Schedae Informaticae Vol. 22 (2013): 27–40
doi: 10.4467/20838476SI.13.003.2088

Testing decipherability of directed figure codes with domino graphs[1]

Małgorzata Moczurad, Włodzimierz Moczurad
Institute of Computer Science, Jagiellonian University,
Prof. Stanisława Łojasiewicza 6, 30-348 Kraków, Poland
e-mail: *mmoczurad@ii.uj.edu.pl, wkm@ii.uj.edu.pl*

Abstract. Various kinds of decipherability of codes, weaker than unique decipherability, have been studied since 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 setting extends decipherability questions from words to 2D structures. In the present paper we develop a (variant of) domino graph that will allow us to decide some of the decipherability kinds by searching the graph for specific paths. Thus the main result characterizes directed figure decipherability by graph properties.

1. Introduction

The term *unique decipherability* refers to a property of a set of words X whereas every message composed from these words can be uniquely decoded, i.e. an exact sequence of words is recovered; this corresponds to X^* being free over X. The set X is then called a *uniquely decipherable (UD) code*; the term *code* alone is also used. Words in X are often called *codewords*.

However, in some applications it might be sufficient to decode the message with respect to a feature weaker than the exact sequence of codewords – like the multiset, the set or just the number of codewords – giving rise to three kinds of decipherability, known as *multiset* (MSD), *set* (SD) and *numeric decipherability* (ND), respectively.

Multiset decipherability was introduced by Lempel in [1], whilst numeric decipherability originates in [2] by Head and Weber. The same authors in [3] develop

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

"domino graphs" providing a useful technique for decipherability verification. A paper by Guzman [4] defined set decipherability and presented a unifying approach to different decipherability notions using varieties of monoids. Decipherability literature is already quite rich and includes e.g. papers by Restivo [5], Blanchet-Sadri and Morgan [6], Blanchet-Sadri [7], Burderi and Restivo [8, 9] and Salomaa et al. [10]; the latter is not directly concerned with decipherability, but uses ND codes (under the name of *length codes*) to study prime decompositions of languages.

Extensions of classical words and codes have also been widely studied. For instance, Aigrain and Beauquier introduced polyomino codes in [11]; two-dimensional rectangular pictures were studied by Giammarresi and Restivo in [12], whilst in [13] Mantaci and Restivo described an algorithm to verify tree codes. The interest in picture-like structures is not surprising, given the huge amounts of pictorial data in use. Unfortunately, properties related to decipherability rarely carry over. In particular, decipherability testing (i.e. testing whether a given set is a UD code) is undecidable for polyominoes and similar structures, cf. [14, 15].

In [16] 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. in [17], 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 [18]) and is a significant change in comparison to previously mentioned picture models, facilitating the use of directed figures for e.g. 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 [19, 20].

In [21] we extended the previous results by considering not just UD codes, but also MSD, SD and ND codes over directed figures. We proved 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 remained open.

In the present paper we define a variant of domino graphs that allows us to decide some of the decipherability kinds by searching the graph for specific paths. Thus the main result characterizes directed figure decipherability by graph properties.

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 that kinds. In Section 4. we summarize existing decidability results for decipherability verification. Finally, in Section 5. we define the domino graph and state the main result.

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, $\mathrm{tr}_u : \mathbb{Z}^2 \ni (x, y) \mapsto (x + u_x, y + u_y) \in \mathbb{Z}^2$. By extension, for a set $V \subseteq \mathbb{Z}^2$ and an arbitrary function $f : V \to \Sigma$ define $\mathrm{tr}_u : \mathcal{P}(\mathbb{Z}^2) \ni V \mapsto \{\mathrm{tr}_u(v) \mid v \in V\} \in \mathcal{P}(\mathbb{Z}^2)$ and $\mathrm{tr}_u : \Sigma^V \ni f \mapsto f \circ \mathrm{tr}_{-u} \in \Sigma^{\mathrm{tr}_u(V)}$.

Definition 1 (Directed figure, cf. [16]) *Let $D \subseteq \mathbb{Z}^2$ be finite and non-empty, $b, e \in \mathbb{Z}^2$ and $l : D \to \Sigma$. A quadruple $f = (D, b, e, l)$ is a* directed figure *(over Σ) with*

domain	$\mathrm{dom}(f)$	$=$	D,
start point	$\mathrm{begin}(f)$	$=$	b,
end point	$\mathrm{end}(f)$	$=$	e,
labelling function	$\mathrm{label}(f)$	$=$	l.

