

SOME CLASSES OF RATIONAL FUNCTIONS FOR PICTURES^{*,**}

FRANTIŠEK MRÁZ¹, FRIEDRICH OTTO² AND DANIEL PRŮŠA³

Abstract. With the aid of homogeneous morphisms, we turn the deterministic two-dimensional two-way ordered restarting automaton and its extended variant into devices that compute transductions of pictures, and we study the resulting classes of transductions in detail.

Mathematics Subject Classification. 68Q45.

1. INTRODUCTION

Digital image processing is a rapidly developing discipline [5]. It has a long term impact on the theory of formal languages, as it motivates researchers to extend notions of grammars and automata from one-dimensional strings to two-dimensional pictures. The aim of many tasks of digital image processing (as *e.g.* filtering, editing or restoration) is to perform a transformation of the input image. While there have been many contributions to the theory of picture languages concerning automata and grammars (see, *e.g.*, [4] for an overview), it seems that transformations on pictures have not yet been studied from the automata-theoretical point of view. Here we present such a study based on the deterministic ordered restarting automaton (or *det-ORWW-automaton*).

The *det-ORWW-automaton* was introduced in [11]. Such an automaton has a finite-state control, a tape with end markers that initially contains the input, and a window of size three. Based on its state and the contents of its window, the automaton can either perform a *move-right step*, which shifts the window one position to the right and changes the state, or a combined *rewrite/restart step*, which replaces the symbol in the middle of the window by a symbol that is strictly smaller with respect to a predefined ordering on the working alphabet, moves the window back to the left end of the tape, and resets the state to the initial state, or an *accept step*, which causes the automaton to halt and accept. While the nondeterministic variant of this type of automaton even accepts some languages that are not growing context-sensitive [8], it has been shown in [11] that the deterministic variant accepts exactly the regular languages. In [13], it was observed that states are actually not needed for *det-ORWW-automata*, as each automaton of this form can be transformed into

Keywords and phrases. Picture language, restarting automaton, transformation, rational function.

* *Some of the results of this paper have been announced at NCMA 2015 in Porto, Portugal, August 2015. An extended abstract appeared in the proceedings of that conference [12].*

** *The first and the third author were supported by the Czech Science Foundation under the project 15-04960S.*

¹ Charles University, Faculty of Mathematics and Physics, Department of Computer Science, Malostranské nám. 25, 118 00 Praha 1, Czech Republic. frantisek.mraz@mff.cuni.cz

² Fachbereich Elektrotechnik/Informatik, Universität Kassel, 34109 Kassel, Germany. otto@theory.informatik.uni-kassel.de

³ Czech Technical University, Faculty of Electrical Engineering, Karlovo nám. 13, 121 35 Prague 2, Czech Republic. prusapa1@fel.cvut.cz

a stateless det-ORWW-automaton that accepts the same language as the given automaton. Taking the size of the working alphabet as a measure of the descriptive complexity of a stateless det-ORWW-automaton, it was shown that stateless det-ORWW-automata are exponentially more succinct than even nondeterministic finite-state acceptors [7]. Finally in [15], these automata were turned into transducers by combining them with morphisms, and it was shown that they characterize the rational functions that map the empty word to itself.

In [10,11,14], we introduced and studied various extensions of the det-ORWW-automaton to two dimensions. All these types of two-dimensional det-ORWW-automata have a window of size 3-by-3 that can move across a given rectangular input picture, starting at the top left corner. A sequence of move operations ends with a combined rewrite/restart step, which replaces the symbol in the middle of the window by a smaller symbol, moves the window back to the top left corner, and resets the state to the initial state, it ends with an accept step, or it ends when no further transition is applicable. The various types of two-dimensional det-ORWW-automata differ in the form of the window movements that are allowed. Here we first consider the most restricted one of these models, the *deterministic two-dimensional two-way ordered restarting automaton* (det-2D-2W-ORWW-automaton, for short), which can only perform move-right and move-down steps. Even though, it is already powerful enough to accept a class of picture languages that strictly extends the class DREC, the class of deterministically recognizable two-dimensional languages [14]. Based on this result and the above observation on the (string) transformations that are computed by det-ORWW-automata in combination with morphisms, we have chosen to extend the det-2D-2W-ORWW-automaton into a two-dimensional transducer by combining it with a homogeneous morphism (see Sect. 3 for the definitions). In this way we obtain a class of two-dimensional transductions that can be interpreted as a class of *rational functions on pictures*. As we will see, the transductions based on det-2D-2W-ORWW-automata include all spacial filters that are used in digital image processing (see, e.g., [5]), and they are closed under composition. However, the operations of horizontal mirroring, vertical mirroring, and rotation cannot be realized by these devices.

Therefore, we also consider the *deterministic two-dimensional extended two-way ordered restarting automaton* (det-2D-x2W-ORWW-automaton, for short) from [10], which is obtained from the det-2D-2W-ORWW-automaton by an extension of the move-right and move-down operations. It is known that already the stateless variant of these automata can simulate the deterministic Sgaffito automaton of [18], which in turn can simulate the det-2D-2W-ORWW-automaton, and that the det-2D-x2W-ORWW-automaton with states is strictly more expressive than its stateless variant [16]. We will see that the resulting class of transductions contains the operations of horizontal and vertical mirroring. However, it is only closed under a restricted form of composition. In addition, it appears that the operation of rotation cannot even be realized by these devices.

This paper is structured as follows. In Section 2, we recall some basic notions on pictures and picture languages, we restate the definition of the det-2D-2W-ORWW-automaton, and we recall that each such automaton can be turned into an equivalent det-2D-2W-ORWW-automaton that has only a single state. Hence, for these automata states are actually not needed, and accordingly, they are called *stateless*. In Section 3, we extend the det-2D-2W-ORWW-automaton into a transducer by combining it with a homogeneous two-dimensional morphism, and we present a detailed example of a transduction computed in this way. Then, in Section 4 we prove the closure properties mentioned before, and we show that the operations of horizontal mirroring, vertical mirroring, and rotation cannot be realized by these devices. In Section 5 we study transformations computed by det-2D-x2W-ORWW-automata. Finally, in Section 6 we observe that each of the recognizable picture languages of [3] occurs as the range of a transduction that is computed by a det-2D-2W-ORWW-automaton.

2. PICTURES AND PICTURE LANGUAGES

We use the common notation and terms on pictures and picture languages (see, e.g., [4]). For a finite alphabet Σ , $\Sigma^{*,*}$ denotes the set of rectangular pictures over Σ , that is, if $P \in \Sigma^{*,*}$, then P is a two-dimensional array of symbols from Σ . We denote the number of rows and columns of a picture P by $\text{rows}(P)$ and $\text{cols}(P)$, respectively. The pair $(\text{rows}(P), \text{cols}(P))$ is called the *dimension* of P . For all $m, n \geq 1$, $\Sigma^{m,n}$ denotes the set

#	⊤	⊤	⋯	⊤	⊤	#
⊢	P					⊣
⋮						⋮
⊢						⊣
#	⊥	⊥	⋯	⊥	⊥	#

FIGURE 1. The boundary picture \widehat{P} .

of pictures of dimension (m, n) over Σ . A *picture language* over Σ is a subset of $\Sigma^{*,*}$. For $1 \leq i \leq \text{rows}(P)$ and $1 \leq j \leq \text{cols}(P)$, $P_{i,j}$ identifies the symbol located in row i and column j of P .

Let $\mathcal{S} = \{\vdash, \dashv, \top, \perp, \#\}$ be a set of five special markers, called *sentinels*. In what follows, we will always assume implicitly that $\Sigma \cap \mathcal{S} = \emptyset$ for any alphabet Σ considered. In order to enable our automata to be defined below to detect the border of a picture $P \in \Sigma^{m,n}$ easily, we define the *boundary picture* \widehat{P} over $\Sigma \cup \mathcal{S}$ of dimension $(m + 2, n + 2)$, which is illustrated in Figure 1. Here the symbols \vdash, \dashv, \top and \perp uniquely identify the corresponding borders (left, right, top, bottom) of \widehat{P} , while the symbol $\#$ marks the corners of \widehat{P} .

In [10, 11], we introduced several variants of a two-dimensional deterministic ordered restarting automaton. These models are restricted variants of the *deterministic two-dimensional four-way ordered restarting automaton*, which has a finite set of states and a scanning window of dimension $(3, 3)$. Based on its current state and the current contents of the window, it can move in any of the four possible directions and change its state, or it can perform a combined rewrite/restart step in which it replaces the symbol in the central position of its window by a symbol that is smaller with respect to a predefined ordering. It accepts by executing a specific accept instruction. As shown in [14], these automata are quite expressive, as they accept at least all deterministic context-free (string) languages. Therefore, several restricted variants have been studied in [10, 11, 14] that are obtained from these four-way automata by restricting the possible window movements. Here we concentrate on the most restricted one of these models, the *deterministic two-dimensional two-way ordered restarting automaton*. Formally this automaton is defined as follows, where $\mathcal{H} = \{R, D\}$ is the set of possible *window movements*.

Definition 2.1. A *deterministic two-dimensional two-way ordered restarting automaton* (a det-2D-2W-ORWW-automaton⁴) is given through a 7-tuple $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$, where

- Q is a finite set of states containing the initial state q_0 ,
- Σ is a finite input alphabet and Γ is a finite tape alphabet containing Σ such that $\Gamma \cap \mathcal{S} = \emptyset$, where \mathcal{S} is the aforementioned set of sentinels,
- $>$ is a partial ordering on Γ , and
- $\delta : Q \times (\Gamma \cup \mathcal{S})^{3,3} \rightarrow (Q \times \mathcal{H}) \cup \Gamma \cup \{\text{Accept}\}$ is the transition function that satisfies the following three restrictions for all $q \in Q$ and all $C \in (\Gamma \cup \mathcal{S})^{3,3}$:
 - (1) if $C_{2,3} = \dashv$, then $\delta(q, C) \neq (q', R)$ for all $q' \in Q$,
 - (2) if $C_{3,2} = \perp$, then $\delta(q, C) \neq (q', D)$ for all $q' \in Q$,
 - (3) if $\delta(q, C) = b \in \Gamma$, then $C_{2,2} > b$ with respect to the ordering $>$.

A det-2D-2W-ORWW-automaton is called *stateless* if $Q = \{q_0\}$, that is, if the initial state is its only state. As such an automaton cannot change its state, the state q_0 is actually irrelevant. Accordingly, such an automaton is called *stateless*, and we suppress the components referring to states from its description. In addition, we use the prefix *stl-* to denote stateless types of automata.

⁴The suffix -WW in the name of this model of automaton is inherited from the original model of the restarting automaton (see [6]), where it expresses the fact that these automata can reWrite (in contrast to versions which can only delete symbols) and can use Working (non-input) symbols in their rewrite operations.

