INFORMATION AND COMPUTATION **125**, 32–45 (1996)
ARTICLE NO. 0018

Monadic Second-Order Logic over Rectangular Pictures and Recognizability by Tiling Systems*

DORA GIAMMARRESI

Dipartimento di Matematica, Università dell'Aquila, via Vetoio, Loc. Coppito, 67100 L'Aquila, Italy
E-mail address: dora@smaq20.univaq.it

ANTONIO RESTIVO

Dipartimento di Matematica e Applicazioni, Università di Palermo, via Archirafi, 34, 90123 Palermo, Italy
E-mail address: restivo@altair.math.unipa.it

AND

SEBASTIAN SEIBERT AND WOLFGANG THOMAS†

Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität Kiel, Olshausenstrasse 40, 24098 Kiel, Germany
E-mail address: {ss,wt}@informatik.uni-kiel.d400.de

It is shown that a set of pictures (rectangular arrays of symbols) is recognized by a finite tiling system iff it is definable in existential monadic second-order logic. As a consequence, finite tiling systems constitute a notion of recognizability over two-dimensional inputs which at the same time generalizes finite-state recognizability over strings and also matches a natural logic. The proof is based on the Ehrenfeucht–Fraïssé technique for first-order logic and an implementation of "threshold counting" within tiling systems. © 1996 Academic Press, Inc.

1. INTRODUCTION

In the present paper, "pictures" are two-dimensional rectangular arrays of symbols of a given alphabet. (Another possible term instead of "picture" would be "two-dimensional word"; here we decide for the shorter term, as done in [8].) While over one-dimensional arrays of symbols (i.e., strings) the notion of finite-state recognizability is well established and known to be very robust, a comparable theory of "recognizable picture languages" is not yet known. One reason is that a generalization of finite string automata to the two-dimensional case is possible in different ways (a detailed survey can be found in [14]), another is so far the lack of elegant connections between these automaton models and logic-based formalisms or algebraic notions of recognizability. Such connections are well-known for string languages, notably Büchi's characterization of recognizable string languages in monadic second-order logic and the algebraic notion of recognizability in terms of finite monoids. In the context of graphs, a divergence of these approaches is apparent from the work of Courcelle [3]. In his algebraic theory of recognizable graph languages, it turns out, for example, that there are "recognizable" non-recursive sets of grid graphs (i.e., pictures), which (by non-recursiveness) can neither be defined in monadic second-order logic nor by any natural version of finite-state automaton on graphs.

In the present paper we take up two recent proposals to introduce "recognizability" for sets of pictures (picture languages) by finite-state devices, introduced by Giammarresi and Restivo [8] and Thomas [18], respectively. In both cases the idea is to define recognizability by "projections of local properties". The definition of [8] emphasizes conceptual simplicity and a close relation to conventional finite automata, while in [18] more general graphs than pictures are included and a bridge to existential monadic second-order logic is established. The main result of the present paper states that both approaches are equivalent for picture languages (and thus are equivalent to existential monadic second order logic).

Let us sketch these two approaches informally. In the framework of [8] and [9], recognition is defined in terms of a finite set of square pictures of dimension 2 which correspond somehow to automaton transitions and are

* Work supported by the ESPRIT Basic Research Actions Program of the EC under Project ASMICS 2 (Contract 6317). An extended abstract of this paper appeared in "Proc. STACS'94," Springer Lecture Notes in Computer Science, Vol. 775, Springer-Verlag, New York/Berlin, 1994.

† To whom correspondence should be addressed.

0890-5401/96 $18.00

called "tiles" here. In a picture to be recognized (say over the alphabet Σ), each quadruple of positions that form a square is to be covered by a tile (with symbols say in the alphabet Γ) such that a coherent assignment of picture positions to labels in Γ is built up, and such that a projection from Γ to Σ reestablishes the considered picture. If the symbols in Γ are pairs of the form (picture symbol, "state") and the projection to Σ just cancels the second component, then the tiles can be viewed as local "automaton transitions", and tiling a given picture means to construct a run of the automaton on it. On the other hand, as shown in [15], this notion of recognizability is equivalent to another one defined by means of a particular kind of cellular automata called 2-OTA in [12].

The approach of [18] refers to the more general setting of graphs of bounded degree. The idea of "tiling by transitions" is applied in a similar way as above; however, instead of 2×2-squares larger local neighbourhoods of a graph are associated with transitions (or tiles). Furthermore, the resulting covering of a graph by such tiles is required to satisfy so-called "constraints", i.e. conditions stating that certain tiles should occur at least (or at most) a certain number of times. By results from the model theory of graphs, recognizability of a graph set by such a graph accepting device is equivalent to definability in existential monadic second-order logic (i.e., using formulas which start with existential quantifiers over sets of graph vertices, followed by a first-order formula).

The main result below states that finite tiling systems of 2×2-tiles can specify exactly the same picture properties as the more general graph acceptors, and hence over pictures are equivalent in expressive power to existential monadic second-order logic. (As a consequence, all first-order properties of pictures are captured by finite tiling systems.) All results together indicate that recognizability of picture languages in the present sense is also a robust notion, which at the same time has a natural logical meaning and generalizes in a straightforward way the automaton recognizability for sets of strings.

As a more technical application of the main result we obtain that the class of picture languages definable in existential monadic second-order logic is not closed under complement. For this purpose, we use the reduction of existential monadic second-order logic to finite tiling systems, as carried out in the present paper, and apply an elementary combinatorial argument due to [9] and reproduced here for the reader's convenience. This simple proof is in some contrast to the involved proof of Fagin [6] and its more recent (and streamlined) version in [7], which applies an extension of Ehrenfeucht–Fraïssé games and shows nonclosure under complement of existential monadic second-order logic with respect to the class of arbitrary finite graphs. From nonclosure of recognizable picture language under complement we can also infer that it is not possible to eliminate the nondeterminism in our tilings (as long as deterministic versions allow complementation).

Recognizable picture languages do not share some properties that are fundamental in the theory of recognizable string languages (or tree languages). We already mentioned that, contrary to the case of words or trees, the class of recognizable picture languages is not closed under complementation. Furthermore, while the emptiness problem is decidable for finite automata over words or trees, it is undecidable for finite tiling systems over rectangular pictures (using a reduction to the halting problem for Turing machines, see e.g. [8]).

The paper is structured as follows: In Section 2 we collect the necessary terminology, summarize the basic facts on recognizability by finite tiling systems, and introduce the logical framework. Section 3 gives the proof of the main result; here the essential point is the step from the general graph acceptors in the sense of [18] to the finite tiling systems of [8] and [9], i.e. the reduction of tiles to size 2×2 and the elimination of the "occurrence constraints". In the final section, we add some comments on open questions, in particular on possible approximations of the class of recognizable picture languages by calculi of regular expressions.

2. PRELIMINARIES

2.1. *Basic Notations*

We first introduce some basic notations and terminology about pictures and picture languages. For further background see the survey [14].

Let Σ be a finite alphabet. A *picture* over Σ is a two-dimensional rectangular array of elements of Σ. Given a picture p, let $\ell_1(p)$ and $\ell_2(p)$ denote the number of rows and columns, respectively, of p. The pair $(\ell_1(p), \ell_2(p))$ is called the *size* of the picture p. Furthermore, if $1 \leqslant i \leqslant \ell_1(p)$ and $1 \leqslant j \leqslant \ell_2(p)$, we let $p(i, j)$ denote the symbol in p with coordinates (i, j). A picture p of size (m, n) can thus be viewed as a function:

$$p\colon \{1, 2, ..., m\} \times \{1, 2, ..., n\} \to \Sigma$$

The set of all pictures over the alphabet Σ is denoted by Σ^{**}. A *picture language* over Σ is a subset of Σ^{**}. For two pictures $p \in \Sigma^{**}$, $c \in Q^{**}$ of the same size (m, n), we define the product $p \times c \in (\Sigma \times Q)^{**}$ by $(p \times c)(i, j) = (p(i, j), c(i, j))$.

In describing pictures, we we will often need to identify the symbols that are on the border. If p is a picture of size (m, n), we indicate by $\hat{p}$ the picture of size $(m+2, n+2)$ obtained by surrounding p with a special boundary symbol $\# \notin \Sigma$.

$$\hat{p} = \begin{matrix} \# & \# & \cdots & \# & \# \\ \# & p_{11} & \cdots & p_{1n} & \# \\ \vdots & & \ddots & & \vdots \\ \# & p_{m1} & \cdots & p_{mm} & \# \\ \# & \# & \cdots & \# & \# \end{matrix}.$$

Using the "function notation", we have:

$$\hat{p}\colon \{0, 1, ..., m+1\} \times \{0, 1, ..., n+1\} \to \Sigma \cup \{\#\}$$

and $\hat{p}(0, j) = \hat{p}(i, 0) = \hat{p}(m+1, j) = \hat{p}(i, n+1) = \#$, for $0 \leqslant i \leqslant m+1$, $0 \leqslant j \leqslant n+1$.