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

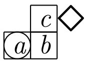
The set of all directed figures over Σ is denoted by $\Sigma^\diamond$. Two directed figures x, y are *equal* (denoted by $x = y$) if there exists $u \in \mathbb{Z}^2$ such that

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

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.*

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



Definition 2 (Catenation) *Let $x = (D_x, b_x, e_x, l_x)$ and $y = (D_y, b_y, e_y, l_y)$ be directed figures. If $D_x \cap \mathrm{tr}_{e_x - b_y}(D_y) = \emptyset$, a* catenation *of x and y is defined as*

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

where

$$l(z) = \begin{cases} l_x(z) & \text{for } z \in D_x, \\ \mathrm{tr}_{e_x - b_y}(l_y)(z) & \text{for } z \in \mathrm{tr}_{e_x - b_y}(D_y). \end{cases}$$

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

Definition 3 (m-catenation) *Let $x = (D_x, b_x, e_x, l_x)$ and $y = (D_y, b_y, e_y, l_y)$ be directed figures. An* m-catenation *of x and y with respect to a* merging function $m : \Sigma \times \Sigma \to \Sigma$ *is defined as*

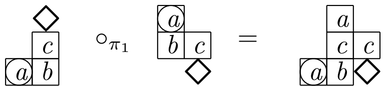
$$x \circ_m y = (D_x \cup \mathrm{tr}_{e_x - b_y}(D_y), b_x, \mathrm{tr}_{e_x - b_y}(e_y), l),$$

where

$$l(z) = \begin{cases} l_x(z) & \text{for } z \in D_x \setminus \mathrm{tr}_{e_x - b_y}(D_y), \\ \mathrm{tr}_{e_x - b_y}(l_y)(z) & \text{for } z \in \mathrm{tr}_{e_x - b_y}(D_y) \setminus D_x, \\ m(l_x(z), \mathrm{tr}_{e_x - b_y}(l_y)(z)) & \text{for } z \in D_x \cap \mathrm{tr}_{e_x - b_y}(D_y). \end{cases}$$

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.*



Observe that $\circ$ is associative, whilst $\circ_m$ is associative if and only if m is associative. Thus for associative m, $\Sigma^\diamond_m = (\Sigma^\diamond, \circ_m)$ is a monoid (which is never free).

Abusing this notation, we also write $X^\diamond$ (resp. $X^\diamond_m$) to denote the set of all figures that can be composed by $\circ$ catenation (resp. $\circ_m$ m-catenation) from figures in $X \subseteq \Sigma^\diamond$. 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, "$x \in X^\diamond_\bullet$" should be read as "$x \in X^\diamond$ (resp. $x \in X^\diamond_m$)".

From now on let m be an arbitrary associative merging function.

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 $\Sigma^\diamond \setminus \{\varepsilon\}$.

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

Definition 5 (UD m-code) *Let X be a code over Σ. X is a* uniquely decipherable m-code, *if for any $x_1, \ldots, x_k, y_1, \ldots, y_l \in X$ the equality $x_1 \circ_m \cdots \circ_m x_k = y_1 \circ_m \cdots \circ_m y_l$ implies that $(x_1, \ldots, x_k)$ and $(y_1, \ldots, y_l)$ 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 $x_1, \ldots, x_k,\ y_1, \ldots, y_l \in X$ the equality $x_1 \bullet \cdots \bullet x_k = y_1 \bullet \cdots \bullet y_l$ implies that $\{\ \{x_1, \ldots, x_k\}\ \}$ and $\{\ \{y_1, \ldots, y_l\}\ \}$ 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 $x_1, \ldots, x_k,\ y_1, \ldots, y_l \in X$ the equality $x_1 \bullet \cdots \bullet x_k = y_1 \bullet \cdots \bullet y_l$ implies that $\{x_1, \ldots, x_k\}$ and $\{y_1, \ldots, y_l\}$ 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 $x_1, \ldots, x_k,\ y_1, \ldots, y_l \in X$ the equality $x_1 \bullet \cdots \bullet x_k = y_1 \bullet \cdots \bullet y_l$ 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. The converse does not hold.*