To simplify the presentation we say that the window of \mathcal{M} is *at position* (i, j) to mean that the field in the center of the window is at row i and column j . Given a picture $P \in \Sigma^{m,n}$ as input, \mathcal{M} begins its computation in state q_0 with its read/write window reading the subpicture of dimension $(3, 3)$ of \hat{P} at the upper left corner, that

is, the window is at position $(1, 1)$ of P . Thus, \mathcal{M} sees the subpicture $\begin{pmatrix} \# & \top & \top \\ \vdash & P_{1,1} & P_{1,2} \\ \vdash & P_{2,1} & P_{2,2} \end{pmatrix}$. Applying its transition

function, \mathcal{M} now moves through \hat{P} . If \mathcal{M} is in a state q , sees C in its read/write window and $\delta(p, C) = (q', R)$, then \mathcal{M} moves its read/write window one position to the right and enters state q' . Similarly, if $\delta(p, C) = (q', D)$, the automaton moves its read/write window one position down and enters state q' . The automaton \mathcal{M} continues moving through the picture until it reaches a state q and a position with current contents C of the read/write window such that either $\delta(p, C)$ is undefined, or $\delta(p, C) = \text{Accept}$, or $\delta(p, C) = b$ for some letter $b \in \Gamma$ which is smaller than $C_{2,2}$ by definition. In the first case, \mathcal{M} gets stuck, and so the current computation ends without accepting, in the second case, \mathcal{M} halts and accepts, and in the third case, \mathcal{M} replaces the symbol $C_{2,2}$ by the symbol b , moves its read/write window back to the upper left corner, and reenters its initial state q_0 . This latter step is called a *combined rewrite/restart step*. A part of a computation that begins with an initial configuration (or after a rewrite/restart step) and that ends with the first (next) rewrite/restart step is called a *cycle*, and the part that follows after the last rewrite/restart step is called a *tail computation*. Thus, the computation of \mathcal{M} on input P consists of a sequence of cycles that is followed by a *tail computation*.

A picture $P \in \Sigma^{*,*}$ is *accepted* by \mathcal{M} , if the computation of \mathcal{M} on input P ends with an accept instruction. By $L(\mathcal{M})$ we denote the language consisting of all pictures over Σ that \mathcal{M} accepts, and by $\mathcal{L}(\text{det-2D-2W-ORWW})$ ($\mathcal{L}(\text{stl-det-2D-2W-ORWW})$) we denote the class of picture languages that are accepted by (stateless) det-2D-2W-ORWW-automata. In fact, stateless det-2D-2W-ORWW-automata are as expressive as those with states.

Theorem 2.2. [12] *From a given det-2D-2W-ORWW-automaton \mathcal{M} , one can construct a stateless det-2D-2W-ORWW-automaton \mathcal{M}_0 in polynomial time such that $L(\mathcal{M}_0) = L(\mathcal{M})$.*

Actually, if $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$, then the stateless det-2D-2W-ORWW-automaton \mathcal{M}_0 has a tape alphabet of size $O((|Q| + |\Gamma|) \cdot |\Gamma|^5)$.

By REC we denote the class of *recognizable picture languages* of Giammarresi and Restivo [3], and by DREC we denote the *deterministically recognizable two-dimensional languages* of Anselmo, Giammarresi, and Madonia [1, 2]. Concerning these classes of picture languages, the following inclusion results have been obtained (see, e.g., [14]):

$$\text{DREC} \subsetneq \mathcal{L}(\text{stl-det-2D-2W-ORWW}) = \mathcal{L}(\text{det-2D-2W-ORWW}) \subsetneq \text{P},$$

where P denotes the polynomial-time recognizable languages.

3. COMPUTING TRANSDUCTIONS

While processing a given input picture P , a det-2D-2W-ORWW-automaton \mathcal{M} transforms P into a picture over its tape alphabet. Accordingly, one could simply take this transformation as the *transduction computed by \mathcal{M}* . However, as we will see below in Example 3.2, \mathcal{M} needs to encode some technical information on the current computation within the picture it is processing. This information disturbs the image of P being produced. Therefore, we adopt the definition from [15] to the two-dimensional case, that is, we associate a transduction with a det-2D-2W-ORWW-automaton and a morphism.

Definition 3.1.

- (a) Let Γ and Δ be two finite alphabets. A *homogeneous morphism* from $\Gamma^{*,*}$ into $\Delta^{*,*}$ is defined by two integers $b, h \geq 1$ and a mapping $\varphi : \Gamma \rightarrow \Delta^{h,b}$ that associates with each letter $a \in \Gamma$ a picture $\varphi(a)$ of dimension (h, b) over Δ . Then φ can be extended naturally into a morphism $\varphi : \Gamma^{*,*} \rightarrow \Delta^{*,*}$ that maps a picture $P \in \Gamma^{m,n}$ into a picture $\varphi(P) \in \Delta^{m \cdot h, n \cdot b}$.

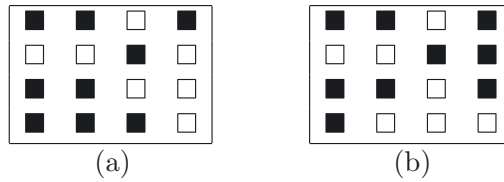


FIGURE 2. An example picture P_0 from L_{sq} (a) and the picture $\tau(P_0)$ (b).

(b) Let $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ be a det-2D-2W-ORWW-automaton, let Δ be a finite (output) alphabet, and let $\varphi : \Gamma^{*,*} \rightarrow \Delta^{*,*}$ be a homogeneous morphism. For $P \in L(\mathcal{M})$, let \tilde{P} denote the final tape inscription that \mathcal{M} produces during its accepting computation on input P . With P we associate the output picture $\varphi(\tilde{P})$. In this way the pair (\mathcal{M}, φ) defines a transduction $\varphi_{\mathcal{M}} : \Sigma^{*,*} \rightarrow \Delta^{*,*}$ with domain $L(\mathcal{M})$.

Next we present an example of a transduction on pictures that is computed by a det-2D-2W-ORWW-automaton.

Example 3.2. Let $\Sigma = \{\square, \blacksquare\}$ and $L_{\text{sq}} = \{P \in \Sigma^{*,*} \mid \text{rows}(P) = \text{cols}(P) \geq 1\}$, that is, L_{sq} consists of all square pictures over Σ . An example picture P_0 is given in Figure 2a.

Next we define a transformation τ on L_{sq} by taking, for $n := \text{rows}(P) = \text{cols}(P)$,

$$\begin{aligned} \tau(P) &:= Q \in \Sigma^{n,n}, \text{ where } Q_{n,i} = P_{i,n} \text{ and } Q_{i,n} = P_{n,i}, 1 \leq i \leq n, \\ &\text{and } Q_{i,j} = P_{i,j}, 1 \leq i, j < n, \end{aligned}$$

that is, $\tau(P)$ is obtained from P by interchanging the last column with the last row, leaving all other entries untouched. See Figure 2b for the picture $\tau(P_0)$.

We present a det-2D-2W-ORWW-automaton $\mathcal{M}_\tau = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ and a homogeneous morphism $\varphi : \Gamma^{*,*} \rightarrow \Sigma^{*,*}$ such that $(\mathcal{M}_\tau, \varphi)$ realizes the transformation $\varphi_{\mathcal{M}_\tau} = \tau$.

We take $\Gamma = \Sigma \cup \{[a, b]_i \mid a, b \in \Sigma, 1 \leq i \leq 5\}$, we define a partial ordering $>$ by taking $a > [a, b]_1 > [a, c]_2 > [a, d]_3 > [a, e]_4 > [a, f]_5$ for all $a, b, c, d, e, f \in \Sigma$, and we define the transition function δ in such a way that \mathcal{M}_τ proceeds as follows given a picture $P \in \Sigma^{n,n}$ as input:

- (1) The information on the last column is moved to the main diagonal and then down to the bottom row. This is achieved as follows. First, the symbols $P_{1,n-1}, P_{1,n-2}, \dots, P_{1,2}$ are rewritten, from right to left, by the symbols $[P_{1,n-1}, P_{1,n}]_1, [P_{1,n-2}, P_{1,n}]_1, \dots, [P_{1,2}, P_{1,n}]_1$, and then the symbol $P_{1,1}$ is rewritten by the symbol $[P_{1,1}, P_{1,n}]_2$. Then in row 2, the information on the last symbol $P_{2,n}$ is copied from right to left, replacing the symbols $P_{2,j}$ by $[P_{2,j}, P_{2,n}]_1$ for $j = n - 1, n - 2, \dots, 3$ and by replacing the symbol $P_{2,2}$ by the symbol $[P_{2,2}, P_{2,n}]_2$. Observe that \mathcal{M}_τ recognizes position $(2, 2)$ by observing that the top row of its window contains an auxiliary symbol with index 2 that is followed by two auxiliary symbols with index 1. Finally, the symbol $P_{2,1}$ is replaced by the symbol $[P_{2,1}, P_{1,n}]_2$ by copying the information on $P_{1,n}$ from the auxiliary symbol at position $(1, 1)$. Now rows 3 to $n - 1$ are dealt with in the same way, that is, the symbol $P_{i,n}$ is copied to the left, letter by letter, using auxiliary symbols with index 1 until it finally reaches the main diagonal, where it is encoded by the symbol $[P_{i,i}, P_{i,n}]_2$, and then the information on $P_{j,n}$, $j = i - 1, i - 2, \dots, 1$, is copied from the first $i - 1$ symbols in row $i - 1$ to the first $i - 1$ symbols in row i . This continues until $P_{n-1,1}$ is rewritten by $[P_{n-1,1}, P_{1,n}]_2$. Finally, the entries in row n are rewritten from right to left, replacing $P_{n,i}$ by $[P_{n,i}, P_{i,n}]_2$, $1 \leq i \leq n$. When entry $P_{n,1}$ has been rewritten, then \mathcal{M}_τ realizes after the next restart that phase 1 is complete. In Figure 3a the picture resulting from P_0 by phase 1 is presented.
- (2) Now the information on the bottom row is moved up to the main diagonal, and then it is moved to the last column. This is achieved as follows. First, the information on the symbol $P_{n,1}$ is moved up the first column, replacing the symbols $[P_{i,1}, P_{1,n}]_2$ by $[P_{i,1}, P_{n,1}]_3$, $2 \leq i \leq n - 1$, and finally replacing the symbol $[P_{1,1}, P_{1,n}]_2$ by $[P_{1,1}, P_{n,1}]_4$. Next the information on the symbol $P_{n,2}$ is moved up the second column,

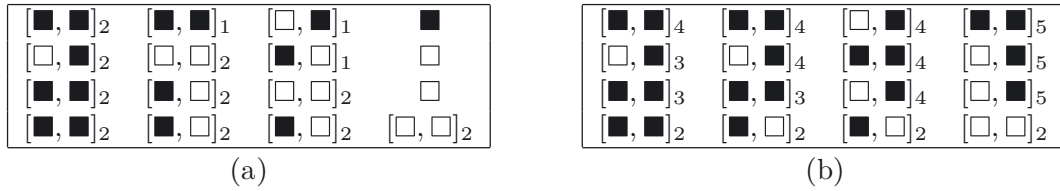


FIGURE 3. The example picture P_0 after phase 1 (a) and after phase 2 (b).

replacing the symbols $[P_{i,2}, P_{2,n}]_2$ by $[P_{i,2}, P_{n,2}]_3$, $3 \leq i \leq n - 1$, by replacing the symbol $[P_{2,2}, P_{2,n}]_2$ by $[P_{2,2}, P_{n,2}]_4$, and finally replacing the symbol $[P_{1,2}, P_{1,n}]_1$ by $[P_{1,2}, P_{n,1}]_4$. Then columns 3 to $n - 1$ are dealt with analogously, and finally, the symbol $P_{i,n}$ is rewritten into $[P_{i,n}, P_{n,i}]_5$, $i = n - 1, n - 2, \dots, 1$. Figure 3b shows the example picture P_0 after phase 2.

- (3) Finally, the homogeneous morphism φ maps the symbols of the form $[a, b]_3$ or $[a, b]_4$ to a , as these are the symbols that remain in those positions that do not belong to the last row or the last column, and those of the form $[a, b]_2$ and $[a, b]_5$ to b , as these are the symbols in the last row and the last column, respectively.

As the window of \mathcal{M}_τ is of dimension $(3, 3)$, it is easily seen that δ can be defined in the required way. It follows that $(\mathcal{M}_\tau, \varphi)$ correctly computes the transformation τ .