A different approach to describe pictures is that of considering them as model theoretic structures. We use here notations similar to [18] where general labelled graphs are considered in the framework of relational structures.

A picture p over an alphabet Σ of size (m, n) can be viewed as a vertex-labelled graph of $m \times n$ vertices. It can be represented as the following relational structure:

$$\underline{p} \quad = (\mathrm{dom}(p), S_1, S_2, (P_a)_{a \in \Sigma})$$

where:

— $\mathrm{dom}(p) = \{1, 2, ..., m\} \times \{1, 2, ..., n\}$

— S_1 and S_2 are the *successor relations* for the two components of points of $\mathrm{dom}(p)$, that is: $(i, j)\ S_1 (i+1, j)$ for $1 \leqslant i < m$, $1 \leqslant j \leqslant n$, and $(i, j)\ S_2\ (i, j+1)$ for $1 \leqslant i \leqslant m$, $1 \leqslant j < n$

— $P_a = \{(i, j) |\ p(i, j) = a\}$ for $a \in \Sigma$ gives the set of points in $\mathrm{dom}(p)$ that are labelled with a.

For a concise description of picture languages we recall from [14] some definitions of concatenation operations between pictures.

Let r and s be two pictures over an alphabet Σ of size (m, n) and (m', n') respectively.

$$r = \begin{matrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{matrix} \qquad s = \begin{matrix} s_{11} & \cdots & s_{1n'} \\ \vdots & \ddots & \vdots \\ s_{m'1} & \cdots & s_{m'n'} \end{matrix}.$$

The *column concatenation* and the *row concatenation* of r and s (denoted by $r \oplus s$ and $r \ominus s$ respectively) are partial operations, defined only if $m = m'$ and $n = n'$ respectively, and are given by:

$$r \oplus s = \begin{matrix} r_{11} & \cdots & r_{1n} & s_{11} & \cdots & s_{1n'} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} & s_{m'1} & \cdots & s_{m'n'} \end{matrix}$$

$$r \ominus s = \begin{matrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \\ s_{11} & \cdots & s_{1n'} \\ \vdots & \ddots & \vdots \\ s_{m'1} & \cdots & s_{m'n'} \end{matrix}$$

Similarly to string language theory, concatenation operations between pictures can be extended to picture languages. Let L_1, L_2 be picture languages over an alphabet Σ, the *column concatenation* of L_1 and L_2 (denoted by $L_1 \oplus L_2$) is defined by

$$L_1 \oplus L_2 = \{x \oplus y \mid x \in L_1 \text{ and } y \in L_2\}.$$

Analogously the *row concatenation* of L_1 and L_2 (denoted by $L_1 \ominus L_2$) is defined by

$$L_1 \ominus L_2 = \{x \ominus y \mid x \in L_1 \text{ and } y \in L_2\}.$$

Furthermore, given a picture language L, the *column Kleene closure* of L (denoted by $L^{*\oplus}$) and the *row Kleene closure* of L (denoted by $L^{*\ominus}$) are defined respectively by

$$L^{*\oplus} = \bigcup_i L^{\oplus i}, \qquad L^{*\ominus} = \bigcup_i L^{\ominus i}$$

where $L^{\oplus 1} = L$, $L^{\oplus n} = L^{\oplus (n-1)} \oplus L$ and $L^{\ominus 1} = L$, $L^{\ominus n} = L^{\ominus (n-1)} \ominus L$.

2.2. *Recognizability by "Finite Tiling Systems"*

In this section we recall a definition of recognizability for picture languages that was introduced in [8]. We use here notations slightly different from [8] and [9] in order to make them compatible to notations of model theory.

Let Σ_1 and Σ_2 be two finite alphabets such that $|\Sigma_1| \geqslant |\Sigma_2|$ and $\pi\colon \Sigma_1 \to \Sigma_2$ a mapping. The *projection by mapping* π *of a picture* $p \in \Sigma_1^{**}$ is the picture $p' \in \Sigma_2^{**}$ of same dimension as p such that $p'(i, j) = \pi(p(i, j))$ for all $1 \leqslant i \leqslant \ell_1(p)$, $1 \leqslant j \leqslant \ell_2(p)$. We use the notation $\pi(p)$ to indicate the projection of picture p by mapping π. The definition of projection of a picture can be extended in a natural way to sets of pictures. Given a picture language $L \subseteq \Sigma_1^{**}$, the *projection of* L *by* $\pi\colon \Sigma_1 \to \Sigma_2$ is defined as $\pi(L) = \{\pi(p) | p \in L\} \subseteq \Sigma_2^{**}$.

Given a picture p of size (m, n), if $h \leqslant m$, $k \leqslant n$, we denote by $T_{h,k}(p)$ the set of all subpictures (i.e., contiguous rectangular subblocks) of p of size (h, k). Let Γ be a finite alphabet.

DEFINITION 2.1. A picture language $L \subseteq \Gamma^{**}$ is *local* if there exists a set Δ of pictures (or "tiles") of size (2, 2) over $\Gamma \cup \{\#\}$, such that $L = \{p \in \Gamma^{**} \mid T_{2,2}(\hat{p}) \subseteq \Delta\}$.

If $L = \{p \in \Gamma^{**} \mid T_{2,2}(\hat{p}) \subseteq \Delta\}$ we call Δ a *local representation by tiles* for the language L. We denote by LOC the family of *local picture languages*. Notice that the definition of LOC extends the classical notion of local word languages (cf. [4]) to two dimensions.

As an example of a picture language in LOC we mention the set $L_0 \subseteq \{0, 1\}^{**}$ of square pictures (of size at least (2, 2)) in which all nondiagonal positions carry symbol 0 whereas the diagonal positions (of the form (i, i)) carry symbol 1. An appropriate set of tiles consists of the 16 different (2, 2)-subblocks of the following picture:

#	#	#	#	#	#
#	1	0	0	0	#
#	0	1	0	0	#
#	0	0	1	0	#
#	0	0	0	1	#
#	#	#	#	#	#

The family of recognizable picture languages is defined by means of the family LOC and projections as follows.

DEFINITION 2.2. A picture language $L \subseteq \Sigma^{**}$ is *recognizable* if there exists a local language L' over an alphabet Γ and a mapping $\pi: \Gamma \to \Sigma$ such that $L = \pi(L')$.

An example of a recognizable picture language is the set of squares (over the one letter alphabet $\Sigma = \{a\}$). As a suitable local language we take the set L_0 of squares with marked diagonal as considered above and apply the projection $\pi: \{0, 1\} \to \{a\}$.

When dealing with recognizability it is often convenient to assume that over pictures $p \in \Sigma^{**}$ the alphabet Γ used for projection has the special form $\Gamma = \Sigma \times Q$ (and the projection $\pi: \Gamma \to \Sigma$ just cancels the Q-component). From a projection $\pi: \Gamma \to \Sigma$ with arbitrary alphabet Γ we may pass to this special case by setting $Q = \Gamma$, substituting each letter $q \in \Gamma$ by $(\pi(q), q)$ in every picture of L', and using the canonical projection from $\Sigma \times Q$ to Σ. In the sequel we usually work with this case $\Gamma = \Sigma \times Q$ and denote a tiling system by the triple (Σ, Q, Δ) (and sometimes just by the set Δ). The language L recognized by such a tiling system will be denoted $L = L(\Sigma, Q, \Delta)$.

We denote by REC the family of all picture languages that are recognizable. We speak of *recognizability by finite tiling systems.*

A tiling system (Σ, Q, Δ) represents a natural generalization to two dimensions of a finite automaton recognizing a string language. Indeed, when restricting to the one-dimensional case, the tiling system (Σ, Q, Δ) determines a nondeterministic finite automaton $\mathscr{A}$ over the input alphabet Σ as follows: The state set of $\mathscr{A}$ is $Q \dot{\cup} \{q_0\}$, where q_0 is the (new) initial state, and a state q is final if some tile $((a, q), \#)$ occurs in Δ. For $q \neq q_0$, (q, a, q') is a transition of $\mathscr{A}$ if some tile $((a', q), (a, q'))$ occurs in Δ, and (q_0, a, q') is a transition if $(\#, (a, q')) \in \Delta$.

The following figure illustrates this correspondence:

Run by $\mathscr{A}$: $q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_0 \xrightarrow{a} q_2$ (q_2 is a final state)

Tiling by Δ: $\#$ (a, q_1) (b, q_0) (a, q_2) $\#$

involving the tiles $(\#, (a, q_1))$, $((a, q_1), (b, q_0))$, $((b, q_0), (a, q_2))$, and $((a, q_2), \#)$.