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.*

The diagram illustrates inclusions between different families of codes, with all inclusions strict. A similar diagram can be made for m-codes.

$$\begin{tikzcd}
& \wp(\Sigma^\diamond) & \\
\mathrm{SD} \arrow[ur] & & \mathrm{ND} \arrow[ul] \\
& \mathrm{MSD} \arrow[ul] \arrow[ur] & \\
& \mathrm{UD} \arrow[u] & \\
& \emptyset \arrow[u] &
\end{tikzcd}$$

Definition 9 (Two-sided and one-sided codes) *Let $X = \{x_1, \ldots, x_n\}$ be a code over Σ. If there exist $\alpha_1, \ldots, \alpha_n \in \mathbb{N}$, not all equal to zero, such that $\sum_{i=1}^{n} \alpha_i \mathrm{tran}(x_i)$ $= (0,0)$, then X is called* two-sided*. Otherwise it 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.

Theorem 1 (cf. [21]) *Let X be a code over Σ. If X is two-sided then X is not an ND m-code (and consequently neither an MSD nor UD m-code).*

4. Summary of decidability results

In this section we summarize all non-trivial decidability results for the decipherability verification. We also quote Theorem 4 of [21], including the proof, since we will use the definitions of a *configuration* and a *reduced configuration* contained there.

In the following table 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	$+$	$+$	$+$	?

Theorem 2 (cf. [21]) *Let X be a one-sided code over Σ. It is decidable whether X is a $\{$UD, MSD, SD or ND$\}$ $\{$code or m-code$\}$.*

Proof 3 *Let $X = \{x_1, \ldots, x_n\} \subseteq \Sigma^\diamond$ and let $\mathrm{begin}(x) = (0,0)$ for each $x \in X$. Since X is one-sided, there exists a vector τ such that*

$$\forall x \in X : \tau \cdot \mathrm{tran}(x) > 0.$$

We can assume that figures are sorted by angle in the following way:

$$\angle(R_{-\frac{\pi}{2}}(\tau), \mathrm{tran}(x_1)) \leq \angle(R_{-\frac{\pi}{2}}(\tau), \mathrm{tran}(x_2)) \leq \ldots \leq \angle(R_{-\frac{\pi}{2}}(\tau), \mathrm{tran}(x_n)),$$

($\angle$ denotes an angle between two vectors, R_ϕ denotes a rotation by ϕ).

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

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

define a "bounding area" for figures in X, i.e.,

$$\forall x \in X : \mathrm{dom}(x) \cup \{\mathrm{end}(x)\} \subseteq \bigcap_{u \in \{\tau_E, \tau_N, \tau_W \tau_S\}} \{\mathrm{HP}(u, \mathrm{begin}(x))\},$$

where for $u, v \in \mathbb{Z}^2$, $\mathrm{HP}(u,v)$ denotes a half-plane $\{w \in \mathbb{Z}^2 \mid u \cdot (w - (v+u)) \leq 0\}$.

For $x \in X^\diamond_\bullet$ define

$$\begin{aligned} \mathrm{CE}^+(x) &= \mathrm{HP}(\tau_s, \mathrm{end}(x)) \cap \mathrm{HP}(\tau_n, \mathrm{end}(x)) \cap \mathrm{HP}(\tau_w, \mathrm{end}(x)), \\ \mathrm{CE}^-(x) &= \mathbb{Z}^2 \setminus \mathrm{CE}^+(x), \\ \mathrm{CW}^+(x) &= \bigcup_v \{v + (\mathrm{CE}^+(x) \cap \mathrm{HP}(\tau_e, \mathrm{end}(x)))\}, \\ \mathrm{CW}^-(x) &= \mathbb{Z}^2 \setminus \mathrm{CW}^+(x), \end{aligned}$$

where the union in the definition of $\mathrm{CW}^{+}(x)$ is taken over $v \in \mathbb{Z}^2$ lying within an angle spanned by vectors $-\tau(x_1)$ and $-\tau(x_n)$.