By a slight modification of the above construction it is possible to construct a det-2D-2W-ORWW-automaton and a projection that realize a transduction which swaps some other pair of the first/last row or the first/last column of input pictures. However, we will see in Corollary 4.10 that transductions which rotate the whole picture cannot be realized in this way.

Using the same construction as in the proof of Theorem 2.2 (see [12]), and by extending the morphism φ accordingly, we obtain the following consequence.

Corollary 3.3. *Let (\mathcal{M}, φ) be a det-2D-2W-ORWW-automaton and a homogeneous morphism that realize a transduction $\tau : \Sigma^{*,*} \rightarrow \Delta^{*,*}$. Then one can construct a stateless det-2D-2W-ORWW-automaton \mathcal{M}_0 and a homogeneous morphism φ_0 in polynomial time such that the pair $(\mathcal{M}_0, \varphi_0)$ also realizes the transduction τ .*

By $\mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$ ($\mathcal{F}_{hm}(\text{stl-det-2D-2W-ORWW})$) we denote the class of transductions that are realized by (stateless) det-2D-2W-ORWW-automata with the help of homogeneous morphisms. From Corollary 3.3 we see that these two classes of transductions coincide. In the one-dimensional case, that is, for strings, the class of functions that are computed by det-ORWW-automata with the aid of morphisms coincides with the class RatF_0 of rational functions that map the empty word to itself [15]. Therefore, we feel that it is justified to consider the class of transductions $\mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$ as a reasonable candidate for the class of rational functions on pictures. To justify this choice we now study this class of transductions in some detail, concentrating on the closure with respect to composition.

4. ON THE CLASS $\mathcal{F}_{hm}(\text{DET-2D-2W-ORWW})$

First we argue that the transductions computed by det-2D-2W-ORWW-automata include all spatial filters used in digital image processing. Spatial filtering of a digital image consists in computing a value for each pixel of an image as a function of pixel values within a window centered at the location of the pixel (see [5]). In general, a filter based on a window of dimension $((2a + 1), (2b + 1))$ is a function $f : (\Sigma \cup \{\#\})^{2a+1, 2b+1} \rightarrow \Delta$. By applying the filter f to a picture P of dimension (m, n) , we obtain the picture P' of the same dimension as P : $P'_{i,j} = f(B^{i,j,a,b})$, where $B^{i,j,a,b}$ is the subpicture of P of dimension $((2a + 1), (2b + 1))$ with its central field on the symbol $P_{i,j}$. Formally, $B^{i,j,a,b}_{k,l} = P_{i-a+k, j-b+l}$ for $k = 0, 1, \dots, 2a$ and $l = 0, 1, \dots, 2b$, where $P_{i-a+k, j-b+l} = \#$,

whenever $i - a + k \leq 0$, $i - a + k > m$, $j - b + l \leq 0$, or $j - b + l > n$. We denote this as $P' = P * f$. Filters for which the function f is a linear combination of some values associated to symbols from $\Sigma \cup \{\#\}$ are known also as *convolution filters*. They are used in image pre-processing for blurring, contrast enhancing, *etc.*

If $f : (\Sigma \cup \{\#\})^{2a+1, 2b+1} \rightarrow \Delta$ is a filter function with $a = b = 1$, then we can easily construct a det-2D-2W-ORWW-automaton \mathcal{M} and a homogeneous morphism φ such that, for each picture P , $\varphi_{\mathcal{M}}(P) = P * f$. Let $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ be the following det-2D-2W-ORWW-automaton, where $\Gamma = \Sigma \cup \Gamma'$, $\Gamma' = (\Sigma \cup \{\#\})^{3,3}$, and let $\varphi(B) = f(B)$ for all $B \in \Gamma'$. The automaton \mathcal{M} works on an input picture P as follows. From a restarting configuration the automaton moves down the first column of the picture until it finds the last row containing a symbol from Σ . This could also be the last row of the picture. Then it starts to move to the right on this row until it finds the last symbol from Σ (which is followed by the right sentinel \dashv or a symbol from Γ'). Then \mathcal{M} rewrites this symbol into the symbol from Γ' that corresponds to the contents of its scanning window. In such a rewrite step, all sentinels are represented by the symbol $\#$. If no symbol can be rewritten anymore, that is, the symbol at position $(1, 1)$ has already been rewritten into a symbol from Γ' , then \mathcal{M} accepts. Trivially, the morphism φ applied to the final picture over Γ' computes the filtered picture $P * f$.

Now it is easy to generalize the above construction to functions $f : (\Sigma \cup \{\#\})^{2a+1, 2b+1} \rightarrow \Delta$ for any values $a, b \geq 1$. One simply chooses $\Gamma' = (\Sigma \cup \{\#\})^{2a+1, 2b+1}$ and uses repeated rewrites to replace each symbol $P_{i,j} \in \Sigma$ of an input picture P by the symbol $B \in \Gamma'$ that corresponds to the subpicture of dimension $(2a + 1, 2b + 1)$ of P which is centered at the position (i, j) . Together with the homogeneous morphism φ that maps the letter B to $f(B)$ we obtain a realization of the transduction defined by the filter function f through a pair (\mathcal{M}, φ) .

Next we show that the class of transductions $\mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$ is closed under the operation of composition. First, we consider the special case that the first transduction in a composition uses a projection instead of a more general homogeneous morphism.

Lemma 4.1. *Let $\mathcal{M}_1 = (Q_1, \Sigma_1, \Gamma_1, \mathcal{S}, q_1, \delta_1, >_1)$ be a det-2D-2W-ORWW-automaton, let $\pi_1 : \Gamma_1 \rightarrow \Sigma_2$ be a projection, and let $\tau_1 : \Sigma_1^{*,*} \rightarrow \Sigma_2^{*,*}$ be the transduction that is realized by (\mathcal{M}_1, π_1) . Further, let $\mathcal{M}_2 = (Q_2, \Sigma_2, \Gamma_2, \mathcal{S}, q_2, \delta_2, >_2)$ be a det-2D-2W-ORWW-automaton, let $\varphi_2 : \Gamma_2^{*,*} \rightarrow \Delta^{*,*}$ be a homogeneous morphism, and let $\tau_2 : \Sigma_2^{*,*} \rightarrow \Delta^{*,*}$ denote the transduction that is realized by $(\mathcal{M}_2, \varphi_2)$. Then there exist a det-2D-2W-ORWW-automaton \mathcal{M} and a homogeneous morphism φ such that (\mathcal{M}, φ) realizes the transduction $\tau = \tau_2 \circ \tau_1$.*

Proof. The det-2D-2W-ORWW-automaton \mathcal{M} will first simulate \mathcal{M}_1 , then it will determine the corresponding output picture, and finally it will simulate \mathcal{M}_2 on the latter. The main issue is how to distinguish between these stages. This must be achieved through the contents of the various cells on the tape.

We take $\mathcal{M} = (Q_1 \cup Q_2, \Sigma_1, \Gamma_1 \cup \Gamma_2, \mathcal{S}, q_1, \delta, >)$, where we assume that $Q_1 \cap Q_2 = \emptyset$ and $\Gamma_1 \cap \Gamma_2 = \emptyset$, we define the ordering $>$ through

$$a > b \text{ if } (a, b \in \Gamma_1 \text{ and } a >_1 b) \text{ or } (a, b \in \Gamma_2 \text{ and } a >_2 b) \text{ or } (a \in \Gamma_1 \text{ and } b \in \Gamma_2),$$

and we define δ in such a way that it realizes the following three phases:

- (1) \mathcal{M} simulates \mathcal{M}_1 on the input picture until \mathcal{M}_1 reaches a configuration in which \mathcal{M}_1 would apply an accepting instruction $(q, C) \rightarrow \text{Accept}$ ($q \in Q_1$, $C \in \Gamma_1^{3,3}$). Instead of this accept instruction, \mathcal{M} performs the rewrite/restart step $(q, C) \rightarrow \pi_1(C_{2,2})$.
- (2) Let \mathcal{P} be the time-ordered sequence of cells which \mathcal{M}_1 visits from its last restart to the accept step. \mathcal{M} still simulates the steps of \mathcal{M}_1 , however, a cycle is ended as soon as the window contains at least one symbol from Γ_2 . In this case, assuming that the central cell of the scanning window stores the symbol $a \in \Gamma_1$, \mathcal{M} rewrites this symbol into $\pi_1(a)$ and restarts. Since \mathcal{M}_1 is deterministic, \mathcal{M} will traverse shorter and shorter prefixes of \mathcal{P} until it scans a symbol from Γ_2 immediately after a restart. An example of \mathcal{P} and all its cells that are rewritten into symbols from Γ_2 during the first and second phase is depicted in Figure 4.
- (3) The presence of a symbol from Γ_2 in the window immediately after a restart informs \mathcal{M} that it should now simulate \mathcal{M}_2 . Note that at this stage, the cells contain symbols from both, Γ_1 and Γ_2 . During this phase, however, each symbol $a \in \Gamma_1$ is simply interpreted as the symbol $\pi_1(a)$. \square

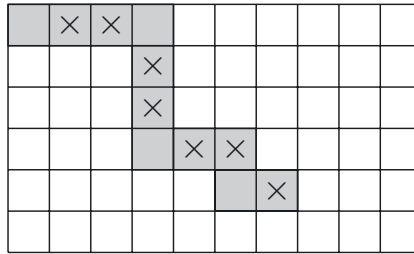


FIGURE 4. The gray cells form the path traversed by \mathcal{M}_1 from its last restart to its accept step. The crossed cells are rewritten to a symbol from Γ_2 by \mathcal{M} in the first and second phase.

Lemma 4.2. *The class of picture languages $\mathcal{L}(\text{det-2D-2W-ORWW})$ is closed under inverse homogeneous morphisms, that is, if $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ is a det-2D-2W-ORWW-automaton and $\varphi : \Delta^{*,*} \rightarrow \Sigma^{*,*}$ is a homogeneous morphism, then there exists a det-2D-2W-ORWW-automaton \mathcal{M}' which accepts $P \in \Delta^{*,*}$ if and only if \mathcal{M} accepts $\varphi(P)$.*

Proof. Let (h, b) be the dimension of the images $\varphi(a)$, where $a \in \Delta$, that is, each picture $P \in \Delta^{m,n}$ is mapped by φ to a picture $\varphi(P) \in \Sigma^{m \cdot h, n \cdot b}$. The idea is to construct a det-2D-2W-ORWW-automaton $\mathcal{M}' = (Q', \Delta, \Gamma', \mathcal{S}, q'_0, \delta', >')$ which simulates the computation of \mathcal{M} on input $\varphi(P)$, where each tape cell of \mathcal{M}' represents a block of dimension (h, b) of the tape of \mathcal{M} . We thus define $\Gamma' = \Delta \cup \Gamma^{h,b}$. Each $B \in \Gamma^{h,b}$ represents the possible contents of a block, where $B_{i,j}$ is the symbol in the i th row and the j th column of that block. Moreover, each input symbol $a \in \Delta$ represents the block contents $\varphi(a)$. In one step, \mathcal{M}' simulates all those steps that are performed by \mathcal{M} within the corresponding block, ending when \mathcal{M} leaves the block or when \mathcal{M} ends the current cycle inside this block. When \mathcal{M} leaves a block, the automaton \mathcal{M}' must perform a corresponding move right or move down step. In order to simulate the next step of \mathcal{M} inside the corresponding block, the state of \mathcal{M}' must record the position at the block perimeter in which \mathcal{M} enters it. Since a det-2D-2W-ORWW-automaton moves its head only to the right and down, there are $h + b - 1$ positions in the first row and the first column of each block at which \mathcal{M} can enter it. Accordingly, we take $Q' = Q \times \{1, 2, \dots, h + b - 1\}$ and $q'_0 = (q_0, 1)$, assuming that the number 1 represents the top-left corner of a block. The partial ordering $>'$ is defined in accordance with $>$, taking $B >' B'$ if $B' \in \Gamma^{h,b}$, $B \in \Gamma^{h,b}$ or $B = \varphi(a)$ for some $a \in \Delta$, and the following condition is met: $\exists i, j : B_{i,j} > B'_{i,j} \wedge \nexists i, j : B'_{i,j} > B_{i,j}$. Finally, δ' is chosen to realize the instructions that implement the simulation of \mathcal{M} . □

By combining Lemmas 4.1 and 4.2 we obtain the intended closure property.