Many closure properties hold for the family REC. We start with the simple remark that REC is closed under projection.

Remark 2.3. Let $L_1 \subseteq \Sigma_1^{**}$, $L_2 \subseteq \Sigma_2^{**}$ be picture languages such that $L_1 = \pi_1(L_2)$ where $\pi_1: \Sigma_2 \to \Sigma_1$. If L_2 is recognizable, then so is L_1.

Proof. If L_2 is recognizable, then there exists a local language $L_3 \subseteq \Sigma_3^{**}$ such that $L_2 = \pi_2(L_3)$ and $\pi_2: \Sigma_3 \to \Sigma_2$. Then, L_1 can be directly obtained as $L_1 = \pi(L_3)$ where $\pi = \pi_1 \circ \pi_2$, and hence L_1 is recognizable. ▮

The proofs of the following theorems can be found in [9].

THEOREM 2.4. *The family* REC *is closed under row and column concatenation, and under row and column Kleene closure.*

THEOREM 2.5. *The family* REC *is closed under Boolean union and intersection.*

In a different setup, it was proved by Inoue and Takanami (cf. [15, 12, 13]) that REC is not closed under Boolean complementation. Their proof refers to two other definitons of the recognizable picture languages, using "on-line tesselation automata" and "one-way parallel sequential array acceptors" and is quite complicated. We report here a short combinatorial proof of this fact given in [9], which refers directly to tiling systems.

(The argument can also be adjusted to recognizable sets of planar directed acyclic graphs in the sense of Bossut *et al.* [2], answering their question about closure of the class of these recognizable sets under complement in the negative.)

THEOREM 2.6. *The family* REC *is* not *closed under complement.*

Proof. Let Σ be an alphabet and let

$$L = \{p \in \Sigma^{**} \mid p = s \ominus s \text{ where } s \text{ is a square}\}.$$

In other words, L contains pictures of size $(2n, n)$ for every n such that the top and the bottom square halves are identical. We show that $L \notin \mathrm{REC}$ while the complement ${}^cL$ belongs to REC.

Suppose that $L \in \mathrm{REC}$, that is L is a projection of a local language L' over an alphabet Γ. A counting argument will show that this leads to a contradiction. Let σ and γ be the sizes of the alphabets Σ and Γ respectively. For an integer n let

$$L_n = \{ p \in \Sigma^{**} \mid p = s \ominus s \text{ where } s \text{ is a square of size } (n, n) \}.$$

The number of pictures in L_n is σ^{n^2}. Let L'_n be the set of rectangles in L' (over Γ) whose projections are in L_n. For the stripe rectangles over Γ of size $(2, n)$ consisting of the n-th and $(n+1)$-st rows in the rectangles of L'_n there are at most γ^{2n} possibilities.

For n sufficiently large, we have $\sigma^{n^2} > \gamma^{2n}$. Therefore, for n sufficiently large, there will be two different pictures $p = s_p \ominus s_p$, $q = s_q \ominus s_q \in L_n$ (with $s_p \neq s_q$) such that the corresponding pictures $p' = s'_p \ominus s''_p$, $q' = s'_q \ominus s''_q \in L'_n$ over the tiling alphabet Γ have the same stripes consisting of the n-th and $(n+1)$-st rows. This implies that, by definition of local language, also the pictures $v' = s'_p \ominus s''_q$ and $w' = s'_q \ominus s''_p$ belong to $L_n{}'$ and therefore the pictures $\pi(v') = s_p \ominus s_q$ and $\pi(w') = s_q \ominus s_p$ belong to L_n. This gives a contradiction.

We now prove that ${}^cL \in \mathrm{REC}$. We decompose ${}^cL$ as ${}^cL = L_1 \cup L_2$ where:

$L_1 = \{ p \in \Sigma^{**} \mid \ell_1(p) \neq 2\ell_2(p) \}$

$L_2 = \{ p \in \Sigma^{**} \mid \ell_1(p) = 2\ell_2(p)$ and the top and bottom halves are not equal$\}$.

It is quite easy to show that L_1 is recognizable, using a tiling system which, within a rectangle, builds up a line declining stepwise two squares by one, starting at the top left corner and missing the bottom right corner. On the other hand, L_2 can be written as:

$$L_2 = L_3 \cap (\Sigma^{**} \ominus (L_4 \cap (\Sigma^{**} \oplus L_5 \oplus \Sigma^{**})) \ominus \Sigma^{**}),$$

where

$L_3 = \{ p \in \Sigma^{**} \mid \ell_1(p) = 2\ell_2(p) \}$

$L_4 = \{ p \in \Sigma^{**} \mid \ell_1(p) = \ell_2(p) + 1 \}$

$L_5 = \{ p \in \Sigma^{**} \mid \ell_2(p) = 1$ and the top and the bottom symbols are different$\}$.

Language L_3 and L_4 can be recognized by techniques similar to the one for L_1 described above (using the opposite of the last condition for e.g. L_3). To see that L_5 is recognizable it suffices to observe that it is actually a recognizable string language. This shows that language $L_2 \in \mathrm{REC}$ and consequently that the whole ${}^cL$ is recognizable. ∎

The previous theorem shows that properties which are fundamental in the theory of recognizable string languages may fail for recognizable picture languages. Another of these properties is the decidability of the emptiness problem. The emptiness problem for recognizable picture languages is formulated as follows:

Given a tiling system (Σ, Q, Δ), is $L(\Sigma, Q, \Delta)$ nonempty?

Using a reduction from the halting problem for Turing machines, the following theorem can be proven (cf. [8]).

THEOREM 2.7. *The emptiness is undecidable for the family* REC.

2.3. *Logical Definability*

We will now give a definition of logical definability of pictures and picture languages. As mentioned above, we will identify a picture $p \in \Sigma^{**}$ with the structure

$$\underline{p} = (\mathrm{dom}(p), S_1, S_2, (P_a)_{a \in \Sigma}).$$

where $\mathrm{dom}(p) = \{1, ..., \ell_1(p)\} \times \{1, ..., \ell_2(p)\}$. Properties of pictures will be described by first-order and monadic second-order formulas, using first-order variables $x, y, z, x_1, x_2, ...,$ for points of $\mathrm{dom}(p)$, i.e. "positions", and monadic second-order variables $X, Y, Z, X_1, X_2, ...,$ for sets of positions.

Atomic formulas are of the form $x = y$, xS_iy (where $i \in \{1, 2\}$), $X(x)$, and $P_a(x)$, interpreted as equality between x and y, $(x, y) \in S_i$, $x \in X$, $x \in P_a$, respectively. *Formulas* are built up from atomic formulas by means of the Boolean connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ and the quantifiers $\exists, \forall$, applicable to first-order as well as to second-order variables. A formula without free variables is called a *sentence*.

If $\varphi(X_1, ..., X_n)$ is a formula with at most $X_1, ..., X_n$ occurring free in φ, p is a picture, and $Q_1, ..., Q_n$ are subsets of $\mathrm{dom}(p)$, we write

$$(\underline{p}, Q_1, ..., Q_n) \models \varphi(X_1, ..., X_n)$$

if p satisfies φ under the above mentioned interpretation, where Q_i is taken as interpretation of X_i. If φ is a sentence, we write $\underline{p} \models \varphi$.

The language $L(\varphi)$ defined by a sentence φ is the set of all pictures $p \in \Sigma^{**}$ such that $\underline{p} \models \varphi$.

DEFINITION 2.8. A picture language L is *monadic second-order definable* ($L \in \mathrm{MSO}$), if there is a monadic second-order sentence φ with $L = L(\varphi)$.

L is *first-order definable* ($L \in \mathrm{FO}$) if there is a sentence φ containing only first-order quantifiers (i.e., ranging over positions only) such that $L = L(\varphi)$.

Finally L is *existential monadic second-order definable* ($L \in \mathrm{EMSO}$), if there is a sentence of the form

$$\varphi = \exists X_1 \cdots \exists X_n \psi(X_1, ..., X_n)$$

where ψ contains only first-order quantifiers, such that $L = L(\varphi)$.

Let us mention some properties of positions and pictures which are easily described by first-order formulas. An upper border position x of a picture, i.e. a position $x = (1, j)$ for some j, is described by $\varphi_{top}(x) := \neg\exists y\ yS_1 x$. Similarly, the other borders can be described by corresponding formulas $\varphi_{\mathrm{left}}(x)$, $\varphi_{\mathrm{right}}(x)$, $\varphi_{\mathrm{bottom}}(x)$. The four corner positions (top left, top right, bottom left, bottom right) are defined by appropriate conjunctions of these formulas (indicated here by $\varphi_{\mathrm{tl}}(x)$, $\varphi_{\mathrm{tr}}(x)$, $\varphi_{\mathrm{bl}}(x)$, $\varphi_{\mathrm{br}}(x)$, respectively).