Immediately from the definition we have following properties, for $x, y \in X^{\diamond}_{\bullet}$:

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

For $x_1, \ldots, x_k$, $y_1, \ldots, y_l \in X^{\diamond}_{\bullet}$ we define a configuration as a pair of sequences $((x_1, \ldots, x_k), (y_1, \ldots, y_l))$. A successor of such configuration is either $((x_1, \ldots, x_k, z), (y_1, \ldots, y_l))$ or $((x_1, \ldots, x_k), (y_1, \ldots, y_l, z))$ 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 $C = ((x_1, \ldots, x_k), (y_1, \ldots, y_l))$ let us denote:

$$\begin{array}{rcl} L(C) & = & \{x_1, \ldots, x_k\}, \\ L_{\bullet}(C) & = & x_1 \bullet \ldots \bullet x_k, \\ R(C) & = & \{y_1, \ldots, x_l\}, \\ R_{\bullet}(C) & = & y_1 \bullet \ldots \bullet y_l. \end{array}$$

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 know that X is not a UD (m-)code and we have the following conditions for other decipherability kinds:

$$\left.\begin{array}{l} \bullet\ \text{if } L(C) = R(C) \text{ as multisets then } X \text{ is not an MSD (m-)code,} \\ \bullet\ \text{if } L(C) = R(C) \text{ as sets then } X \text{ is not an SD (m-)code,} \\ \bullet\ \text{if } |L(C)| = |R(C)| \text{ then } X \text{ is not an ND (m-)code.} \end{array}\right\} \quad (1)$$

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 (m-)code of each kind. In the latter case, if we find one of such configurations, X is already not a UD (m-)code. To verify whether X is an MSD, SD or ND (m-)code, we have to check the above conditions for all possible proper configurations.

Let

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

where for $u = (u_x, u_y) \in \mathbb{Z}^2$ and $n \in \mathbb{N}$, $\mathrm{B}(u, n)$ denotes a ball on integer grid with center u and radius r, i.e.,

$$\mathrm{B}(u, n) = \{(v_x, v_y) \in \mathbb{Z}^2 \mid \ |u_x - v_x| + |u_y - v_y| \leq n\}.$$

The following properties of a proper configuration C are easily verified:

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

$$\mathrm{B}(\mathrm{end}(R_{\bullet}(C)), \rho) \cap (\mathrm{CW}^{+}(L_{\bullet}(C)) \cup \mathrm{CE}^{+}(L_{\bullet}(C))) \neq \emptyset, \quad (3)$$

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

$$\mathrm{label}(L_\bullet(C)) \mid_D \equiv \mathrm{label}(R_\bullet(C)) \mid_D . \tag{4}$$

Notice that we do not need all of the information contained in configurations, just those labellings that can be changed by future catenations. By (4), 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*

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

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

$$|\tau_e \cdot (\mathrm{end}(L_\bullet(C)) - \mathrm{end}(R_\bullet(C)))| \leq |\tau_e|^2, \tag{5}$$

since no single figure advances $\mathrm{end}(L_\bullet(C))$ *or* $\mathrm{end}(R_\bullet(C))$ *by more than* $|\tau_e|$*. Moreover, (2) and (3) restrict the perpendicular span (in the direction of* $R_{-\frac{\pi}{2}}(\tau_e)$*). 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 (m-)code.*

Conditions (2), (3), (4) and (5) appearing in the above proof will be called *RC criteria.*

5. Domino graphs for decipherability testing

We now develop a variant of the domino graph as introduced by Head and Weber in [2, 3]. It will allow us to decide some of the decipherability types by searching the graph for specific paths.

Throughout this section we fix a "merging type" (i.e. either a merging function m, or no merging function) and use it for all catenations. Note that reduced configurations, and hence the domino graph, depend on it. We also assume that all codes are one-sided, since reduced configurations are not defined for two-sided codes.

Let $rc(C)$ denote the reduced configuration associated with a configuration C. Given a figure $z \in X$ we define an *extension of a reduced configuration* $rc((x_1, \ldots, x_k),$ $(y_1, \ldots, y_l))$ *by* (z, ε) as a new reduced configuration $rc((x_1, \ldots, x_k, z), (y_1, \ldots, y_l))$. It is clear that the extension is well-defined, since $rc((x_1, \ldots, x_k), (y_1, \ldots, y_l)) = rc((x'_1, \ldots, x'_{k'}), (y'_1, \ldots, y'_{l'}))$ implies $rc((x_1, \ldots, x_k, z), (y_1, \ldots, y_l)) = rc((x'_1, \ldots, x'_{k'}, z), (y'_1, \ldots, y'_{l'}))$. Extension by (ε, z) is defined similarly. Note that in the non-merging case a particular extension may be undefined.

