7. Consider

$$\begin{aligned}
\Sigma &= \{a, b\}, \\
X &= (x_1, x_2, x_3) = (a, ba, bab), \\
Y &= (y_1, y_2, y_3) = (ab, aba, b).
\end{aligned}$$

We have $x_1x_2x_3 = y_1y_2y_3$. Figure f with two different tilings with elements of \mathbf{X} is presented in Figure 10 and Figure 11 (where thick arrows show the flow of information through annex-figures).

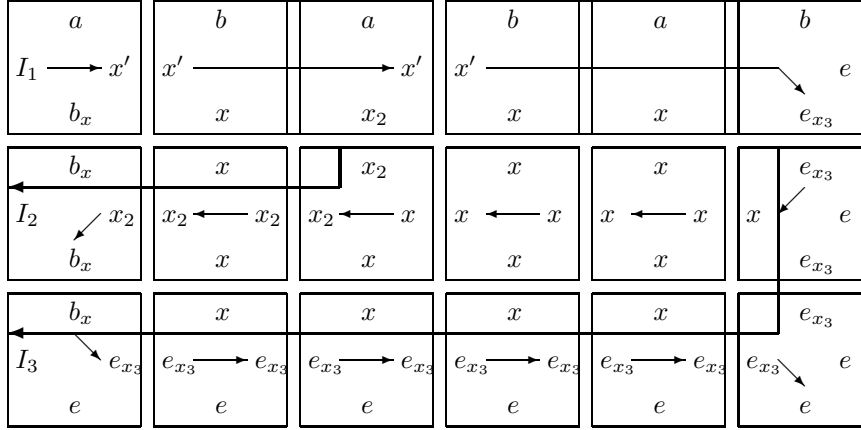
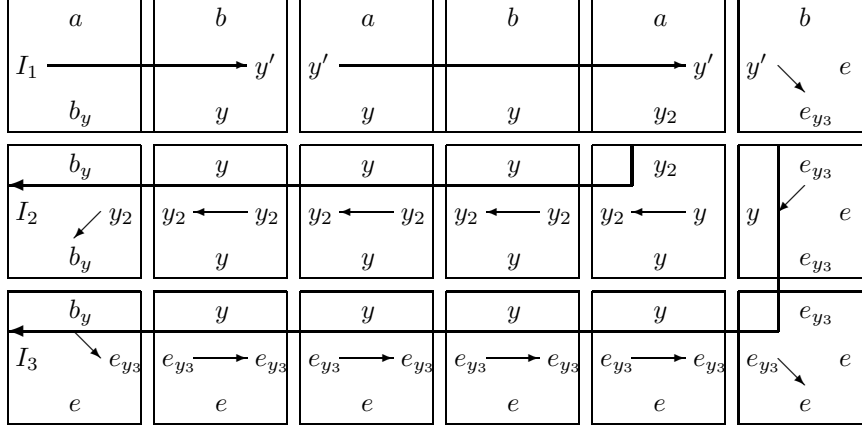


Fig. 10: “X”-tiling of f .

Lemma 2. If \mathbf{X} is not a UD (MSD, SD) code then the related PCP instance has a solution.

Proof of Lemma 2: Let f be a figure of minimal size (with respect to the size of its domain) which admits two tilings with elements of \mathbf{X} , i.e. there exist $f_1, \dots, f_p, g_1, \dots, g_q \in \mathbf{X}$ such that $f_1 \neq g_1$ and $f_1 \circ \dots \circ f_p = g_1 \circ \dots \circ g_q$.

Consider directed hooked squares tiling the figure f (these are annex-figures and squares of which basic-figures are built). Let d be the westernmost among the northernmost of them. We have following possibilities:

Fig. 11: “Y”-tiling of f .

- *Case 1:* $d \in \bigcup_{z \in \{x, y\}} \bigcup_{j \in \{1, \dots, k\}} \{E_z[j, N.S]_W^N, E_z[j, N.W]_W^N, E_z[j, E.W]_W^N\}$

Since d is the westernmost among the northernmost of all squares tiling f , it cannot have north and west neighbour squares, *i.e.* squares hooked to it at the north and west sides, respectively. Hence $f = d$, which contradicts the definition of the double tiling of f .
- *Case 2:* $d \in \bigcup_{z \in \{x, y\}} \bigcup_{j \in \{1, \dots, k\}} \{E_z[j, N.S]_S^W, E_z[j, N.W]_S^W\}$