As a simple application, the set of words, considered as the set of pictures of size $(1, n)$ where $n \in \mathbb{N}$, is definable by the first-order sentence $\forall x \varphi_{\mathrm{top}}(x)$.

Let us verify that the set of squares over a given alphabet is monadic second-order definable. It suffices to postulate a set of positions which (1) contains the left upper corner, (2) is "closed under diagonal successors" (i.e. passing from (i, j) to $(i+1, j+1)$), and (3) does not hit the bottom or right border, excepting the bottom right corner.

An existential monadic second-order sentence expressing this is the following:

$$\exists X(\exists x(\varphi_{\mathrm{tl}}(x) \wedge X(x))$$
$$\wedge \forall x \forall y \forall z (X(x) \wedge xS_1 y \wedge yS_2 z) \rightarrow X(z))$$
$$\wedge \forall x((\varphi_{\mathrm{bottom}}(x) \vee \varphi_{\mathrm{right}}(x)) \rightarrow (\neg X(x) \vee \varphi_{\mathrm{br}}(x)))$$

A universal monadic second-order sentence can also be taken instead. It states that any set of positions which satisfies items 1 and 2 above (and hence also the smallest such set, consisting just of the diagonal positions) has to contain the bottom right corner. Applying the results of the subsequent section one can show that no first-order sentence defines the set of squares.

3. EQUIVALENCE THEOREM

In this section we prove the main result of the present paper.

THEOREM 3.1. *For any picture language L: $L \in$ REC iff $L \in$ EMSO.*

First we shall show how to describe definability by a finite tiling system using an existential monadic second-order formula (Section 3.1). For the reverse direction, from existential monadic second-order logic to finite tiling systems, the notion of a locally threshold testable picture language will provide an intermediate stage (Section 3.2). Using this, in Sections 3.2–3.4 the converse of the Theorem is shown.

3.1. *From Tiling Systems to Existential Monadic Second-Order Logic*

PROPOSITION 3.2. *For any picture language L: If $L \in$ REC then $L \in$ EMSO.*

Proof. Let $L \subseteq \Sigma^{**}$ be recognizable, say defined by the finite tiling system (Σ, Γ, Δ), where $\Gamma = \Sigma \times Q$, and Δ is a set of (2, 2)–pictures over $\Gamma \cup \{\#\}$. We have

$p \in L$ iff there is a picture $c \in Q^{**}$
of the same size as p such that
$\widehat{p \times c}$ is tilable by Δ.

We have to formalize the right-hand side by an EMSO-formula to be interpreted in $\underline{p}$. Given $Q = \{q_1, ..., q_k\}$, we shall do this using set variables $X_1, ..., X_k$ where $X_l(x)$ is intended to mean $c(x) = q_l$. The above equivalence can then be reformulated as follows

$p \in L$ iff $\underline{p}$ satisfies the following condition:
$\exists X_1 \cdots \exists X_k(X_1, ..., X_k$ form a partition of $\mathrm{dom}(p)$
and for the picture c given by $c(i, j) = q_l$ iff $(i, j) \in X_l$
the picture $\widehat{p \times c}$ is tilable by Δ).

The partition condition on $X_1, ..., X_k$ is formalized by the following first-order formula:

$$\varphi_{\mathrm{partition}}(X_1, ..., X_k):$$
$$\forall z (X_1(z) \vee \cdots \vee X_k(z)) \wedge \bigwedge_{i \neq j} \neg(X_i(z) \wedge X_j(z)).$$

(If the empty picture is not in L, add the conjunction member $\exists x\ x = x$.) Next we have to express that each subpicture of size (2, 2) of $\widehat{p \times c}$ (where c is defined by a given partition $X_1, ..., X_k$ as above) belongs to Δ.

Each tile $\delta \in \Delta$ can be numerated in the form

$$\delta = \begin{array}{|c|c|} \hline \delta_1 & \delta_2 \\ \hline \delta_3 & \delta_4 \\ \hline \end{array}$$

also written as $(\delta_1, \delta_2, \delta_3, \delta_4)$, where $\delta_i \in (\Sigma \times Q) \cup \{\#\}$ for $i \in \{1, ..., 4\}$.

We divide Δ into nine disjoint sets:

$$\Delta = \Delta_{\mathrm{m}} \cup \Delta_{\mathrm{t}} \cup \Delta_{\mathrm{b}} \cup \Delta_{\mathrm{l}} \cup \Delta_{\mathrm{r}} \cup \Delta_{\mathrm{tl}} \cup \Delta_{\mathrm{tr}} \cup \Delta_{\mathrm{bl}} \cup \Delta_{\mathrm{br}}$$

where e.g. Δ_m contains all "middle tiles", i.e. those without $\#$, Δ_t contains the "top tiles", having the form

$$\begin{array}{|c|c|} \hline \# & \# \\ \hline \delta_3 & \delta_4 \\ \hline \end{array}$$

with $\delta_3, \delta_4 \in \Sigma \times Q$, and so on. By nine corresponding formulas $\psi_{\rm m}, \psi_{\rm t}, ..., \psi_{\rm br}$ we describe in each case which of the four positions $x_1, ..., x_4$ of a tile should match the picture (excluding the boundary #). (The formulas $\psi_{\rm m}, \psi_{\rm t}, ...,$ are variants of the formulas $\varphi_{\rm top}(x), \varphi_{\rm bottom}(x), ...$ in Section 2.3 above; in fact, for the corner positions one may take the previous formulas.) We set

$$\psi_{\rm m}(x_1, ..., x_4) := x_1 S_1 x_3 \wedge x_1 S_2 x_2 \wedge x_3 S_2 x_4 \wedge x_2 S_1 x_4$$

$$\psi_{\rm t}(x_3, x_4) := x_3 S_2 x_4 \wedge \neg\exists x\, x S_1 x_3 \wedge \neg\exists x\, x S_1 x_4$$

etc., up to

$$\psi_{\rm br}(x_1) := \neg\exists x\, x_1 S_1 x \wedge \neg\exists x\, x_1 S_2 x$$

Considering middle tiles for the moment, tilability of $p \times c$ means that for each quadruple (i, j), $(i+1, j)$, $(i, j+1)$, $(i+1, j+1)$ of positions of $\mathrm{dom}(p \times c)$, i.e. for each quadruple satisfying $\psi_{\rm m}(x_1, ..., x_4)$, the tile

$(p \times c)(i, j)$	$(p \times c)(i, j+1)$
$(p \times c)(i+1, j)$	$(p \times c)(i+1, j+1$

belongs to Δ_m. For the borders and corners of $\mathrm{dom}(p)$ an analogous tiling should be possible with respect to the appropriate subsets of Δ.

More formally, we use nine formulas $\chi_{\rm m}, \chi_{\rm t}, ..., \chi_{\rm br}$ expressing this for the nine possible cases. If for a tile δ we have $\delta_i = (a, q_l)$, where $i \in \{1, ..., 4\}$, let $\varphi_{\delta_i}(x)$ be an abbreviation for $P_a(x) \wedge X_l(x)$. Now let

$$\chi_{\rm m}\colon \psi_{\rm m}(x_1, ..., x_4) \rightarrow \bigvee_{(\delta_1, ..., \delta_4) \in \Delta_{\rm m}} (\varphi_{\delta_1}(x_1) \wedge \varphi_{\delta_2}(x_2) \wedge \varphi_{\delta_3}(x_3) \wedge \varphi_{\delta_4}(x_4))$$

$$\chi_{\rm t}\colon \psi_{\rm t}(x_3, x_4) \rightarrow \bigvee_{(\#, \#, \delta_3, \delta_4) \in \Delta_{\rm t}} (\varphi_{\delta_3}(x_3) \wedge \varphi_{\delta_4}(x_4))$$

etc., up to

$$\chi_{\rm br}\colon \psi_{\rm br}(x_1) \rightarrow \bigvee_{(\delta_1, \#, \#, \#) \in \Delta_{\rm br}} \varphi_{\delta_1}(x_1)$$

Thus we obtain as the desired existential monadic second-order sentence defining L:

$$\exists X_1 \cdots \exists X_k (\varphi_{\rm partition} \wedge \forall x_1 \cdots \forall x_4$$
$$(\chi_{\rm m} \wedge \chi_{\rm t} \wedge \chi_{\rm b} \wedge \chi_{\rm l} \wedge \chi_{\rm r} \wedge \chi_{\rm tl} \wedge \chi_{\rm tr} \wedge \chi_{\rm bl} \wedge \chi_{\rm br}))$$ ∎

In the remainder of the paper we show the reverse of Theorem 3.1, i.e., that an EMSO-definable picture language is a projection of a local set. We proceed as follows. Since EMSO-definable picture languages are projections of first-order definable sets (see proof of Proposition 3.4 below), it suffices to verify that first-order definable sets are projections of local sets. This will be done in three stages: First we show (in Sections 3.2 and 3.3) that the first-order definable picture languages coincide with the so-called *locally threshold testable picture languages* (generalizing local picture languages). Similarly as for the corresponding notion in string language theory (see e.g. Straubing's monograph [17]), membership of a picture in such a language is determined by the occurrence of subpictures from a certain finite set, subject to extra conditions on the number of these occurrences (which are counted up to a fixed threshold). The considered subpictures may be of any size (d, d) with $d \geqslant 2$. In Section 3.4, the proof of the main result will be finished by showing that locally threshold testable sets are projections of local sets. This will be verified by an elimination of the "occurrence constraints" in the definition of locally threshold testable sets (Lemma 3.9), and subsequently by a reduction from $(d \times d)$-tiles to (2×2)-tiles (Lemma 3.10).