A reduced configuration, as defined in Theorem 2, is a pair $((e_L, l_L), (e_R, l_R))$ with end points $e_L, e_R \in \mathbb{Z}^2$ and labellings l_L, l_R which are partial mappings $\mathbb{Z}^2 \to \Sigma$. Informally, the extension of $((e_L, l_L), (e_R, l_R))$ by (z, ε) is the reduced configuration

$((e'_L, l'_L), (e_R, l_R))$, where $e'_L = e_L + \mathrm{tran}(z)$ and l'_L is obtained by "catenating" l_L with z and constraining the domain appropriately.

A reduced configuration is called *final*, if it is of the form $((e, l), (e, l))$, i.e. its left and right components are equal. Note that $rc(C)$ is final iff $L_\bullet(C) = R_\bullet(C)$.

Let $RC(X)$ be the set of all reduced configurations over X which satisfy the RC criteria, i.e. $RC(X) = \{rc((x_i), (y_j)) \mid (x_i), (y_j)\}$, with (x_i) and (y_j) ranging over all finite, non-empty sequences of elements of X satisfying the RC criteria. By Theorem 2, $RC(X)$ is finite for every one-sided code X.

Definition 10 (Domino graph) *Let X be a one-sided code over Σ. A* domino graph *of X is the directed graph (V, E) with $V = RC(X) \cup \{0\}$ and $E = E_0 \cup E_1$, where*

- *E_0 contains all edges $(0, v)$ such that $v \in RC(X)$ and $v = rc((x), (y))$ for some $x, y \in X$, $x \neq y$,*
- *E_1 contains all edges (v_1, v_2) such that $v_1, v_2 \in RC(X)$, v_1 is not final and v_2 is an extension of v_1 by (z, ε) or (ε, z), for some $z \in X$.*

Additionally, we define a domino function $d : E \to \wp((X \cup \{\varepsilon\}) \times (X \cup \{\varepsilon\}))$ *that associates labels to the edges:*

$$\begin{aligned} d(0, v) &= \{(x, y) \in X \times X \mid v = rc((x), (y))\} \\ d(v_1, v_2) &= \{(x, y) \in (X \times \{\varepsilon\}) \cup (\{\varepsilon\} \times X) \mid v_2 \textit{ is an extension of } v_1 \textit{ by } (x, y)\}. \end{aligned}$$

Observe that for an edge (v_1, v_2) with $v_1 \neq 0$, $d(v_1, v_2)$ either contains pairs of the form (z, ε), or (ε, z), but not both. Moreover, if for instance (z, ε) and $(z', \varepsilon) \in d(v_1, v_2)$ then $\mathrm{tran}(z) = \mathrm{tran}(z') \neq (0, 0)$, since X is one-sided and two reduced configurations v_1 and v_2 determine a unique translation vector required to extend v_1 to v_2. Hence, $d(0, v)$ are the only values of d that contain pairs with both figures non-empty.

The domino function can be extended to paths in a domino graph G: given a path $p = (e_1, e_2, \ldots, e_n)$, where e_i's are edges in G, define

$$d(p) = d(e_1) \bullet d(e_2) \bullet \cdots \bullet d(e_n),$$

with $\bullet$ denoting the obvious extension of figure catenation to sets of figure pairs, i.e. for $A, B \subseteq X \times X$

$$A \bullet B = \{(x_A \bullet x_B, y_A \bullet y_B) \mid (x_A, y_A) \in A, (x_B, y_B) \in B\}.$$

Given a path $p = (e_1, e_2, \ldots, e_n)$, we also define a *realization of p* to be any sequence of figure pairs $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ such that $(x_i, y_i) \in d(e_i)$. Note that $x_i, y_i \in X \cup \{\varepsilon\}$.