Theorem 4.3. *The class of transductions $\mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$ is closed under composition.*

Proof. Let $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ and $\mathcal{M}' = (Q', \Sigma', \Gamma', \mathcal{S}, q'_0, \delta', >')$ be two det-2D-2W-ORWW-automata, $\varphi : \Gamma^{*,*} \rightarrow (\Sigma')^{*,*}$ and $\varphi' : (\Gamma')^{*,*} \rightarrow \Delta^{*,*}$ be two homogeneous morphisms, τ and τ' be the transductions realized by (\mathcal{M}, φ) and (\mathcal{M}', φ') , respectively. Using the construction from the proof of Lemma 4.2, we can transform the automaton \mathcal{M}' into a det-2D-2W-ORWW-automaton $\overline{\mathcal{M}}$ which accepts $P \in \Gamma^{*,*}$ if and only if \mathcal{M}' accepts $\varphi(P)$. Moreover, any final tape inscription \overline{P} that $\overline{\mathcal{M}}$ produces during its accepting computation contains the corresponding final tape contents \tilde{P} obtained by \mathcal{M}' on the input $\varphi(P)$ in encoded form. We can easily define a homogeneous morphism $\overline{\varphi'}$ which computes $\varphi'(\tilde{P})$ from \overline{P} . A det-2D-2W-ORWW-automaton realizing $\tau' \circ \tau$ can be obtained from \mathcal{M} with the identity morphism and $\overline{\mathcal{M}}$ with $\overline{\varphi'}$ using Lemma 4.1. □

For pictures $P \in \Sigma_1^{*,*}$ and $R \in \Sigma_2^{*,*}$ of the same dimension (m, n) , let $P \otimes R$ denote their *pointwise composition*, which is the picture $S \in (\Sigma_1 \times \Sigma_2)^{m,n}$ such that $S_{i,j} = (P_{i,j}, R_{i,j})$ for all i, j . Now let $\tau_1 : \Sigma_1^{*,*} \rightarrow \Gamma_1^{*,*}$

and $\tau_2 : \Sigma_2^{*,*} \rightarrow \Gamma_2^{*,*}$ be two transductions such that, for all $P \in \Sigma_1^{*,*}$ and $R \in \Sigma_2^{*,*}$, $\tau_1(P)$ and $\tau_2(R)$ have the same dimension, whenever P and R have the same dimension. Then together with the pointwise composition $P \otimes R$ of P and R , also the pointwise composition $\tau_1(P) \otimes \tau_2(R)$ of $\tau_1(P)$ and $\tau_2(R)$ is defined. We define the *parallel composition* of τ_1 and τ_2 as the transduction $\tau_1 \otimes \tau_2 : (\Sigma_1 \times \Sigma_2)^{*,*} \rightarrow (\Gamma_1 \times \Gamma_2)^{*,*}$ given through $(\tau_1 \otimes \tau_2)(P \otimes R) = \tau_1(P) \otimes \tau_2(R)$.

Corollary 4.4. *Let $\tau_1 : \Sigma_1^{*,*} \rightarrow \Gamma_1^{*,*}$ and $\tau_2 : \Sigma_2^{*,*} \rightarrow \Gamma_2^{*,*}$ be transductions that are realizable by det-2D-2W-ORWW-automata and homogeneous morphisms. If the parallel composition $\tau_1 \otimes \tau_2$ of τ_1 and τ_2 is defined, then it is also realizable by a det-2D-2W-ORWW-automaton and a homogeneous morphism.*

Proof. In analogy to the proof of Lemma 4.1, a det-2D-2W-ORWW-automaton \mathcal{M} with input alphabet $\Sigma_1 \times \Sigma_2$ can first simulate the automaton \mathcal{M}_1 , where $(\mathcal{M}_1, \varphi_1)$ realizes the transduction τ_1 , using the first component of each cell and keeping the information in the second component of each cell fixed. Thereafter, it simulates the automaton \mathcal{M}_2 , where $(\mathcal{M}_2, \varphi_2)$ realizes the transduction τ_2 , using the second component of each cell and keeping the information in the first component of each cell fixed. By combining φ_1 and φ_2 componentwise, we obtain a homogeneous morphism φ such that the pair (\mathcal{M}, φ) realizes the transduction $\tau_1 \otimes \tau_2$. \square

Let $\tau_1 : \Sigma^{*,*} \rightarrow \Gamma_1^{*,*}$ and $\tau_2 : \Sigma^{*,*} \rightarrow \Gamma_2^{*,*}$ be transductions such that $\tau_1(P)$ and $\tau_2(P)$ have the same dimension for all pictures P . Then we can define their parallel composition $\tau : \Sigma^{*,*} \rightarrow (\Gamma_1 \times \Gamma_2)^{*,*}$ by taking $\tau(P) = \tau_1(P) \otimes \tau_2(P)$. Also this transformation is realizable, if τ_1 and τ_2 are realizable, as we can easily realize the mapping $a \rightarrow (a, a)$.

The following lemma can be used to prove that a certain transduction cannot be realized by a det-2D-2W-ORWW-automaton and a homogeneous morphism.

Lemma 4.5. *Let $\tau : \Sigma^{*,*} \rightarrow \Gamma^{*,*}$ be a transduction such that the dimensions of P and of $\tau(P)$ coincide for every $P \in \Sigma^{*,*}$, and let $\pi_1, \pi_2 : \Sigma \times \Gamma \rightarrow \Sigma$ be the projections that are defined by $\pi_1((a, c)) = a$ and $\pi_2((a, c)) = c$ for all $(a, c) \in \Sigma \times \Gamma$. If there is a det-2D-2W-ORWW-automaton $\mathcal{M} = (Q, \Sigma \times \Gamma, \Gamma', \mathcal{S}, q_0, \delta, >)$ such that the language*

$$L = \{ \pi_1(P') \mid P' \in L(\mathcal{M}) \wedge \pi_2(P') = \tau(\pi_1(P')) \}$$

is not accepted by any det-2D-2W-ORWW-automaton, then the transduction τ does not belong to the class $\mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$.

Proof. Assume that τ is realized by a det-2D-2W-ORWW-automaton \mathcal{M}_1 and a homogeneous morphism φ . From \mathcal{M}_1 , we can construct a det-2D-2W-ORWW-automaton \mathcal{M}_2 that accepts L as follows. For an input $P \in \Sigma^{*,*}$, \mathcal{M}_2 simulates \mathcal{M}_1 while representing the pair $(P, \tau(P))$ on its tape, that is, in each cell it always encodes the current contents together with the original input, and in a final phase each final letter is replaced by its corresponding image under φ . Then it simulates the computation of \mathcal{M} on this pair and accepts P if and only if \mathcal{M} accepts the pair $(P, \tau(P))$. As this contradicts our assumption on L , we see that $\tau \notin \mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$. \square

Based on this lemma, we will prove below that $\mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$ does neither contain the operation of vertical mirroring nor the operation of rotation. However, before we can do that, we need to introduce the following technical result, where Φ denotes the *column concatenation* of pictures. For any two pictures $P \in \Sigma^{m,n}$ and $Q \in \Sigma^{k,l}$, $P \Phi Q$ is defined only if $m = k$, and in this case $P \Phi Q \in \Sigma^{m,n+l}$ is the picture that is obtained by concatenating the corresponding rows of both pictures.

Theorem 4.6. *Let Σ be an alphabet of size $k > 1$ and $f : \Sigma^{*,*} \rightarrow \Sigma^{*,*}$ be an injective function such that, for any picture $P \in \Sigma^{*,*}$, $f(P)$ has the same number of rows as P . Further, let L be a picture language such that $|L \cap \Sigma^{n,*}| \in 2^{\omega(n \log n)}$. Then the language $L_f = \{ P \Phi f(P) \mid P \in L \} \notin \mathcal{L}(\text{det-2D-2W-ORWW})$.*

Proof. Assume to the contrary that there exists a det-2D-2W-ORWW-automaton $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ such that $L(\mathcal{M}) = L_f$. Without loss of generality we may assume that \mathcal{M} is stateless (Thm. 2.2) and that it accepts only at the lower right corner of the picture. Let n be a sufficiently large positive integer the exact value of which will be determined later. We consider the accepting computations on input pictures of the form $P\Phi f(P) \in L_f$ for $P \in \Sigma^{n,*}$. The left part of the picture containing P will be denoted by P_ℓ , and the right part containing $f(P)$ will be denoted by P_r .

As each cycle of \mathcal{M} starts with the read/write window at the upper left corner of P_ℓ , we see that each cycle of \mathcal{M} consists of a part during which \mathcal{M} is on P_ℓ , which is then possibly followed by a part during which \mathcal{M} is on P_r . We are particularly interested in the behaviour of \mathcal{M} at the border between P_ℓ and P_r , that is, when the middle column of its read/write window is over the last column of P_ℓ . These positions are the only positions in which the contents of the tape fields containing P_r has any influence on the behaviour of \mathcal{M} on P_ℓ . In fact, only the contents of the tape fields within the last column of P_ℓ can have any influence on the behavior of \mathcal{M} on P_r , and only the contents of the first column of P_r can have any influence on the behaviour of \mathcal{M} on P_ℓ .

The computation of \mathcal{M} on input $P\Phi f(P)$ can be divided into two types of phases:

- a *left-phase* consists of a sequence of maximum length of cycles during which \mathcal{M} executes rewrite steps on P_ℓ ;
- a *right-phase* consists of a sequence of cycles during which \mathcal{M} executes rewrite steps on P_r until \mathcal{M} accepts or rewrites a symbol in the leftmost column of P_r .

Observe that during a left-phase, the active (that is, the central) position of the window stays on P_ℓ . Further, during all cycles of a right-phase, \mathcal{M} follows exactly the same path through the subpicture P_ℓ , as \mathcal{M} is deterministic, and as the effects of the rewrite operations performed during this right-phase (apart from the last one, which is executed on the first column of P_r) are invisible for \mathcal{M} as long as it is on P_ℓ .

Obviously, an accepting computation of \mathcal{M} on an input of the form $P\Phi f(P)$ cannot consist of a single left-phase only. The first column of P_r has n entries only that can be rewritten at most $|\Gamma| - 1$ times. Thus, altogether at most $n \cdot (|\Gamma| - 1)$ rewrites can occur in this column, which implies that there are at most $n \cdot (|\Gamma| - 1) + 1$ many right-phases and at most as many left-phases.

With the accepting computation of \mathcal{M} on an input $P\Phi f(P)$ of the form described above, we now associate a *generalized crossing sequence* $\text{GCS}(P\Phi f(P))$ that is defined as follows:

- (1) We begin with a pair (C_ℓ, C_r) , where C_ℓ and C_r are the contents of the last column of P_ℓ and the first column of P_r , respectively.
- (2) Whenever a rewrite operation is performed on the m th row within the last column of P_ℓ , we append the corresponding pair of the form $(m, b)_\ell$ to the sequence, where b is the symbol which is written into the m th row of the last column of P_ℓ .
- (3) Whenever a right-phase starts, we append (m) to the sequence, where m is the row in which the right part P_r is entered.
- (4) Whenever a right-phase ends with an execution of a rewrite operation on the m th row within the first column of P_r , we append the corresponding pair of the form $(m, b)_r$ to the sequence, where b is the symbol which is written into the m th row of the first column of P_r .