3.2. *Locally Threshold Testable Picture Languages*

Given a "threshold number" $t \geqslant 1$ and a square dimension $d \geqslant 1$ we say that two pictures p_1 and p_2 are d, t-equivalent (short: $p_1 \sim_{d,t} p_2$) if for each square picture σ of dimension $\leqslant d$, there are either at least t occurrences of σ in both $\widehat{p_1}$ and $\widehat{p_2}$, or the numbers ($\leqslant t$) of occurrences of σ in $\widehat{p_1}$ and $\widehat{p_2}$ coincide. Note that this definition refers to pictures surrounded by the boundary symbol #. Formally, we define for any pictures p and σ and any natural number $t \geqslant 1$

$$\mathrm{occ}_\sigma^t(p) = \begin{cases} \text{the number of occurrences of } \sigma \text{ in } \hat{p} \\ \qquad \text{if } \sigma \text{ occurs less than } t \text{ times in } \hat{p} \\ t \qquad \text{otherwise} \end{cases}$$

Now we set $p_1 \sim_{d,t} p_2$ if for all squares σ of dimension at most d we have $\mathrm{occ}_\sigma^t(p_1) = \mathrm{occ}_\sigma^t(p_2)$. If we only refer to all squares of dimension d exactly, we write $p_1 \simeq_{d,t} p_2$.

The relations $\sim_{d,t}$ and $\simeq_{d,t}$ are equivalence relations. A picture language L is called *locally d-testable with threshold t* if L is a union of $\sim_{d,t}$-equivalence classes. If this holds for some t, we speak of *locally threshold d-testable* picture languages, and if for some d and t, L is locally d-testable with threshold t, we say L is *locally threshold testable*. Finally, if a picture language L is a union of $\simeq_{d,t}$-classes for some t, L is called *locally strictly threshold d-testable*.

The first step of the reverse direction of Theorem 3.1 is provided by the following theorem.

THEOREM 3.3. *A picture language is first-order definable iff it is locally threshold testable.*

Before giving the proof let us apply the theorem:

PROPOSITION 3.4. *If* $L \in$ EMSO *then L is a projection of a locally threshold testable picture language.*

Proof. Assume the picture language $L \subseteq \Sigma^{**}$ is defined by the existential monadic second-order sentence

$$\varphi\colon \exists X_1 \cdots \exists X_k \psi(X_1, ..., X_k)$$

where ψ is a first-order formula. The formula $\psi(X_1, ..., X_k)$ is satisfied in picture models of the form $(\underline{p}, Q_1, ..., Q_k)$ with $Q_i \subseteq \mathrm{dom}(p)$ for $i = 1, ..., k$. Such an expanded picture model corresponds to a picture over the extended alphabet $\Sigma \times \{0, 1\}^k$ where the m-th additional component is 1 at point (i, j) iff $(i, j) \in Q_m$ (otherwise 0). Let π be the canonical projection from $\Sigma \times \{0, 1\}^k$ to Σ. Then L is the projection under π of the picture language $L' \subseteq (\Sigma \times \{0, 1\}^k)^{**}$ defined by $\psi(X_1, ..., X_k)$ in the sense explained above. By Theorem 3.3, the picture language L' is locally threshold testable. ∎

The projections of locally threshold testable picture languages coincide with the picture languages recognizable by the graph acceptors in the sense of [18]. The subsequent sections will show that (over pictures) these graph acceptors can be reduced to tiling systems.

We now turn to the proof of Theorem 3.3. The direction from right to left is straightforward: Given a picture σ and a number t, it is easy to write down a first-order sentence which states that there are at least t (or at most t, or exactly t) occurrences of σ. Occurrences at border positions or corner positions are handled as in the above Proposition 3.2 on definability in EMSO.

To prove the reverse direction of Theorem 3.3 (the only one we need in this paper), we use some notions and results from model theory, more precisely, first-order model theory of graphs. Here we refer to a special set-up where all models under consideration are picture models $\underline{p}$.

Two picture models $\underline{p}$ and $\underline{q}$ are said to be *n–equivalent* (short: $\underline{p} \equiv_n \underline{q}$) if they satisfy the same first-order sentences of quantifier-depth $\leqslant n$ (i.e., sentences with up to n nested quantifiers). Clearly, $\equiv_n$ is an equivalence relation, and a picture language defined by a first-order sentence of quantifier-depth n is a union of $\equiv_n$-classes. For completing the proof of Theorem 3.3 it clearly suffices to show that for any $\equiv_n$-class C, the picture language $\{p \mid \underline{p} \in C\}$ is locally threshold testable. This, in turn, follows immediately from the following lemma.

LEMMA 3.5. *For all* $n \geqslant 0$ *there are* d, t, *namely* $d = 2 \cdot 3^n + 1$ *and* $t = n \cdot 3^{2n}$, *such that* $p \sim_{d,t} q$ *implies* $\underline{p} \equiv_n \underline{q}$.

The proof is given in the following section, using the technique of back- and forth-systems in the sense of Ehrenfeucht and Fraïssé. Since the details of the proof are not relevant later on, the reader may skip this section and still be able to follow the remainder of the paper.

3.3. *First-Order Definable Sets Are Locally Threshold Testable*

For the proof of Lemma 3.5, we use an algebraic characterization of the n-equivalence $\equiv_n$ between relational structures, due to Fraïssé, in the present context applied to picture models. (Referring to a game theoretic terminology introduced by Ehrenfeucht, one often speaks of "Ehrenfeucht–Fraïssé games".) For more background see e.g. [5].

We fix the alphabet Σ and consider the associated signature with equality, binary relation symbols S_1, S_2 for the two successor relations and unary relation symbols P_a for $a \in \Sigma$.

Given a picture model $\underline{p}$ and an r-tuple $\bar{u} = (u_1, ..., u_r)$ of positions from $\mathrm{dom}(p)$ and a formula $\varphi(\bar{x})$ in free variables $x_1, ..., x_r$ we write $(\underline{p}, \bar{u}) \models \varphi(\bar{x})$ if φ holds in $\underline{p}$ when interpreting x_i by u_i for $1 \leqslant i \leqslant r$. By definition, $(\underline{p}, \bar{u}) \equiv_n (\underline{q}, \bar{v})$ iff

$$(\underline{p}, \bar{u}) \models \varphi(\bar{x}) \Leftrightarrow (\underline{q}, \bar{v}) \models \varphi(\bar{x})$$

for all formulas $\varphi(\bar{x})$ of quantifier-depth $\leqslant n$ in the signature under consideration. For empty $\bar{u}$ and $\bar{v}$ we obtain the relation $\underline{p} \equiv_n \underline{q}$, meaning that $\underline{p}$ and $\underline{q}$ satisfy the same sentences of quantifier-depth at most n.

We now turn to the algebraic characterization of the n-equivalence $\equiv_n$. Let $\underline{p} = (\mathrm{dom}(p), S_1^p, S_2^p, (P_a^p)_{a \in \Sigma})$ and $\underline{q} = (\mathrm{dom}(q), S_1^q, S_2^q, (P_a^q)_{a \in \Sigma})$ be picture models. A finite relation $\{(u_1, v_1), ..., (u_r, v_r)\} \subseteq \mathrm{dom}(p) \times \mathrm{dom}(q)$ is denoted $\bar{u} \mapsto \bar{v}$. Such a relation is called a *partial isomorphism* if the assignment $u_i \mapsto v_i$ determines an injective (partial) function f from $\mathrm{dom}(p)$ to $\mathrm{dom}(q)$, whose domain consists of the elements in $\bar{u}$ and which preserves all relations under consideration, i.e. such that

$$u_j S_1^p u_k \Leftrightarrow v_j S_1^q v_k \qquad \text{and} \qquad u_j S_2^p u_k \Leftrightarrow v_j S_2^q v_k$$

and

$$u_j \in P_a^p \Leftrightarrow v_j \in P_a^q.$$

Now call $(\underline{p}, \bar{u})$ and $(\underline{q}, \bar{v})$ *n-isomorphic*, written: $(\underline{p}, \bar{u}) \cong_n (\underline{q}, \bar{v})$, if there are nonempty sets $I_0, ..., I_n$ of partial isomorphisms, each of them extending $\bar{u} \mapsto \bar{v}$, such that for all $k = 1, ..., n$ the following two properties hold:

(back property) $\forall f \in I_k\ \forall v \in \mathrm{dom}(q)\ \exists u \in \mathrm{dom}(p)$ such that

$$f \cup \{(u, v)\} \in I_{k-1},$$

(forth property) $\forall f \in I_k\ \forall u \in \mathrm{dom}(p)\ \exists v \in \mathrm{dom}(q)$ such that

$$f \cup \{(u, v)\} \in I_{k-1}.$$

The following fundamental result is due to Fraïssé (cf. [5], Chap. XII.2).