Since d has no north and west neighbours, north and west hooks of d are uniquely determined by f . Each of figures listed is uniquely determined by its north and west hooks. Hence d is also uniquely determined by f . Now d has no west neighbour and it has the start point at its west side, which implies that it must be the first one in a sequence of figures whose catenation gives f , *i.e.* $d = f_1 = g_2$. Then either $f = d$ (contradiction as previously), or $f' = f_2 \circ \dots \circ f_p = g_2 \circ \dots \circ g_q$ is a smaller figure with two tilings, which contradicts the minimality of f .
- *Case 3:* $d \in \bigcup_{z \in \{x, y\}} \bigcup_{j \in \{1, \dots, k\}} \{M_z[j, N.S]_W^E, M_z[j, N.W]_W^E, M_z[j, E.W]_W^E\} \cup \{N_x \square_W^E, N_y \square_W^E\}$

As in *Case 1*, d is uniquely determined by f . Since d has no west neighbour and it has the end point at its west side, it must be the last one in a sequence of figures whose catenation gives f , *i.e.* $d = f_p = g_q$. Then either $f = d$ (contradiction as previously), or $f' = f_1 \circ \dots \circ f_{p-1} = g_1 \circ \dots \circ g_{q-1}$ is a smaller figure with two tilings, which contradicts the minimality of f .
- *Case 4:* $d \in \bigcup_{z \in \{x, y\}} \bigcup_{j \in \{1, \dots, k\}} \{BM_z[j, E.W]_E^N, BE_z[j, E.W]_E^N\}$

Now d must be the first one in the tiling since it has the start point at its north side and it is the northernmost in the tiling. Observe that there exists no square with e -hook at the north side. Hence $BM_z[j, E.W]_E^N$ and $BE_z[j, E.W]_E^N$ ($z \in \{x, y\}$) cannot be the first elements of two different tiling sequences of f . Consequently, d is uniquely determined by f and $d = f_1 = g_1$. Contradiction as in *Case 2*.

- *Case 5:* $d \in \bigcup_{z \in \{x, y\}} \bigcup_{j \in \{1, \dots, k\}} \{BM_z[j, E.W]_S^E, BE_z[j, E.W]_S^E\}$

As in *Case 4*, d is uniquely determined by f . If $d = BE_z[j, E.W]_S^E$ (for $z \in \{x, y\}$) then d is the last element of a tiling sequence. Contradiction as in *Case 3*. If $d = BM_z[j, E.W]_S^E$ (for $z \in \{x, y\}$) then (for some $i \in \{1, \dots, p\}$) $f = f_1 \circ f_2 \circ \dots \circ f_{i-1} \circ d \circ f_{i+1} \circ \dots \circ f_p$, where $f_1 \in \{M_z[j, E.W]_W^E, M_z[j, N.W]_W^E\}$ and $f_2 = \dots = f_{i-1} = M_z[j, E.W]_W^E$. Contradiction as in *Case 1*.

This leads us to a conclusion that:

- *Case 6:* Directed hooked square d is a part of a basic-figure. In particular, d is a “first part” of f_1 and g_1 .

Now it is easy to observe the following properties of f 's tiling:

1. If f_1 is a figure that encodes one of the words from X , then all f_i ($i \in \{1, \dots, p\}$) are figures encoding “ x -part” of the related PCP instance (since there is no figure that links a figure from “ x -part” with a figure from “ y -part”). In the same way, if f_1 encodes a word from Y , then all f_i encode “ y -part” of the PCP instance. A similar statement holds for g_i ($i \in \{1, \dots, q\}$).
2. First “row” of figures in the tiling is a sequence of middle solution figures (may be empty) which is ended by an end solution figure (that ends the row) and may be started with a begin solution figure.
3. The sequence of middle solution figures from the first row implies that in the tiling, leftmost column's hooks (I_j hooks of some BM and BE annex-figures) correspond to the sequence of indices of words encoded by those figures.

This leads us to a simple observation that the only possible two tilings of f are tilings of the form defined in the proof of Lemma 1. Hence the related PCP instance has a solution.

(*End of proof of Lemma 2.*) □

(*Proof of Theorem 4, continued.*)

Lemmas 1 and 2 complete the proof for UD, as well as MSD and SD codes, since it is clear that exactly the same reasoning can be applied in the MSD and SD cases. ND codes, however, have to be dealt with separately, since both factorizations have exactly the same number of figures. An additional technique to handle the ND case is as follows: replace basic directed hooked squares for both “ x -part” and “ y -part” with 25 squares. In the “ x -part” the 25 squares will be connected (into one figure), while in the “ y -part” they will be disconnected. See Figure 12 and Figure 13, where a construction is presented for two kinds of figures. In both figures, p and p_i are new symbols, different for each original directed hooked square. Other kinds of figures can be dealt with in a similar way.