As there are at most $n \cdot (|\Gamma| - 1) + 1$ many right-phases, at most $n \cdot (|\Gamma| - 1)$ many rewrites in column n , and at most $n \cdot (|\Gamma| - 1)$ many rewrites in column $n + 1$, we see that each sequence $\text{GCS}(\mathcal{M}, P\Phi f(P))$ is of length at most $3 \cdot n \cdot (|\Gamma| - 1) + 2 \leq 3 \cdot n \cdot |\Gamma|$. There are at most $|\Sigma|^{2n}$ possible pairs of columns of length n at the beginning of a generalized crossing sequence. Hence, the number of possible generalized crossing sequences is at most $|\Sigma|^{2n} \cdot (\sum_{i=1}^{3 \cdot n \cdot |\Gamma|} (n + 2 \cdot |\Gamma| \cdot n)^i) \in 2^{O(n \cdot \log n)}$. However, there are $2^{\omega(n \cdot \log n)}$ many pictures of the form $P\Phi f(P)$. If n is sufficiently large, then there are more pictures of this form than there are possible generalized crossing sequences. Hence, there exist two pictures $P_1, P_2 \in \Sigma^{n,n}$ such that $P_1 \neq P_2$, but $\text{GCS}(\mathcal{M}, P_1\Phi f(P_1)) = \text{GCS}(\mathcal{M}, P_2\Phi f(P_2))$. Now it is easy to see that \mathcal{M} will also accept the picture $P_1\Phi f(P_2) \notin L$ – a contradiction. \square

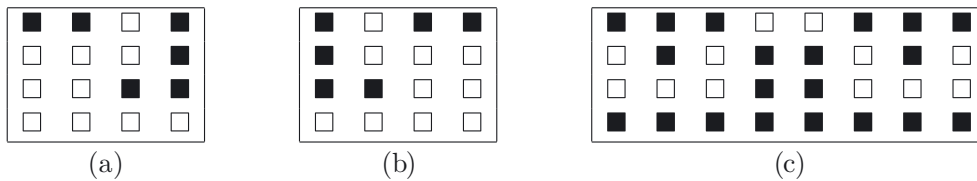


FIGURE 5. An example of a picture P (a), its vertical mirror P^{VM} (b) and a picture from L_{pal} (c).

Now we can prove the following negative result, where Σ is any finite alphabet of cardinality at least 2 and $\tau^{\text{VM}} : \Sigma^{*,*} \rightarrow \Sigma^{*,*}$ denotes the operation of *vertical mirroring*, that is, τ^{VM} maps a picture $P \in \Sigma^{m,n}$, $m, n \geq 2$, to its vertical mirror $P^{\text{VM}} \in \Sigma^{m,n}$, that is, for all i, j , $P_{i,j}^{\text{VM}} = P_{i,n+1-j}$.

Lemma 4.7. $\tau^{\text{VM}} \notin \mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$.

Proof. For simplicity we take $\Sigma = \{\square, \blacksquare\}$. An example of a picture P and its image $\tau^{\text{VM}}(P)$ are given in Figures 5a and 5b. Further, let $L_{\text{pal}} = \{P\Phi P^{\text{VM}} \mid P \in \Sigma^{m,n}, m, n \geq 2\}$, that is, L_{pal} contains exactly those pictures $Q \in \Sigma^{*,*}$ with $m \geq 2$ rows and $2n$ ($n \geq 2$) columns for which $Q = Q^{\text{VM}}$ (that is, each row of Q is a palindrome). A picture from L_{pal} is depicted in Figure 5c. As there are $2^{m \cdot n}$ many pictures of dimension (m, n) over Σ , Theorem 4.6 implies that L_{pal} is not accepted by any det-2D-2W-ORWW-automaton.

On the other hand, if both Q and $\tau^{\text{VM}}(Q)$ are represented in one picture over $\Sigma \times \Sigma$, then a det-2D-2W-ORWW-automaton can easily verify whether Q is in L_{pal} or not by checking whether $Q_{x,y} = (\tau^{\text{VM}}(Q))_{x,y}$ for all coordinates (x, y) . Thus, Lemma 4.5 applies here to show that $\tau^{\text{VM}} \notin \mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$. \square

Hence, the operation of *vertical mirroring* cannot be realized by a det-2D-2W-ORWW-automaton. By symmetry it follows that the operation of *horizontal mirroring* cannot be realized in this way, either. We will show that also the operation of *rotating* a picture by 90 degrees cannot be realized by a det-2D-2W-ORWW-automaton.

Let P^{rot} denote the picture that is obtained from a picture $P \in \Sigma^{m,n}$ by rotating it clockwise by 90 degrees, that is, for all i, j , $P_{i,j}^{\text{rot}} = P_{j,m+1-i}$.

Let $\Sigma = \{\square, \blacksquare\}$, and let L_{rot} denote the following picture language over Σ :

$$L_{\text{rot}} = \{P\Phi(P^{\text{rot}})^{\text{rot}} \mid P \in \Sigma^{n,n}, n \geq 1\},$$

that is, a picture in L_{rot} consists of the column concatenation of a square picture P over Σ and its image $(P^{\text{rot}})^{\text{rot}}$ obtained by a two-fold application of the operation of rotation.

Corollary 4.8. $L_{\text{rot}} \notin \mathcal{L}(\text{det-2D-2W-ORWW})$.

Proof. All pictures in the language L_{rot} are of the form $P\Phi f(P)$, where $f(P) = (P^{\text{rot}})^{\text{rot}}$. The function f is injective and $|L_{\text{rot}} \cap \Sigma^{n,*}| = 2^{\theta(n^2)} \subset 2^{\omega(n \log n)}$. Hence according to Theorem 4.6, the language L_{rot} cannot be accepted by any det-2D-2W-ORWW-automaton. \square

Based on this result, we can now derive the following negative result.

Lemma 4.9. Let $\Sigma = \{\square, \blacksquare\}$, and let $\tau^{\text{rr}} : \Sigma^{*,*} \rightarrow \Sigma^{*,*}$ be the transduction that maps a picture $P \in \Sigma^{m,n}$ to its two-fold rotation $(P^{\text{rot}})^{\text{rot}} \in \Sigma^{m,n}$, that is, for all i, j , $P_{i,j}^{\text{rr}} = P_{m+1-i, n+1-j}$. Then $\tau^{\text{rr}} \notin \mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$.

Proof. Observe that $\tau^{\text{rr}}(P\Phi(P^{\text{rot}})^{\text{rot}}) = P\Phi(P^{\text{rot}})^{\text{rot}}$ and that the language L_{rot} is not accepted by any det-2D-2W-ORWW-automaton. On the other hand, if both P and $\tau^{\text{rr}}(P)$ are represented in one picture over $\Sigma \times \Sigma$, then a det-2D-2W-ORWW-automaton can easily verify whether P is in L_{rot} or not by checking whether $P_{x,y} = (\tau^{\text{rr}}(P))_{x,y}$ for all coordinates (x, y) . Thus, by Lemma 4.5, $\tau^{\text{rr}} \notin \mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$. \square

As by Theorem 4.3 the class of transformations $\mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$ is closed under composition, we obtain the following consequence from Lemma 4.9.

Corollary 4.10. $\tau^{\text{rot}} \notin \mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$, where τ^{rot} is the operation of rotation.

5. COMPUTING TRANSDUCTIONS BY DET-2D-x2W-ORWW-AUTOMATA

A det-2D-2W-ORWW-automaton cannot scan a given picture completely in a single cycle, which leads to rather complicated algorithms, as witnessed by Example 3.2. Therefore, we now turn to an extension of this model in which the move-right and move-down steps are slightly more general.

Definition 5.1. [10] A *deterministic two-dimensional extended two-way ordered restarting automaton*, a det-2D-x2W-ORWW-automaton for short, is given through a 7-tuple $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$, where all components are defined as for a det-2D-2W-ORWW-automaton, but the move-right and move-down steps are extended as follows:

- (1) When the window contains the right border marker \dashv , but not the bottom marker, then an *extended move-right step* shifts the window to the beginning of the next row, that is, if the central position of the window is on the last field of row i for some $i < \text{rows}(P)$, then it is moved to the first field of row $i + 1$.
- (2) When the window contains the bottom marker \perp , but not the right border marker, then an *extended move-down step* shifts the window to the top of the next column, that is, if the central position of the window is on the bottom-most field of column j for some $j < \text{cols}(P)$, then it is moved to the top-most field of column $j + 1$.
- (3) In any cycle, as soon as \mathcal{M} executes an extended move-right (move-down) step, then for the rest of this cycle, it cannot execute any extended move-down (move-right) steps.

Finally, \mathcal{M} is called a *stateless det-2D-x2W-ORWW-automaton* (or a *stl-det-2D-x2W-ORWW-automaton*) if it has just a single state.

Obviously, for one-row pictures the det-2D-x2W-ORWW-automaton behaves exactly as the det-2D-2W-ORWW-automaton. However, it is known that stl-det-2D-x2W-ORWW-automata are strictly weaker in expressive power than the variants with states [10, 16], and from Theorem 2.2 we see that they are at least as expressive as det-2D-2W-ORWW-automata.

Together with a homogeneous morphism $\varphi : \Gamma \rightarrow \Delta^{h,b}$, a (stateless) det-2D-x2W-ORWW-automaton $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ defines a transduction $\varphi_{\mathcal{M}} : \Sigma^{*,*} \rightarrow \Delta^{*,*}$ in analogy to Definition 3.1. By $\mathcal{F}_{hm}(\text{stl-det-2D-x2W-ORWW})$ and $\mathcal{F}_{hm}(\text{det-2D-x2W-ORWW})$ we denote the corresponding classes of transductions.

In combination with Lemma 4.7, the following result shows that $\mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$ is a proper subclass of $\mathcal{F}_{hm}(\text{det-2D-x2W-ORWW})$. Recall from Lemma 4.7 that τ^{VM} denotes the operation of vertical mirroring for pictures of dimension (m, n) ($m, n \geq 2$).

Theorem 5.2. $\tau^{\text{VM}} \in \mathcal{F}_{hm}(\text{det-2D-x2W-ORWW})$.

Proof. Let Σ be a finite (input) alphabet. We use the idea from the proof that the language L_{copy} that consists of all pictures $P \Phi P$, where P is any picture from $\Sigma^{n,n}$ ($n \geq 1$), is accepted by a det-2D-x2W-ORWW-automaton [16].

A det-2D-x2W-ORWW-automaton \mathcal{M} and a homogeneous morphism φ for computing τ^{VM} are obtained as follows. Let $P \in \Sigma^{m,n}$ be a given input picture, where $m, n \geq 2$. Then \mathcal{M} proceeds as follows, where we distinguish between the processing of the first row (Phase 1) and the processing of all other rows (Phase 2). Below we use the notation Σ -*syllable* to denote any maximal factor of the tape contents that only consists of letters from Σ .

Phase 1. Using the second row and the auxiliary symbols $[a, b]^{\uparrow 1}$, $[a, b]^{\uparrow 2}$ for $a, b \in \Sigma$, \mathcal{M} computes the reversal of the first row as follows. To simplify the presentation we assume in the following that n is an even number, but it is easily seen how to extend the construction also to uneven numbers.