THEOREM 3.6. *For* $n \geqslant 0$: $(\underline{p}, \bar{u}) \equiv_n (\underline{q}, \bar{v})$ *iff* $(\underline{p}, \bar{u}) \cong_n (\underline{q}, \bar{v})$.

So in the proof of Lemma 3.5 it suffices to establish $\underline{p} \cong_n \underline{q}$ instead of $\underline{p} \equiv_n \underline{q}$.

For the proof it will be convenient to introduce a notion of "neighbourhood" within pictures, by which we mean subpictures of a given size and centered around given vertices: The *m-block around vertex* (i, j) *in the picture* p, denoted by m-block$((i, j), p)$, is the subpicture of p with designated center vertex (i, j) and with domain

$$\mathrm{dom}(p) \cap (\{i-m, ..., i+m\} \times \{j-m, ..., j+m\}).$$

(This definition is an adaption of the notion of "m-sphere", as introduced by Hanf [10], to the domain of picture models. Lemma 3.5 is a version of Hanf's "sphere lemma", cf. [18].)

For any $m \geqslant 0$ there are only finitely many possible isomorphism types of m-blocks, called *m-block types* below. Define, for a natural number t and an m-block type τ, the occurrence number $\mathrm{block}-\mathrm{occ}^t_\tau(p)$ analogously to $\mathrm{occ}^t_\sigma(p)$ above, namely to be the number of occurrences of τ in p if this number is $< t$, otherwise t.

An m-block is a (centered) picture of size at most $(2m+1, 2m+1)$. Hence the numbers $\mathrm{occ}^t_\sigma(p)$ for pictures σ of size $(2m+1, 2m+1)$ determine the numbers $\mathrm{block}-\mathrm{occ}^t_\tau(p)$ for m-block types τ.

Proof of Lemma 3.5. Assume $p \sim_{d,t} q$ with $d = 2 \cdot 3^n + 1$ and $t = n \cdot 3^{2n}$. By the preceding remark we obtain

$$\mathrm{block}-\mathrm{occ}^t_\tau(p) = \mathrm{block}-\mathrm{occ}^t_\tau(q) \qquad (+)$$

for all m-block types τ with $m \leqslant 3^n$.

In order to show $\underline{p} \equiv_n \underline{q}$ we verify $\underline{p} \cong_n \underline{q}$. We have to find suitable sets $I_0, ..., I_n$ of partial isomorphisms with the forth and back properties. The set I_k will contain functions of the form $f: (u_1, ..., u_{n-k}) \mapsto (v_1, ..., v_{n-k})$, i.e., functions with a domain of $n-k$ vertices (or less vertices if $u_i = u_j$ for certain $i \neq j$).

By definition, let $f: (u_1, ..., u_{n-k}) \mapsto (v_1, ..., v_{n-k})$ belong to I_k iff

$$\left(\bigcup_{i=1}^{n-k} 3^k\text{-block}(u_i, p), u_1, ..., u_{n-k}\right) \cong \left(\bigcup_{i=1}^{n-k} 3^k\text{-block}(v_i, q), v_1, ..., v_{n-k}\right), \qquad (*)_k$$

i.e., there is an isomorphism from the union of the 3^k-blocks around the u_i to the union of the 3^k-blocks around the v_i which maps u_i to v_i for $i = 1, ..., n-k$.

The set I_n is nonempty since the empty function belongs to it. A function f in I_0 is a partial isomorphism since the condition $(*)_k$ for $k=0$ implies that vertex labels as well as successor relations between vertices are preserved (note that for $k=0$ we are dealing with 1-blocks). A fortiori, the functions in I_k for $k = 1, ..., n$ are partial isomorphisms.

We now verify the forth property for $I_n, ..., I_0$ (the back property is checked analogously), and hence by induction also obtain that $I_{n-1}, ..., I_0$ are nonempty. Suppose that $f: (u_1, ..., u_{n-k}) \mapsto (v_1, ..., v_{n-k})$ belongs to I_k and $(*)_k$ holds. Let $u(= u_{n-(k-1)}) \in \mathrm{dom}(p)$ be given. We have to find $v(= v_{n-(k-1)}) \in \mathrm{dom}(q)$ such that

$$\left(\bigcup_{i=1}^{n-(k-1)} 3^{k-1}\text{-block}(u_i, p), u_1, ..., u_{n-(k-1)}\right) \cong \left(\bigcup_{i=1}^{n-(k-1)} 3^{k-1}\text{-block}(v_i, p), v_1, ..., v_{n-(k-1)}\right) (*)_{k-1}$$

Case 1. If $u \in \frac{2}{3} \cdot 3^k$-block$(u_i, p)$ for some u_i, we may choose v from the corresponding block $\frac{2}{3} \cdot 3^k$-block(v_i, q) accordingly; note that, by Case 1, 3^{k-1}-block(u, p) is contained in 3^k-block(u_i, p) and thus, by the isomorphism $(*)_k$, 3^{k-1}-block(v, q) is contained in 3^k-block(v_i, q). So $(*)_k$ supplies an isomorphism from 3^{k-1}-block(u, p) onto 3^{k-1}-block(v, q) which maps u to v, and hence $(*)_{k-1}$ is established.

Case 2. If Case 1 does not apply, 3^{k-1}-block(u, p), say of type τ, is disjoint from all 3^{k-1}-block(u_i, p). In order to establish $(*)_{k-1}$, we have to find a copy of τ in q which is disjoint from all blocks 3^{k-1}-block(v_i, q). By choice of u in Case 2 we know that there are more copies of τ in p than those with center in one of the sets $\frac{2}{3} \cdot 3^k$-block(u_i, p). It suffices to ensure that there are more copies of τ in q than those with center in one of the sets $\frac{2}{3} \cdot 3^k$-block(v_i, q). (Taking a copy whose center avoids these blocks will give us the desired disjointness to all 3^{k-1}-block(v_i, q).) Since there is, by $(*)_k$, the same number of copies of τ in the union of the sets $\frac{2}{3} \cdot 3^k$-block(u_i, p) and the union of the sets $\frac{2}{3} \cdot 3^k$-block-(v_i, q) (where $i = 1, ..., n-k$), we are done if

$$\mathrm{block}-\mathrm{occ}^{s+1}_\tau(p) = \mathrm{block}-\mathrm{occ}^{s+1}_\tau(q)$$

where s is the cardinality of either of the two unions. We can estimate this cardinality by

$$s \leqslant (n-k) \cdot (2 \cdot \tfrac{2}{3} \cdot 3^{k-1} + 1)^2 < n \cdot 3^{2n}.$$

Thus we have $s + 1 \leqslant t$ when t is as specified at the beginning of the proof, and the claim follows now from the equality $(+)$. ∎

3.4. *From Locally Threshold Testability to Tiling Systems*

To finish the proof of the main result, starting from Proposition 3.4, we shall simulate threshold counting of subblocks of size (d, d) by a tiling system with tiles of size

(d, d), and then reduce the tiles under consideration to size $(2, 2)$.

In the framework of tiling systems we are usually speaking about a fixed dimension d, as opposed to the notion of threshold counting where d is only an upper bound. Therefore we will decompose a locally threshold countable picture language first into a finite union of languages which are locally strictly threshold j-testable for $j \leqslant d$, i.e. describable by counting only subpictures of a fixed size j. Since REC is closed under union, it suffices to show that each of these sublanguages is recognizable.

We use the notation Σ_d^{**} for the set of all pictures of size (m, n) with $m \geqslant d$ and $n \geqslant d$. In particular, $\Sigma^{**} = \Sigma_0^{**}$.

LEMMA 3.7. *Each locally threshold d-testable language L can be decomposed into $L_0 \cup L_1 \cup \cdots \cup L_{d-2}$, where $L_i \subseteq \Sigma_i^{**}$ $(0 \leqslant i \leqslant d-2)$ is locally strictly threshold $(i+2)$-testable.*

Proof. L_0 is either the empty set or consists only of the empty picture (if that is in L), in which case it is definable by the condition that $\mathrm{occ}_\sigma^1 = 1$ for σ being the $(2, 2)$-block consisting only of $\#$.