Observe that the construction for UD, MSD and SD codes actually uses vectors from a closed half-plane only. The construction for ND codes can also be carried out in this way; however, more complicated encoding figures are required then. □

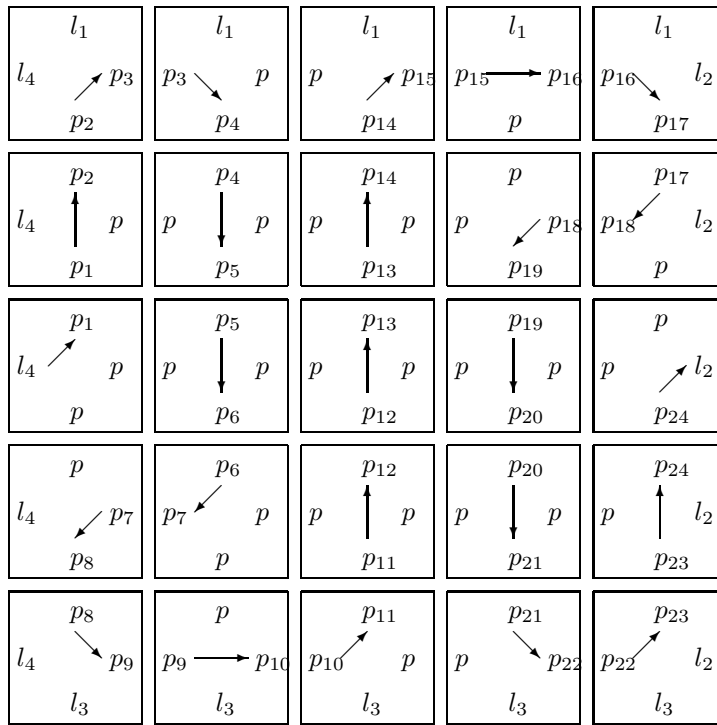


Fig. 12: Replacement figures for $DHS(l_1, l_2, l_3, l_4)_E^W$.

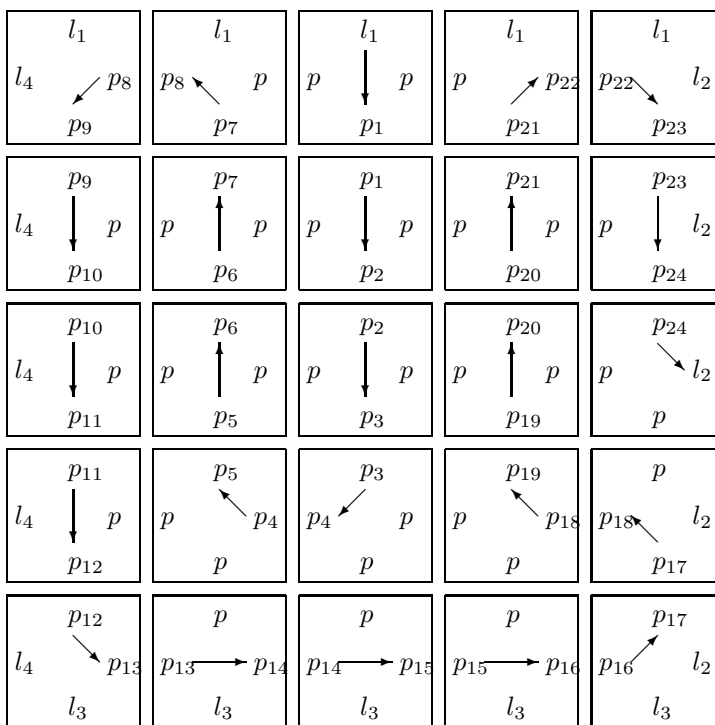


Fig. 13: Replacement figures for $\text{DHS}(l_1, l_2, l_3, l_4)_E^N$.

4.3 Summary of decidability results

The following table summarizes the status of decipherability decidability. Decidable cases are marked with a +, undecidable ones with a –. Combinations that are still open are denoted with a question mark.

		UD	MSD	ND	SD
1	One-sided codes	+	+	+	+
2	One-sided m -codes	+	+	+	+
3	Two-sided codes	–	–	–	–
4	Two-sided m -codes	+	+	+	?
5	Two-sided codes with parallel vectors	+	+	+	+
6	Two-sided m -codes with parallel vectors	+	+	+	?