• In each cycle \mathcal{M} scans the first row from left to right. Depending on the form of the string contained in this row, one of the following steps is executed:

- (1) If the first row contains a string from the set $\{[a, b]_2 \mid a, b \in \Sigma\}^* \cdot \Sigma^+ \cdot \{[a, b]_2 \mid a, b \in \Sigma\}^*$, then \mathcal{M} must interchange the first letter of the Σ -syllable, say c , with the last letter of this syllable, say d . Accordingly, it executes an extended move-right step, and then it replaces the symbol x in row 2 that is below the letter c by the symbol $[x, d]^{\uparrow 1}$ and restarts. In the next cycle, when \mathcal{M} encounters a symbol $c \in \Sigma$ in row 1 below which the symbol $[x, d]^{\uparrow 1}$ is written, then it rewrites c into $[c, d]_1$ and restarts.
- (2) If the first row contains a string from the set $\{[a, b]_2 \mid a, b \in \Sigma\}^* \cdot \{[a, b]_1 \mid a, b \in \Sigma\} \cdot \Sigma^+ \cdot \{[a, b]_2 \mid a, b \in \Sigma\}^*$, then \mathcal{M} must move the information on the letter $[c, d]_1$ to the last letter of the Σ -syllable. If that syllable ends with the letter d , then \mathcal{M} replaces that occurrence of d by $[d, c]_1$ and restarts, otherwise, it halts without accepting.
- (3) If the first row contains a string from the set $\{[a, b]_2 \mid a, b \in \Sigma\}^* \cdot \{[a, b]_1 \mid a, b \in \Sigma\} \cdot \Sigma^* \cdot \{[a, b]_1 \mid a, b \in \Sigma\} \cdot \{[a, b]_2 \mid a, b \in \Sigma\}^*$, then the letters with index 1 must be replaced by the corresponding letters with index 2. Accordingly, \mathcal{M} executes an extended move-right step, and then it replaces the symbol $[x, d]^{\uparrow 1}$ that is below the first letter $[c, d]_1$ by the symbol $[x, d]^{\uparrow 2}$ and restarts. In the next cycle, when \mathcal{M} encounters a symbol $[c, d]_1$ in row 1 below which the symbol $[x, d]^{\uparrow 2}$ is written, then it rewrites $[c, d]_1$ into $[c, d]_2$ and restarts.
- (4) If the first row contains a string from the set $\{[a, b]_2 \mid a, b \in \Sigma\}^+ \cdot \Sigma^* \cdot \{[a, b]_1 \mid a, b \in \Sigma\} \cdot \{[a, b]_2 \mid a, b \in \Sigma\}^*$, then \mathcal{M} replaces the symbol $[d, c]_1$ by $[d, c]_2$ and restarts.

• If the first row contains a string from $\{[a, b]_2 \mid a, b \in \Sigma\}^*$, then Phase 1 is complete. Each letter $a = P_{1,j} \in \Sigma$ has been replaced by the auxiliary symbol $[a, b]_2$, where $b = P_{1,n+1-j}$. Thus, the operation of mirroring has been executed successfully on row 1. Now, by marking all items in row 1 from right to left, this information is moved to the top leftmost field. Hence, after the next restart, \mathcal{M} realizes that it is in Phase 2.

Phase 2. For row i , $i = 2, 3, \dots, m$, \mathcal{M} proceeds as follows, where we interpret a symbol of the form $[x, c]^{\uparrow 2}$ in row 2 simply as the input symbol x . Observe that for all $i \geq 2$, \mathcal{M} already sees row i in (the lower part) of its window when it scans row $i - 1$. This means that it is not necessary to encode information on row i into any other row, which simplifies the processing of this row considerably.

• In each cycle \mathcal{M} moves right across row $i - 1$ scanning already row i with the bottom line of its window. Depending on the form of the string contained in row i , one of the following steps is executed following an extended move-right step at the end of row $i - 1$:

- (1) If row i contains a string from the set $\{[a, b]_2 \mid a, b \in \Sigma\}^* \cdot \Sigma^+ \cdot \{[a, b]_2 \mid a, b \in \Sigma\}^*$, then \mathcal{M} must interchange the first letter of the Σ -syllable, say c , with the last letter of this syllable, say d . Accordingly, it replaces the symbol c in row i by the symbol $[c, d]_1$ and restarts.
- (2) If row i contains a string from the set $\{[a, b]_2 \mid a, b \in \Sigma\}^* \cdot \{[a, b]_1 \mid a, b \in \Sigma\} \cdot \Sigma^+ \cdot \{[a, b]_2 \mid a, b \in \Sigma\}^*$, then \mathcal{M} must move the information on the letter $[c, d]_1$ to the last letter of the Σ -syllable. If that syllable ends with the letter d , then \mathcal{M} replaces that occurrence of d by $[d, c]_1$ and restarts, otherwise, it halts without accepting.
- (3) If row i contains a string from the set $\{[a, b]_2 \mid a, b \in \Sigma\}^* \cdot \{[a, b]_1 \mid a, b \in \Sigma\} \cdot \Sigma^* \cdot \{[a, b]_1 \mid a, b \in \Sigma\} \cdot \{[a, b]_2 \mid a, b \in \Sigma\}^*$, then the letters with index 1 must be replaced by the corresponding letters with index 2. Accordingly, \mathcal{M} replaces the first symbol of the form $[c, d]_1$ by the symbol $[c, d]_2$ and restarts.
- (4) If row i contains a string from the set $\{[a, b]_2 \mid a, b \in \Sigma\}^+ \cdot \Sigma^* \cdot \{[a, b]_1 \mid a, b \in \Sigma\} \cdot \{[a, b]_2 \mid a, b \in \Sigma\}^*$, then \mathcal{M} replaces the symbol of the form $[d, c]_1$ by $[d, c]_2$ and restarts.

• If row i contains a string from $\{[a, b]_2 \mid a, b \in \Sigma\}^*$, then this row is complete. Each letter $a \in \Sigma$ has been replaced by the auxiliary symbol $[a, b]_2$, where $b = P_{i, n+1-j}$, if $a = P_{i, j}$. Thus, the operation of mirroring has been executed successfully on row i . Now, by marking all items in row i from right to left, this information is moved to the leftmost field in row i . Hence, after the next restart, \mathcal{M} continues with row $i + 1$. Once all rows have been processed in this way, \mathcal{M} halts and accepts.

Finally, the homogeneous morphism φ is defined to map a symbol of the form $[a, b]_2$ to the input symbol b . Then $\varphi_{\mathcal{M}}(P) = \tau^{\text{VM}}(P)$, that is, the pair (\mathcal{M}, φ) realizes the operation of vertical mirroring on all pictures of height $m \geq 2$. □

Analogously, the operation of horizontal mirroring can be realized by a det-2D-x2W-ORWW-automaton and a homogeneous morphism. However, it appears that the operation of rotation cannot be realized in this way. Further, without states det-2D-x2W-ORWW-automata cannot realize the operations of vertical or horizontal mirroring. Here we state and prove this result for the operation of vertical mirroring.

Theorem 5.3. $\tau^{\text{VM}} \notin \mathcal{F}_{hm}(\text{stl-det-2D-x2W-ORWW})$.

Proof. Let $\Sigma = \{a, b, \square\}$ and

$$L = \{P \in \Sigma^{2, 2n} \mid n \geq 1, P_{1, i} \in \{a, b\} \text{ and } P_{2, i} = \square, 1 \leq i \leq 2n\}.$$

Assume that the operation of vertical mirroring can be realized by a stl-det-2D-x2W-ORWW-automaton $\mathcal{M} = (\Sigma, \Gamma, \mathcal{S}, \delta, >)$ and a homogeneous morphism $\varphi : \Gamma \rightarrow \Sigma$.

We now analyze the accepting computations of \mathcal{M} on pictures of the form $P_u^{(l)} \uplus P_v^{(r)} \in L$, where, for $u = a_1 a_2 \dots a_n \in \{a, b\}^n$ and $v = b_1 b_2 \dots b_n \in \{a, b\}^n$,

$$P_u^{(l)} = \begin{bmatrix} a_1 & a_2 & \dots & a_n & a \\ \square & \square & \dots & \square & \square \end{bmatrix} \text{ and } P_v^{(r)} = \begin{bmatrix} a & b_1 & b_2 & \dots & b_n \\ \square & \square & \square & \dots & \square \end{bmatrix}.$$

Given $P_u^{(l)} \uplus P_v^{(r)}$ as input, \mathcal{M} will perform an accepting computation, which consists of a finite sequence of cycles that is followed by an accepting tail computation, and which produces a picture

$$\hat{P}_u^{(l)} \uplus \hat{P}_v^{(r)} = \begin{bmatrix} c_n & \dots & c_2 & c_1 & c_0 & d_0 & d_n & \dots & d_2 & d_1 \\ e_n & \dots & e_2 & e_1 & e_0 & f_0 & f_n & \dots & f_2 & f_1 \end{bmatrix}$$

such that $\varphi(c_i) = b_i$ and $\varphi(d_i) = a_i$ for all $1 \leq i \leq n$, $\varphi(c_0) = \varphi(d_0) = a$, and $\varphi(e_j) = \varphi(f_j) = \square$ for all $0 \leq j \leq n$.

We split this computation into a finite number of *phases*, where we distinguish between four types of phases:

- (1) A *left-only phase* (O) consists of a sequence of cycles in which the window of \mathcal{M} stays on the left half of the picture.
- (2) An *upper-right phase* (U) consists of a sequence of cycles in which all rewrite steps are performed on the right half of the picture, and in addition, in the first of these cycles, \mathcal{M} enters the right half of the picture through move-right steps in row 1.
- (3) A *lower-left phase* (L) is a sequence of cycles in which all rewrite steps are performed in the left half of the picture, and in addition, the first of these cycles contains an extended move-right step.
- (4) A *lower-right phase* (R) is a sequence of cycles in which all rewrite steps are performed in the right half of the picture, and in addition, in the first of these cycles, \mathcal{M} enters the right half of the picture through a move-right step in row 2 or after executing an extended move-down step.

Obviously, the sequence of cycles of the computation of \mathcal{M} on input $P_u^{(l)} \uplus P_v^{(r)}$ can uniquely be split into a sequence of phases if we require that each phase is of maximum length. Thus, this computation can be described in a unique way by a string α over the alphabet $\Omega = \{O, U, L, R\}$.

Concerning the possible changes from one phase to the next there are some restrictions based on the fact that \mathcal{M} is stateless.

- While \mathcal{M} is in a lower-right phase (R), it just moves through the left half of the current picture after each rewrite/restart step. Thus, \mathcal{M} cannot get into another phase until it performs a rewrite step that replaces a symbol in the first column of the right half of the picture. Only then may follow a left-only phase (O) or a lower-left phase (L). However, in a fixed column, less than $2 \cdot |\Gamma|$ many rewrite steps can be performed, and so $|\alpha|_R \leq 1 + 2 \cdot |\Gamma|$ follows.
- When \mathcal{M} is in a lower-left phase (L), then it can next get into a lower-right phase (R) or into an upper-right phase (U). However, when \mathcal{M} got into the lower-left phase, then it moved all the way right across the first row. Thus, it cannot get into an upper-right phase (U) before a rewrite step is performed that replaces a symbol in the last column of the left half of the picture. As there are less than $2 \cdot |\Gamma|$ many rewrite steps that can be performed on this column, we see that $|\alpha|_L \leq 1 + |\alpha|_R + 2 \cdot |\Gamma| \leq 2 + 4 \cdot |\Gamma|$ follows.
- When \mathcal{M} is in an upper-right phase (U), then it can next get into a lower-left phase (L) or a left-only phase (O). However, when \mathcal{M} got into the upper-right phase, then it moved across the left half of the first row, and so it can get into a left-only phase only after a symbol in the first column of the right half of the picture has been rewritten. It follows that $|\alpha|_U \leq 1 + |\alpha|_L + 2 \cdot |\Gamma| \leq 3 + 6 \cdot |\Gamma|$.
- A left-only phase (O) can be followed by any other phase. Thus, we obtain that $|\alpha|_O \leq 1 + |\alpha|_R + |\alpha|_L + |\alpha|_U \leq 7 + 12 \cdot |\Gamma|$.