L_i $(0 < i < d-2)$ contains the subset of all pictures in L of size (i, n) or (m, i) for any $n, m \geqslant i$. L_i is definable by referring only to subpictures of size $(i+2, i+2)$. These subpictures carry the border symbol $\#$ on top and bottom border or on left and right border, which ensures that only pictures of the appropriate size are in L_i. The counting conditions for the $(i+2, i+2)$-subpictures are obtained from the original description of L by taking into account the combinatorial possibilities of composing smaller subpictures.

L_{d-2} finally contains all pictures from L of size (m, n) where $m, n \geqslant d-2$, and it is definable by subpictures of size (d, d) exactly. It is defined again by taking into account the combinatorial possibilities of composing smaller pictures. In contrast to the previous case we do not postulate the occurrence of the border symbol, since we collect in L_{d-2} also pictures of arbitrary large size. ∎

DEFINITION 3.8. Let $d \geqslant 2$ be a positive integer. A picture language $L \subseteq \Sigma_{d-2}^{**}$ is *d-local* if there exists a set $\Delta^{(d)}$ of pictures of size (d, d) (or "d-tiles") over $\Sigma \cup \{\#\}$, such that $L = \{p \in \Sigma^{**} \mid T_{d,d}(\hat{p}) \subseteq \Delta^{(d)}\}$.

In other words, if a rectangle belongs to a *d-local* picture language then it can be recognized by looking at its subpictures of size (d, d).

Now we will reduce locally threshold testable picture languages to d-local languages. We may exclude the empty picture (L_0 from Lemma 3.7) which makes up an already local language in itself.

Using the decomposition of Lemma 3.7, it remains to deal with any of the languages $L_i (i > 0)$ defined therein which are all of the form assumed in the following lemma.

LEMMA 3.9. *Let $d \geqslant 3$ be a positive integer. A locally strictly threshold d-testable picture language $L \subseteq \Sigma_{d-2}^{**}$ is the projection of a d-local language.*

Proof. Let L be locally strictly threshold d-testable, in particular a union of $\simeq_{d,t}$-equivalence classes for some threshold t, let $k = (|\Sigma \cup \{\#\}|)^{d^2}$, and let $\sigma_1, \sigma_2, ..., \sigma_k$ be the different d-tiles over $\Sigma \cup \{\#\}$.

Then every $\simeq_{d,t}$-class can be represented by a k-tuple $(t_1, t_2, ..., t_k)$ of natural numbers, where $0 \leqslant t_i \leqslant t$ and t_i gives the number of occurrences of σ_i, counted up to the threshold t, in the pictures of that class. We can think of the language L as given by a set of *accepting k-tuples* corresponding to the $\simeq_{d,t}$-classes contained in L.

We will describe a procedure to define a d-local language L' over an appropriate alphabet $\Sigma \times \Gamma$ whose projection to Σ is the language L.

We start by giving the idea of the construction. When we are given a picture p, we can decide whether p belongs to L as follows. We perform a scanning of p using a square window of size (d, d) and count (up to the threshold t) how many times we see, through the window, each of all σ_i's $(i = 1, ..., k)$.

The scanning is carried out along all horizontal stripes of height d and explores them independently from left to right. By an alphabet expansion, which introduces an additional component within the labels of tiles, the intermediate results of the threshold counting are recorded. By scanning the rightmost tiles from top to bottom, the threshold counting over the whole picture is completed. The given picture will belong to the language L if and only if this procedure leads to an accepting k-tuple at the bottom-right corner.

We now formally define the d-local language L' by presenting the set $\Delta^{(d)}$ of d-tiles. Let $\mathscr{C}$ be the set of all k-tuples $(t_1, t_2, ..., t_k)$, $0 \leqslant t_i \leqslant t$ for $i = 1, ..., k$. The d-local language L' will be defined over the alphabet $\Sigma \times \mathscr{C}$, and the generic element of $\Delta^{(d)}$ will be of the form $\sigma \times c$, where $\sigma \in (\Sigma \cup \{\#\})^{d \times d}$, and $c \in (\mathscr{C} \cup \{\#\})^{d \times d}$. (Strictly speaking, we use $(\#, \#)$ as the present border symbol and postulate that σ and c have the same occurrences of $\#$, i.e. $\sigma(i, j) = \# \Leftrightarrow c(i, j) = \#$.) We define the d-tiles in a way that their "local compatibility" corresponds to the scanning procedure we described above.

We start defining the *middle*-, *top*-, and *bottom*-tiles, i.e. those without the $\#$ symbol, respectively having $\#$ only in the top row, or having $\#$ only in their bottom row.

> Let such a tile be of the form $\sigma_i \times c$. We require the following condition: if $c(d-1, d-2) = (r_1, ..., r_i, ..., r_k)$ then $c(d-1, d-1) = (r_1, ..., r_i + 1, ..., r_k)$, where the sum is done up to the threshold t. $(*)$

Note that we have chosen position $(d-1, d-1)$ to keep track of a counting from left to right in each horizontal stripe of height d. Since we have assumed $d \geqslant 3$, this position contains a symbol different from $\#$. Also note that a tiling of a picture of height m by d-tiles has $m+3-d$ rows of tiles, which is $\leqslant m$ for $d \geqslant 3$. Consequently there is for each row of tiles an own row of positions in the picture where the tiles are counted.

The above condition $(*)$ takes tracks of the updating of the k-tuple during the scanning: when we move the window one position to the right, we see in position $(d-1, d-2)$ the last result of counting, and we add 1 to the component correponding to the σ_i that we are observing through the window. And we write the new k-tuple in position $(d-1, d-1)$.

- The *left*-, *top-left*-, and *bottom-left*-tiles correspond to the initial conditions of the countings in the scanning procedure. That is for such a tile $\sigma_i \times c$ the condition $c(d-1, d-1) = (0, ..., 0, 1, 0, ..., 0)$ holds, where the 1 is on position i.

- The *right*-tiles contain the "partial totals", i.e. the result of counting on the horizontal stripes, in $c(d-1, d-2)$. We simulate the computation of the final sum in the $(d-1)$st column of the tiles (i.e. in the last non-$\#$ column) by imposing the following condition: $c(d-1, d-1) = c(d-2, d-1) + c(d-1, d-2) + (0, ..., 0, 1, 0, ..., 0)$ where the sums are done componentwise and up to the threshold t, and the 1 is in position i, counting the actual (right-)tile. Then $c(d-2, d-1)$ contains the total sum up to the previous row.

- The *top-right* tiles perform the initialization of the total sum with the partial sum of the first row, which is done by the same condition as for middle-tiles: if $c(d-1, d-2) = (r_1, ..., r_i, ..., r_k)$ then $c(d-1, d-1) = (r_1, ..., r_i + 1, ..., r_k)$, where σ_i is the actual top-right-tile.

Note that the above condition implies that bottom-rightmost elements of the pictures in L' contain the k-tuple representing the counting result (for each d-tile) over the entire picture.

- The *bottom-right* tiles first have to finish the formation of the sum

$$c(d-1, d-1) = c(d-2, d-1) + c(d-1, d-2) + (0, ..., 0, 1, 0, ..., 0),$$

performed by the same condition as for the right tiles. Additionally, they contain the "acceptance condition" in the bottom-rightmost non-$\#$ element. More precisely, we put the restriction that $c(d-1, d-1)$ can only be one of the *accepting* k-tuples for the language L.

The above construction of set $\Delta^{(d)}$ defines a language L' such that given $p \in \Sigma^{**}$, the existence of $p' \in L'$ such that $p = \pi(p')$ is equivalent to the existence of a scanning for p which performs the indicated counting procedure and arrives at an accepting k-tuple. ∎

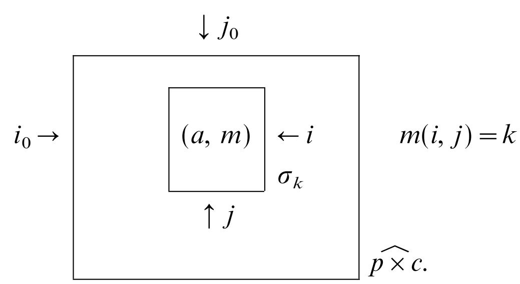
We can now finish the proof of Theorem 3.1 by the following lemma:

LEMMA 3.10. *A d-local picture language is a projection of a local language* (*and hence recognizable*).

Proof. Let $L \subseteq \Sigma_d^{**}$ be a d-local picture language and let $\Delta^{(d)} = \{\sigma_1, \sigma_2, ..., \sigma_h\}$ be the set of d-tiles over $\Sigma \cup \{\#\}$ that defines it. We will define a local language L' over a larger alphabet $\Gamma = \Sigma \times \mathscr{C}$ such that $\pi(L') = L$ by constructing explicitly its set $\Delta^{(2)}$ of 2-tiles.