For a path p starting in the vertex 0, $d(p)$ describes an attempt at constructing two distinct factorizations of some figure. If p can be made to reach a final vertex, this is indeed accomplished (p is "successful") and we know that X is not a UD (m-)code. To check for other decipherability kinds, all successful paths have to be checked for specific properties, similar to conditions (1) in the proof of Theorem 2. This is reflected in the following theorem:

Theorem 4 *Let X be a one-sided code over Σ.*

1. *X is not a UD (m-)code iff the domino graph of X contains a path from 0 to a final vertex.*

2. *X is not an MSD (m-)code iff the domino graph of X contains a path p from 0 to a final vertex such that there exists a realization of p, $((x_1, y_1), \ldots, (x_n, y_n))$, with $\{\ \{x_1, \ldots, x_n\}\ \}$ and $\{\ \{y_1, \ldots, y_n\}\ \}$ being different as multisets.*

3. *X is not an SD (m-)code iff the domino graph of X contains a path p from 0 to a final vertex such that there exists a realization of p, $((x_1, y_1), \ldots, (x_n, y_n))$, with $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ being different as sets.*

4. *X is not an ND (m-)code iff the domino graph of X contains a path p from 0 to a final vertex such that there exists a realization of p, $((x_1, y_1), \ldots, (x_n, y_n))$, with the number of non-empty x_i's different than the number of non-empty y_i's.*

Proof 5 *(1) If X is not a UD (m-)code then there exist $x_1, \ldots, x_k$, $y_1, \ldots, y_l \in X$ such that $x_1 \bullet \cdots \bullet x_k = y_1 \bullet \cdots \bullet y_l$, with $(x_1, \ldots, x_k) \neq (y_1, \ldots, y_l)$. In particular, $x_1 \neq y_1$ can be taken.*

Now the path can be constructed by starting at 0, going to $rc((x_1), (y_1))$ and then adjoining consecutive x_i's or y_i's so that a bounded span is maintained within the configuration, as imposed by the RC criteria. More formally, the path is $(v_0, \ldots, v_{k+l-1})$, where $v_0 = 0$, $v_1 = rc((x_1), (y_1))$ and v_{i+1} is an extension of v_i by (x_s, ε) or (ε, y_t); the extension is chosen so that the span is kept within limit; x_s and y_t denote the next unused figure from the appropriate sequence. Clearly the path will eventually arrive at $v_{k+l-1} = rc((x_1, x_2, \ldots, x_k), (y_1, y_2, \ldots, y_l))$, which is final.

Note that we cannot allow the configurations to "grow" beyond the span limit. Hence the following construction could be invalid:

$$
\begin{array}{l}
0, \\
rc((x_1), (y_1)), \\
rc((x_1, x_2), (y_1)), \\
\ldots \\
rc((x_1, x_2, \ldots, x_k), (y_1)), \\
rc((x_1, x_2, \ldots, x_k), (y_1, y_2)), \\
\ldots \\
rc((x_1, x_2, \ldots, x_k), (y_1, y_2, \ldots, y_l)).
\end{array}
$$

Conversely, if the domino graph of X contains a path from 0 to a final vertex, take any realization of p, $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$, and define two sequences of figures by taking the x_i's with empty figures omitted and y_i's with empty figures omitted. Since the last vertex on the path is final, it follows that the two sequences have equal (m-)catenations, hence X is not a UD (m-)code.

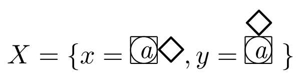
(2) If X is not an MSD (m-)code then there exist $x_1, \ldots, x_k$, $y_1, \ldots, y_l \in X$ such that $x_1 \bullet \cdots \bullet x_k = y_1 \bullet \cdots \bullet y_l$, with $\{\ \{x_1, \ldots, x_k\}\ \}$ and $\{\ \{y_1, \ldots, y_l\}\ \}$ being different as multisets. It is clear that a path can now be constructed as in (1) with a realization explicitly constructed to contain pairs of the form (x_i, ε) and (ε, y_i). Hence, $\{\ \{x_1, \ldots, x_n\}\ \} \neq \{\ \{y_1, \ldots, y_n\}\ \}$, $n = k + l - 1$.