It follows that $|\alpha| \leq |\alpha|_O + |\alpha|_R + |\alpha|_L + |\alpha|_U \leq 13 + 24 \cdot |\Gamma|$, that is, each computation of \mathcal{M} consists of at most $13 + 24 \cdot |\Gamma|$ many phases.

Now we associate a *generalized crossing sequence* $GCS(\mathcal{M}, P_u^{(l)} \Phi P_v^{(r)})$ to the computation of \mathcal{M} on the input picture $P_u^{(l)} \Phi P_v^{(r)}$ as follows.

Let $\alpha(u, v) \in \{O, U, L, R\}^+$ be the description of the sequence of phases of the accepting computation of \mathcal{M} on input $P_u^{(l)} \Phi P_v^{(r)}$. After each letter X of $\alpha(w)$, we insert a 2-by-2 picture $\begin{pmatrix} c & d \\ e & f \end{pmatrix}$ such that $\begin{pmatrix} c \\ e \end{pmatrix}$ is the contents of the rightmost column of the left half and $\begin{pmatrix} d \\ f \end{pmatrix}$ is the contents of the leftmost column of the right half of the picture at the end of the phase represented by the letter X . Thus, $GCS(\mathcal{M}, P_u^{(l)} \Phi P_v^{(r)})$ is a string of length at most $26 + 48 \cdot |\Gamma|$ over the finite alphabet $\Omega \cup \Gamma^{2,2}$ of size $4 + |\Gamma|^4$, that is, there are only finitely many different such crossing sequences.

If n is sufficiently large, then there are three strings $u, v, w \in \{a, b\}^n$, $u = a_1 a_2 \dots a_n$, $v = b_1 b_2 \dots b_n$, and $w = b'_1 b'_2 \dots b'_n$, such that $v \neq w$, but $GCS(\mathcal{M}, P_u^{(l)} \Phi P_v^{(r)}) = GCS(\mathcal{M}, P_u^{(l)} \Phi P_w^{(r)})$. The computation of M on input $P_u^{(l)} \Phi P_v^{(r)}$ yields a final picture of the form

$$\tilde{P}_u^{(l)} \Phi \tilde{P}_v^{(r)} = \begin{bmatrix} c_n \dots c_2 c_1 c_0 d_0 d_n \dots d_2 d_1 \\ e_n \dots e_2 e_1 e_0 f_0 f_n \dots f_2 f_1 \end{bmatrix}$$

such that $\varphi(c_i) = b_i$ and $\varphi(d_i) = a_i$ for all $1 \leq i \leq n$, $\varphi(c_0) = \varphi(d_0) = a$, and $\varphi(e_j) = \varphi(f_j) = \square$ for all $0 \leq j \leq n$, and the computation of M on input $P_u^{(l)} \Phi P_w^{(r)}$ yields a final picture of the form

$$\hat{P}_u^{(l)} \Phi \hat{P}_w^{(r)} = \begin{bmatrix} c'_n \dots c'_2 c'_1 c'_0 d'_0 d'_n \dots d'_2 d'_1 \\ e'_n \dots e'_2 e'_1 e'_0 f'_0 f'_n \dots f'_2 f'_1 \end{bmatrix}$$

such that $\varphi(c'_i) = b'_i$ and $\varphi(d'_i) = a_i$ for all $1 \leq i \leq n$, $\varphi(c'_0) = \varphi(d'_0) = a$, and $\varphi(e'_j) = \varphi(f'_j) = \square$ for all $0 \leq j \leq n$. However, as the two generalized crossing sequences coincide, we see that $c'_i = c_i$ and $e'_i = e_i$ for all $1 \leq i \leq n$. This implies that $b'_i = \varphi(c'_i) = \varphi(c_i) = b_i$ for all $1 \leq i \leq n$, which means that $v = b_1 b_2 \dots b_n = b'_1 b'_2 \dots b'_n = w$, contradicting our choice of v and w . Thus, the operation of vertical mirroring cannot be realized by a stl-det-2D-x2W-ORWW-automaton in combination with a homogeneous morphism. \square

Together with Theorem 5.2 this shows that $\mathcal{F}_{hm}(\text{stl-det-2D-x2W-ORWW})$ is a proper subclass of $\mathcal{F}_{hm}(\text{det-2D-x2W-ORWW})$. Next, we have the following closure property for the transformations that can be realized by (stateless) det-2D-x2W-ORWW-automata.

Theorem 5.4. *Let $\mathcal{M}_1 = (Q_1, \Sigma_1, \Gamma_1, \mathcal{S}, q_1, \delta_1, >_1)$ be a det-2D-x2W-ORWW-automaton, let $\pi_1 : \Gamma_1 \rightarrow \Sigma_2$ be a projection, and let $\tau_1 : \Sigma_1^{*,*} \rightarrow \Sigma_2^{*,*}$ be the transduction that is realized by (\mathcal{M}_1, π_1) . Further, let $\mathcal{M}_2 = (Q_2, \Sigma_2, \Gamma_2, \mathcal{S}, q_2, \delta_2, >_2)$ be a det-2D-x2W-ORWW-automaton, let $\varphi_2 : \Gamma_2^{*,*} \rightarrow \Delta^{*,*}$ be a homogeneous morphism, and let $\tau_2 : \Sigma_2^{*,*} \rightarrow \Delta^{*,*}$ denote the transduction that is realized by $(\mathcal{M}_2, \varphi_2)$. Then there exist a det-2D-x2W-ORWW-automaton \mathcal{M} and a homogeneous morphism φ such that (\mathcal{M}, φ) realizes the transduction $\tau = \tau_2 \circ \tau_1$. Moreover, if \mathcal{M}_1 and \mathcal{M}_2 are stateless, then also \mathcal{M} can be chosen to be stateless.*

Proof. The arguments given in the proof of Lemma 4.1 also apply to (stl-)det-2D-x2W-ORWW-automata. The only problem occurs in that in the proof of that lemma, the (stl-)det-2D-x2W-ORWW-automaton \mathcal{M} must distinguish between two parts of its computation. This was achieved by marking the shortest path from the top leftmost position of the current picture to the position (i, j) at which the automaton \mathcal{M}_1 accepts.

The (stl-)det-2D-x2W-ORWW-automaton \mathcal{M} also marks this position and restarts. Now \mathcal{M} should simulate \mathcal{M}_2 . However, unless the marked position (i, j) happens to be inside the initial position of the window, \mathcal{M} does not know that it now should simulate \mathcal{M}_2 . Therefore, it still behaves just like \mathcal{M}_1 during the tail of its accepting computation. Hence, no rewrite will be performed, but the marked position (i, j) will be reached eventually. Now \mathcal{M} adds a mark also to the position (i', j') that it reached prior to the position (i, j) . However, here a problem arises if position (i, j) was reached by an extended move-right (move-down) step, as then the previous position (i', j') is not inside the window when the position (i, j) is reached. In that case, \mathcal{M} marks the first unmarked position to the right of (below) position (i, j) . Continuing in this manner, the final part of row i (the lower part of column j) will eventually be completely marked. Hence, when \mathcal{M} reaches position (i', j') , then it already sees that the symbol in the next row (column) is marked, and instead of performing an extended move-right (move-down) step, the symbol at position (i', j') is marked. Thus, after finitely many cycles a path from the position (i, j) , at which \mathcal{M}_1 accepted, to the initial position $(1, 1)$ is completely marked. For the case of stateless automata, now all cells must be marked so that, afterwards, \mathcal{M} always realizes that it is to simulate \mathcal{M}_2 without having to remember this fact. \square

Theorem 5.4 says that the classes $\mathcal{F}_{hm}(\text{stl-det-2D-x2W-ORWW})$ and $\mathcal{F}_{hm}(\text{det-2D-x2W-ORWW})$ of transductions are closed under a restricted form of composition. However, the latter is not closed under general composition.

Theorem 5.5. (a) *The class of picture languages $\mathcal{L}(\text{det-2D-x2W-ORWW})$ is not closed under inverse homogeneous morphisms.*

(b) *The class of transductions $\mathcal{F}_{hm}(\text{det-2D-x2W-ORWW})$ is not closed under composition.*

Proof. Let $\Sigma = \{a, b\}$ and $\Gamma = \{a, b, \square\}$. We define the languages

$$L_1 = \{P \in \Sigma^{1,2n} \mid n \geq 1, P_{1,1} \dots P_{1,n} = (P_{1,n+1} \dots P_{1,2n})^\rho\}$$

where $\rho : \Sigma^* \rightarrow \Sigma^*$ denotes the operation of *reversal* for strings, and

$$L_2 = \{P \in \Gamma^{2,2n} \mid n \geq 1, P_{1,1} \dots P_{1,n} = (P_{1,n+1} \dots P_{1,2n})^\rho, P_{1,i} \in \{a, b\} \text{ and } P_{2,i} = \square \text{ for all } 1 \leq i \leq 2n\}.$$

We utilize two results from [10]. First, it is known that det-2D-x2W-ORWW-automata working over one-row pictures (that is, strings) only accept regular languages. Secondly, the picture language L_2 is accepted by a det-2D-x2W-ORWW-automaton.

(a) Consider the homogeneous morphism $\varphi : \Sigma^{*,*} \rightarrow \Gamma^{*,*}$ that is given by $\varphi(a) = \begin{bmatrix} a \\ \square \end{bmatrix}$ and $\varphi(b) = \begin{bmatrix} b \\ \square \end{bmatrix}$. Then $\varphi^{-1}(L_2) = \{P \mid \varphi(P) \in L_2\} = L_1$, which is not accepted by a det-2D-x2W-ORWW-automaton, as it is not regular. This proves (a).

(b) Next we define a transduction $\tau_1 : \Sigma^{1,*} \rightarrow \Gamma^{2,*}$, which extends each one-row picture over Σ by an additional row of symbols \square , and a transduction $\tau_2 : \Gamma^{2,*} \rightarrow \Sigma^{2,*}$, which transforms each $P \in L_2$ of dimension $(2, 2n)$ to $a^{2,2n}$ and each $P \notin L_2$ of dimension $(2, n)$ to $b^{2,n}$. Both these transductions are in $\mathcal{F}_{hm}(\text{det-2D-x2W-ORWW})$.

Assume that $\tau_2 \circ \tau_1$ is realized by (\mathcal{M}, φ') , where \mathcal{M} is a det-2D-x2W-ORWW-automaton and φ' is a homogeneous morphism. Then

$$\varphi_M(P) = \tau_2(\tau_1(P)) = \begin{bmatrix} a & a & \cdots & a \\ a & a & \cdots & a \end{bmatrix}, \text{ if } P \in L_1, \text{ and}$$

$$\varphi_M(P) = \tau_2(\tau_1(P)) = \begin{bmatrix} b & b & \cdots & b \\ b & b & \cdots & b \end{bmatrix}, \text{ if } P \notin L_1.$$

Thus, from \mathcal{M} we can easily obtain a det-2D-x2W-ORWW-automaton for the language L_1 , which contradicts the observation above. This proves (b). \square

6. DECISION PROBLEMS

Computing the transduction $\varphi_{\mathcal{M}}$ realized by a det-2D-2W-ORWW-automaton or a det-2D-x2W-ORWW-automaton \mathcal{M} and a homogeneous morphism φ on a given input picture only requires polynomial time. However, we will see that the problem of deciding whether there exists a picture P' such that $\varphi_{\mathcal{M}}(P') = P$ for a given picture P is NP-complete.