5 Final remarks

Note that the positive decidability cases depicted in lines 4 and 6 (of the table in Section 4.3) are trivial. By Theorem 1, two-sided UD, MSD or ND m -codes do not exist. For other decidable combinations, respective proofs lead to effective verification algorithms.

On the other hand, the case of two-sided SD m -codes is non-trivial; both SD and non-SD codes of this kind exist. However, none of the proof techniques we have used so far can be adapted to this case.

References

- P. Aigrain and D. Beauquier. Polyomino tilings, cellular automata and codicity. *Theoretical Computer Science*, 147(1–2):165–180, 1995.
- M. Anselmo, D. Giammarresi, and M. Madonia. Two dimensional prefix codes of pictures. In *Developments in Language Theory*, volume 7907 of *Lecture Notes in Computer Science*, pages 46–57. Springer, 2013a.
- M. Anselmo, D. Giammarresi, and M. Madonia. Strong prefix codes of pictures. In *Algebraic Informatics*, volume 8080 of *Lecture Notes in Computer Science*, pages 47–59. Springer, 2013b.
- D. Beauquier and M. Nivat. A codicity undecidable problem in the plane. *Theoretical Computer Science*, 303(2–3):417–430, 2003.
- F. Blanchet-Sadri. On unique, multiset, set decipherability of three-word codes. *IEEE Transactions on Information Theory*, 47(5):1745–1757, 2001.
- F. Blanchet-Sadri and C. Morgan. Multiset and set decipherable codes. *Computers and Mathematics with Applications*, 41(10–11):1257–1262, 2001.
- F. Burderi and A. Restivo. Coding partitions. *Discrete Mathematics and Theoretical Computer Science*, 9(2):227–240, 2007a.
- F. Burderi and A. Restivo. Varieties of codes and Kraft inequality. *Theory of Computing Systems*, 40(4):507–520, 2007b.

- G. Costagliola, F. Ferrucci, and C. Gravino. Adding symbolic information to picture models: definitions and properties. *Theoretical Computer Science*, 337:51–104, 2005.
- D. Giammarresi and A. Restivo. Two-dimensional finite state recognizability. *Fundamenta Informaticae*, 25(3):399–422, 1996.
- F. Guzmán. Decipherability of codes. *Journal of Pure and Applied Algebra*, 141(1):13–35, 1999.
- T. Head and A. Weber. The finest homophonic partition and related code concepts. In *Mathematical Foundations of Computer Science MFCS 1994*, volume 841 of *Lecture Notes in Computer Science*, pages 618–628. Springer, 1994.
- T. Head and A. Weber. Deciding multiset decipherability. *IEEE Transactions on Information Theory*, 41(1):291–297, 1995.
- M. Kolarz. Directed figure codes: Decidability frontier. In *COCOON 2010*, volume 6196 of *Lecture Notes in Computer Science*, pages 530–539. Springer, 2010a.
- M. Kolarz. The code problem for directed figures. *Theoretical Informatics and Applications RAIRO*, 44(4):489–506, 2010b.
- M. Kolarz and W. Moczurad. Directed figure codes are decidable. *Discrete Mathematics and Theoretical Computer Science*, 11(2):1–14, 2009.
- M. Kolarz and W. Moczurad. Multiset, set and numerically decipherable codes over directed figures. In *IWOCA 2012*, volume 7643 of *Lecture Notes in Computer Science*, pages 224–235. Springer, 2012.
- A. Lempel. On multiset decipherable codes. *IEEE Transactions on Information Theory*, 32(5):714–716, 1986.
- S. Mantaci and A. Restivo. Codes and equations on trees. *Theoretical Computer Science*, 255:483–509, 2001.
- W. Moczurad. Brick codes: families, properties, relations. *International Journal of Computer Mathematics*, 74:133–150, 2000.
- W. Moczurad. Directed figure codes with weak equality. In *Intelligent Data Engineering and Automated Learning IDEAL 2010*, volume 6283 of *Lecture Notes in Computer Science*, pages 242–250. Springer, 2010.
- W. Moczurad. Domino graphs and the decipherability of directed figure codes. In *IWOCA 2013*, volume 8288 of *Lecture Notes in Computer Science*, pages 453–457. Springer, 2013.
- A. Restivo. A note on multiset decipherable code. *IEEE Transactions on Information Theory*, 35(3):662–663, 1989.
- A. Salomaa, K. Salomaa, and S. Yu. Variants of codes and indecomposable languages. *Information and Computation*, 207(11):1340–1349, 2009.