Conversely, if the domino graph of X contains a path from 0 to a final vertex with a realization $((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n))$ such that $\{\ \{x_1, \ldots, x_n\}\ \} \neq \{\ \{y_1, \ldots, y_n\}\ \}$, two sequences of figures with equal (m-)catenations, but different as multisets, can be constructed. Hence X is not an MSD (m-)code.

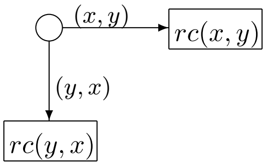
(3, 4) Analogous to (1).

The following examples show domino graphs for a UD m-code and a non-decipherable code. Both examples assume alphabet $\Sigma = \{a\}$ and a merging function $m = \{(a, a) \mapsto a\}$. Edge labels denote values of the domino function d; note that in both examples $|d(e)| = 1$ for all edges. For the sake of brevity, the notation of reduced configurations omits inner parentheses and commas.

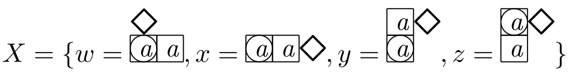
Example 3 *Consider*



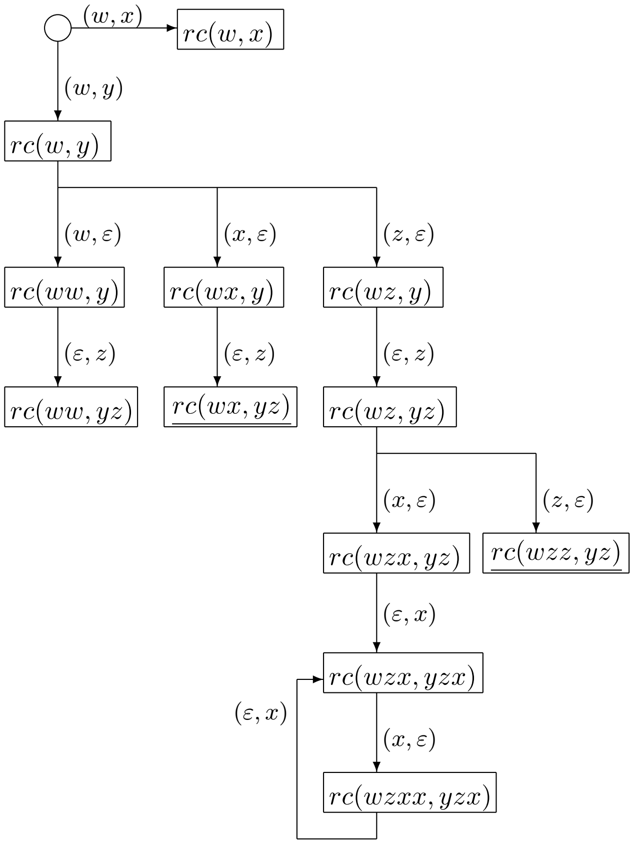
and take $\tau_E = (1, 1), \tau_W = (-\frac{1}{2}, -\frac{1}{2}), \tau_S = (0, -\frac{1}{2}), \tau_N = (-\frac{1}{2}, 0)$. Now $RC(X)$ contains just two elements, $rc(x, y)$ and $rc(y, x)$. Further candidates, i.e. (xx, y), (x, yx), (xy, y), (x, yy) and their symmetrical pairs, fail the RC criteria. Thus the domino graph of X, depicted below, contains no final vertices and X is a UD m-code.



Example 4 *Consider*



and set $\tau_E = (1, 1), \tau_W = (-\frac{1}{2}, -\frac{1}{2}), \tau_S = (0, -1), \tau_N = (-\frac{1}{2}, 0)$. In this example we omit pairs that can be obtained from another pair by exchanging the elements; this does not prevent us from discovering any of the properties characterized by Theorem 4. Final vertices are underlined. Note that the graph contains two successful paths, $0 \to rc(w, y) \to rc(wx, y) \to rc(wx, yz)$ and $0 \to rc(w, y) \to rc(wz, y) \to rc(wz, yz) \to rc(wzz, yz)$. The former disproves UD, MSD and SD decipherability of X (but not ND); the latter disproves all four decipherability kinds.