The following lemma says that each det-2D-2W-ORWW-automaton can be modified in such a way that the input can be reconstructed at the end of any accepting computation.

Lemma 6.1. *For each $L \in \mathcal{L}(\text{det-2D-2W-ORWW})$, there is a det-2D-2W-ORWW-automaton $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ and a projection $\varphi : \Gamma \rightarrow \Sigma$ such that $L = L(\mathcal{M})$ and (\mathcal{M}, φ) realizes the identity.*

Proof. Let $\mathcal{M}_1 = (Q, \Sigma, \Gamma_1, \mathcal{S}, q_0, \delta_1, >_1)$ be a det-2D-2W-ORWW-automaton. We take $\Gamma_2 = \Sigma \cup (\Sigma \times \Gamma_1)$ and define the projections $\varphi, \pi : \Gamma_2 \rightarrow \Sigma$ by $\varphi(a) = \pi(a) = a$ for all $a \in \Sigma$, and $\varphi((a, b)) = a$, $\pi((a, b)) = b$ for all $(a, b) \in \Sigma \times \Gamma_1$. Let $\mathcal{M}_2 = (Q, \Sigma, \Gamma_2, \mathcal{S}, q_0, \delta_2, >_2)$, where we define the ordering $>_2$ by $c >_2 d$ iff $\pi(c) >_1 \pi(d)$ ($c, d \in \Gamma_2$) and the transition function δ_2 as follows:

$$\begin{aligned} \delta_2(q, C) &= (q', R), \text{ if } \delta_1(q, \pi(C)) = (q', R), \\ \delta_2(q, C) &= (q', D), \text{ if } \delta_1(q, \pi(C)) = (q', D), \\ \delta_2(q, C) &= (a, c), \text{ if } (C_{2,2} = a \text{ or } C_{2,2} = (a, b)) \text{ and } \delta_1(q, \pi(C)) = c, \\ \delta_2(q, C) &= \text{Accept}, \text{ if } \delta_1(q, \pi(C)) = \text{Accept}. \end{aligned}$$

Then, $L(\mathcal{M}_1) = L(\mathcal{M}_2)$, and the pair (\mathcal{M}_2, φ) realizes the identity. \square

For a picture P , we denote by $B_{h,w}(P)$ the set of all its subpictures of dimension (h, w) . A *tile* is simply a square picture of dimension $(2, 2)$. A picture language $L \subseteq \Sigma^{*,*}$ is called a *local picture language*, if there exists a finite set Θ of tiles over $\Sigma \cup \{\#\}$ such that $L = \{P \in \Sigma^{*,*} \mid B_{2,2}(\widehat{P}) \subseteq \Theta\}$. The family REC of recognizable picture languages can be characterized as the set of languages that are obtained by projections from local picture languages (see, e.g., [4]). Below we show that every recognizable picture language can be obtained as the image of a transduction from $\mathcal{F}_{hm}(\text{det-2D-2W-ORWW})$. This result will then be further utilized to study the complexity of decision problems on transductions.

Proposition 6.2. *Let $L \subseteq \Delta^{*,*}$ be a picture language from REC. Then there are a det-2D-2W-ORWW-automaton $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ and a projection $\theta : \Gamma \rightarrow \Delta$ such that $L = \theta_{\mathcal{M}}(L(\mathcal{M}))$.*

Proof. Let $L \subseteq \Delta^{*,*}$ be a picture language from REC. Then there exist a local language $L' \in \Sigma^{*,*}$ on some alphabet Σ and a projection $\pi : \Sigma \rightarrow \Delta$ such that $L = \pi(L')$. Every local picture language is accepted by a deterministic four-way finite automaton, which can in turn be simulated by a det-2D-2W-ORWW-automaton (in fact, even the stronger deterministic two-dimensional on-line tessellation automaton can be simulated [14]). Hence, by Lemma 6.1, there are a det-2D-2W-ORWW-automaton $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$ and a projection $\varphi : \Gamma \rightarrow \Sigma$ such that $L(\mathcal{M}) = L'$ and the pair (\mathcal{M}, φ) realizes the identity on L' . It follows that $(\pi \circ \varphi)_{\mathcal{M}}(L(\mathcal{M})) = L$, since $\varphi_{\mathcal{M}}(L(\mathcal{M})) = L'$. \square

From this proposition and the fact that REC contains languages for which the membership problem is NP-complete [9], we now obtain the following complexity result.

Corollary 6.3. *The following problem is NP-complete:*

Instance: A det-2D-2W-ORWW-automaton $\mathcal{M} = (Q, \Sigma, \Gamma, \mathcal{S}, q_0, \delta, >)$, a morphism $\varphi : \Gamma^{*,*} \rightarrow \Delta^{*,*}$, and a picture $P \in \Delta^{*,*}$.
 Question: Is $P \in \varphi_{\mathcal{M}}(L(\mathcal{M}))$?

Proof. It remains to show that the above problem is in NP. Let $P \in \Delta^{*,*}$ be a given picture. Then we can simply guess a picture $P' \in \Sigma^{*,*}$ such that $\text{rows}(P') \leq \text{rows}(P)$ and $\text{cols}(P') \leq \text{cols}(P)$ and check in polynomial time whether $P = \varphi(\tilde{P}')$ holds. \square

Proposition 6.2 can also be applied to show that the “family of transduction images” inherits the undecidability of problems that are not decidable for REC like, e.g., the emptiness and the finiteness problems (see [3]).

7. CONCLUSION

By combining a det-2D-2W-ORWW-automaton or a det-2D-x2W-ORWW-automaton with a homogeneous two-dimensional morphism, we obtained a device for computing transductions of pictures. As in the one-dimensional setting the corresponding devices characterize the rational functions mapping the empty string to itself, the two-dimensional transductions computed by our devices can be seen as defining classes of rational functions for pictures. These classes contain all spatial filters that are used in digital image processing, for det-2D-2W-ORWW-automata, the corresponding class is closed under composition, and for (stl-)det-2D-x2W-ORWW-automata, it is closed under a restricted form of composition. While the operations of horizontal and vertical mirroring are beyond the power of det-2D-2W-ORWW- and stl-det-2D-x2W-ORWW-automata, they can be realized by det-2D-x2W-ORWW-automata. Unfortunately, the class of transductions computed by the latter is not closed under composition, and it remains open whether or not the operation of rotation can be realized by these automata. Also it remains open whether the class of transductions computed by stateless det-2D-x2W-ORWW-automata is closed under general composition.

Observe that the idea of extending a det-2D-2W-ORWW- or a det-2D-x2W-ORWW-automaton with a homogeneous morphism to obtain a realization of a transduction on pictures can also be applied to the deterministic two-dimensional three-way ORWW-automata introduced in [11], to the (deterministic) Sgraffito automaton [18], and to the (deterministic) restarting tiling automaton [17]. Actually, it appears that all our results on the transductions that are realizable by det-2D-2W-ORWW-automata carry over to deterministic Sgraffito automata. However, it remains the question of whether or not the latter automata are more expressive than the former.

REFERENCES

- [1] A. Anselmo, D. Giammarresi and M. Madonia, From determinism to non-determinism in recognizable two-dimensional languages. In *DLT 2007*. In vol. 4588 of *Lect. Notes Comput. Sci.* Edited by T. Harju, J. Karhumäki and A. Lepistö. Springer, Heidelberg (2007) 36–47.
- [2] A. Anselmo, D. Giammarresi and M. Madonia, A computational model for tiling recognizable two-dimensional languages. *Theor. Comput. Sci.* **410** (2009) 3520–3529.

- [3] D. Giammarresi and A. Restivo, Recognizable picture languages. *Int. J. Pattern Recognition and Artificial Intelligence* **6** (1992) 241–256.
- [4] D. Giammarresi and A. Restivo, Two-dimensional languages. In vol. 3 of *Handbook of Formal Languages*. Edited by G. Rozenberg and A. Salomaa. Springer, New York, NY, USA (1997) 215–267.
- [5] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 3rd edition (2007).
- [6] P. Jančar, F. Mráz, M. Plátek and J. Vogel, On monotonic automata with a restart operation. *J. Autom. Lang. Comb.* **4** (1999) 287–311.
- [7] K. Kwee and F. Otto, On some decision problems for stateless deterministic ordered restarting automata. In *DCFS 2015, Proc.* In vol. 9118 of *Lect. Notes Comput. Sci.* Edited by J. Shallit and A. Okhotin. Springer, Heidelberg (2015) 165–176.
- [8] K. Kwee and F. Otto, On the effects of nondeterminism on ordered restarting automata. In *SOFSEM 2016, Proc.* In vol. 9587 of *Lect. Notes Comput. Sci.* Edited by R.M. Freivalds, G. Engels and B. Catania. Springer, Heidelberg (2016) 369–380.
- [9] K. Lindgren, C. Moore and M. Nordahl, Complexity of two-dimensional patterns. *J. Stat. Phys.* **91** (1998) 909–951.
- [10] F. Mráz and F. Otto, Extended two-way ordered restarting automata for picture languages. In *LATA 2014*. In vol. 8370 of *Lect. Notes Comput. Sci.* Edited by A.-H. Dediu, C. Martín-Vide, J.-L. Sierra-Rodríguez and B. Truthe. Springer, Heidelberg (2014) 541–552.
- [11] F. Mráz and F. Otto, Ordered restarting automata for picture languages. In *SOFSEM 2014, Proc.* In vol. 8327 of *Lect. Notes Comput. Sci.* Edited by V. Geffert, B. Preneel, B. Rován, J. Štuller and A. Min Tjoa. Springer, Heidelberg (2014) 431–442.
- [12] F. Mráz, F. Otto and D. Průša, On a class of rational functions for pictures. In *Seventh Workshop on Non-Classical Models of Automata and Applications (NCMA 2015), Proc.* In vol. 318 of *books@ocg.at*. Edited by R. Freund, M. Holzer, N. Moreira and R. Reis Oesterreichische Computer Gesellschaft, Wien (2015) 159–176.
- [13] F. Otto, On the descriptive complexity of deterministic ordered restarting automata. In *DCFS 2014, Proc.* In vol. 8614 of *Lect. Notes Comput. Sci.* Edited by H. Jürgensen, J. Karhumäki and A. Okhotin. Springer, Heidelberg (2014) 318–329.
- [14] F. Otto, Restarting automata for picture languages: A survey on recent developments. In *CIAA 2014, Proc.* In vol. 8587 of *Lect. Notes Comput. Sci.* Edited by M. Holzer and M. Kutrib. Springer, Heidelberg (2014) 16–41.
- [15] F. Otto and K. Kwee, Deterministic ordered restarting automata that compute functions. In *DLT 2015, Proc.* In vol. 9168 of *Lect. Notes Comput. Sci.* Edited by I. Potapov. Springer, Heidelberg (2015) 401–412.
- [16] F. Otto and F. Mráz, Deterministic ordered restarting automata for picture languages. *Acta Informatica* **52** (2015) 593–623.
- [17] D. Průša and F. Mráz, Restarting tiling automata. *Int. J. Found. Comput. Sci.* **24** (2013) 863–878.
- [18] D. Průša, F. Mráz and F. Otto, Two-dimensional Sgraffito automata. *RAIRO: ITA* **48** (2014) 505–539.

Communicated by N. Moreira.

Received December 18, 2015. Accepted October 28, 2016.