6. Final remarks

Theorem 4 allows to express some of the decipherability kinds in terms of domino graph properties. Hence, decipherability verification can now be performed as a graph search. Note that the graph searches are obviously polynomial in the size of the domino graph, but the size of the graph is hard to estimate. It depends, for instance, on the maximum angle α spanned by translation vectors of figures. When α tends to 0, the graph becomes smaller, and when $\alpha = 0$, catenations resemble catenations of words and the vertices of the graph correspond to suffixes, giving an obvious size bound $|V| = O(\|X\|)$, where $\|X\|$ is the total size of figures in X. However, when α increases, the graph grows as well, and in general $|V| \to \infty$ as $\alpha \to \pi$.

7. References

[1] Lempel A.; *On multiset decipherable codes*, IEEE Transactions on Information Theory 32(5), 1986, pp. 714–716.

[2] Head T., Weber A.; *The finest homophonic partition and related code concepts.* In: *Mathematical Foundations of Computer Science MFCS 1994*, vol. 841 of Lecture Notes in Computer Science, Springer, New York 1994, pp. 618–628.

[3] Head T., Weber A.; *Deciding multiset decipherability*, IEEE Transactions on Information Theory 41(1), 1995, pp. 291–297.

[4] Guzmán F.; *Decipherability of codes*, Journal of Pure and Applied Algebra 141(1), 1999, pp. 13–35.

[5] Restivo A.; *A note on multiset decipherable code*, IEEE Transactions on Information Theory 35(3), 1989, pp. 662–663.

[6] Blanchet-Sadri F., Morgan, C.; *Multiset and set decipherable codes*, Computers and Mathematics with Applications 41(10–11), 2001, pp. 1257–1262.

[7] Blanchet-Sadri F.; *On unique, multiset, set decipherability of three-word codes*, IEEE Transactions on Information Theory 47(5), 2001, pp. 1745–1757.

[8] Burderi F., Restivo A.; *Varieties of codes and kraft inequality*, Theory of Computing Systems 40(4), 2007, pp. 507–520.

[9] Burderi F., Restivo A.; *Coding partitions*, Discrete Mathematics and Theoretical Computer Science 9(2), 2007, pp. 227–240.

[10] Salomaa A., Salomaa K., Yu S.; *Variants of codes and indecomposable languages*, Information and Computation 207(11), 2009, pp. 1340–1349.

[11] Aigrain P., Beauquier D.; *Polyomino tilings, cellular automata and codicity*, Theoretical Computer Science 147(1–2), 1995, pp. 165–180.

[12] Giammarresi D., Restivo A.; *Two-dimensional finite state recognizability*, Fundamenta Informaticae 25(3), 1996, pp. 399–422.

[13] Mantaci S., Restivo A.; *Codes and equations on trees*, Theoretical Computer Science 255, 2001, pp. 483–509.

[14] Beauquier D., Nivat M.; *A codicity undecidable problem in the plane*, Theoretical Computer Science 303(2–3), 2003, pp. 417–430.

[15] Moczurad W.; *Brick codes: families, properties, relations*, International Journal of Computer Mathematics 74, 2000, pp. 133–150.

[16] Kolarz M., Moczurad W.; *Directed figure codes are decidable*, Discrete Mathematics and Theoretical Computer Science 11(2), 2009, pp. 1–14.

[17] Costagliola G., Ferrucci F., Gravino C.; *Adding symbolic information to picture models: definitions and properties*, Theoretical Computer Science 337, 2005, pp. 51–104.

[18] Moczurad W.; *Directed figure codes with weak equality*. In: *Intelligent Data Engineering and Automated Learning IDEAL 2010*, vol. 6283 of Lecture Notes in Computer Science, Springer, New York 2010, pp. 242–250.

[19] Kolarz M.; *The code problem for directed figures*, Theoretical Informatics and Applications RAIRO 44(4), 2010, pp. 489–506.

[20] Kolarz M.; *Directed figure codes: Decidability frontier*. In: *COCOON 2010*, vol. 6196 of Lecture Notes in Computer Science, Springer, New York 2010, pp. 530–539.

[21] Kolarz M., Moczurad W.; *Multiset, set and numerically decipherable codes over directed figures*. In: *IWOCA 2012*, vol. 7643 of Lecture Notes in Computer Science, Springer, New York 2012, pp. 224–235.