The idea of the construction is to add for each letter of a picture p the information by which d-tiles it is covered in the original $\Delta^{(d)}$-tiling. Since every d-tile is a $d \times d$-matrix of letters, and a letter in a picture generally has to occur for each position $(i, j) \in (\{1, ..., d\})^2$ as $\sigma(i, j)$ for some $\sigma \in \Delta^{(d)}$, we represent this information in a $d \times d$-matrix $(m(i, j))$ of tile numbers.

So, given a position (i_0, j_0) of a picture p, a matrix $(m(i, j))_{i, j=1}^{d}$ will be appended to the value $a = p(i_0, j_0)$, and this will be the value of the corresponding position in the picture c, i.e. $(p \times c)(i_0, j_0) = (a, m)$. The entry $m(i, j)$ will be the index k of that d-tile in a tiling of $\hat{p}$ whose position (i, j) matches (i_0, j_0). For illustration see the following figure:



Clearly, for a matrix m appended to a, every tile $\sigma_{m(i, j)}$ has to carry letter a at position (i, j). We will fix this condition in the definition of the enlarged alphabet below.

Note that for $d > 2$ there will occur the case that a position (i_0, j_0) is near to the border of $\widehat{p \times c}$, so that e.g. $i > i_0 + 1$ for some $i \leqslant d$. In this case no d-tile can occur as subblock of p whose position (i, j) matches (i_0, j_0). We will denote this case by 0 in the matrix, and have to take care in the construction that 0 occurs at the correct positions, i.e. we should have

$$m(i, j) = 0 \Leftrightarrow i > i_0 + 1 \text{ or } i < d - i_0 - 1 \text{ or } j > j_0 + 1 \text{ or } j < d - j_0 - 1. \qquad (*)$$

Formally we define a new alphabet Γ as a subset of $\Sigma \times \{0, ..., h\}^{d \times d}$ by

$$(a, m) \in \Gamma \Leftrightarrow \text{for all } (i, j) \in (\{1, ..., d\})^2$$
$$(m(i, j) \neq 0 \Rightarrow \sigma_{m(i, j)}(i, j) = a)$$

and complete the proof by the definition of $\Delta^{(2)} \subseteq (\Gamma \cup \{\#\})^{2 \times 2}$, which is built up as

$$\Delta^{(2)} = \Delta_{\mathrm{m}}^{(2)} \dot\cup \Delta_{\mathrm{t}}^{(2)} \dot\cup \Delta_{\mathrm{b}}^{(2)} \dot\cup \Delta_{\mathrm{l}}^{(2)} \dot\cup \Delta_{\mathrm{r}}^{(2)} \dot\cup \Delta_{\mathrm{tl}}^{(2)} \dot\cup \Delta_{\mathrm{tr}}^{(2)} \dot\cup \Delta_{\mathrm{bl}}^{(2)} \dot\cup \Delta_{\mathrm{br}}^{(2)}.$$

Essentially, we have to check that for adjacent letters from Γ the elements of $\Delta^{(d)}$ coded in the matrices are compatible. That means for two matrices m and m' at positions (i_0, j_0) and (i_0, j_0+1), respectively, that they must contain the same tile numbers for those positions where the tiles of the d-tiling overlap both (i_0, j_0) and (i_0, j_0+1), i.e. $m(i, j) = m'(i, j+1)$ for $i = 1, ..., d$, $j = 1, ..., d-1$.

Moreover, we must guarantee that at the border of the 2-tiling in certain positions of the matrices numbers of appropriate border tiles of the original d-tiling occur, and that condition (*) holds.

We define the sets of tiles appropriate for the different corners, respectively borders and the middle, such that the tiling of a picture is controlled from top-left to bottom-right to obey the above conditions. Having this direction in mind, we have to assert only for the bottom-right position of each 2-tile that its matrix (i.e. component c_4) fulfills the mentioned conditions, since the other positions are covered by tiles checked "before". Additionally, we must check that we reach the right and bottom borders by the right tile.

The top-left position of p occurs in a tiling of p exactly as each of the four different positions (1, 1), (1, 2), (2, 1), (2, 2) of some d-tiles. The tile where it takes position (2, 2) must be a top-left tile (having only # on its other positions more to the top or left), and similarly, tiles for cases (1, 2) and (2, 1) have to be left- and top-d-tiles respectively. Formally, we have

$$\begin{array}{|c|c|}\hline \# & \# \\ \hline \# & (a_4, c_4) \\ \hline\end{array} \in \Delta_{\mathrm{tl}}^{(2)} \Leftrightarrow \begin{cases} c_4(i, j) \neq 0 \Leftrightarrow i \leqslant 2 \wedge j \leqslant 2, \\ \sigma_{c_4(2,2)} \in \Delta_{\mathrm{tl}}^{(d)}, \\ \sigma_{c_4(1,2)} \in \Delta_{\mathrm{l}}^{(d)}, \text{ and} \\ \sigma_{c_4(2,1)} \in \Delta_{\mathrm{t}}^{(d)}. \end{cases}$$

The matrix c_4 of a top-tile is mainly determined by the compatibility condition to c_3 (being c_4 of its left predecessor). Only the entries of the form $c_4(i, 1)$ are new. Among these, only $c_4(1, 1)$ and $c_4(2, 1)$ can be different from 0 (a position at the top of $p \times c$, i.e. the second row of $\widehat{p \times c}$, cannot occur in the third or lower row of any d-tile covering it). Moreover, the entries $c_4(1, 1)$ and $c_4(2, 1)$ must be 0 if and only if the position in the picture is close enough to the right border. This condition is fulfilled for the first time, when the previous tile, i.e. $\sigma_{c_4(2,2)} = \sigma_{c_3(2,1)}$ is a top-right tile. Afterwards the zeros are propagated (condition $c_4(2, 2) = 0$). Thus we define

$$\begin{array}{|c|c|}\hline \# & \# \\ \hline (a_3, c_3) & (a_4, c_4) \\ \hline\end{array} \in \Delta_{\mathrm{t}}^{(2)}$$

$$\Leftrightarrow \begin{cases} c_4(i, j) = c_3(i, j-1) \text{ for } i = 1, ..., d, j = 2, ..., d, \\ c_4(1, 1) = 0 \Leftrightarrow c_4(2, 1) = 0 \\ \qquad \Leftrightarrow (c_4(2, 2) = 0 \vee \sigma_{c_4(2,2)} \in \Delta_{\mathrm{tr}}^{(d)}), \\ c_4(2, 1) \neq 0 \Rightarrow \sigma_{c_4(2,1)} \in \Delta_{\mathrm{t}}^{(d)} \cup \Delta_{\mathrm{tr}}^{(d)}, \text{ and} \\ c_4(i, 1) = 0 \text{ for } i > 2. \end{cases}$$

$\Delta_{\mathrm{l}}^{(2)}$ is defined analogously to $\Delta_{\mathrm{t}}^{(2)}$, with c_2 instead of c_3 and roles of rows and columns interchanged.

Correspondingly, in a middle tile, compatibility to the top and to the left determines all entries except $c_4(1, 1)$. If $c_4(1, 2) = 0$ shows that the actual position is close to the right border (resp. $c_4(2, 1) = 0$ for the bottom border) the entry $c_4(1, 1)$ must be 0, too. Otherwise, it may be anything except 0.

$$\begin{array}{|c|c|}\hline (a_1, c_1) & (a_2, c_2) \\ \hline (a_3, c_3) & (a_4, c_4) \\ \hline\end{array} \in \Delta_{\mathrm{m}}^{(2)}$$

$$\Leftrightarrow \begin{cases} c_4(i, j) = c_2(i-1, j) \text{ for } i = 2, ..., d, j = 1, ..., d, \\ c_4(i, j) = c_3(i, j-1) \text{ for } i = 1, ..., d, j = 2, ..., d \text{ and} \\ c_4(1, 1) = 0 \Leftrightarrow (c_4(1, 2) = 0 \vee c_4(2, 1) = 0). \end{cases}$$

Since each position of $p \times c$ has to occur as right-bottom position of some tile in an accepting 2-tiling, it suffices to perform the considered compatibility test for this position, i.e. for the matrices c_4. However, the tests for the right border must be done within a tile where the position to be tested is the left-bottom one. Consequently, we have

$$\begin{array}{|c|c|}\hline (a_1, c_1) & \# \\ \hline (a_3, c_3) & \# \\ \hline\end{array} \in \Delta_{\mathrm{r}}^{(2)}$$

$$\Leftrightarrow \begin{cases} c_3(1, d) = 0 \Leftrightarrow c_3(1, d-1) = 0 \\ \quad \Leftrightarrow (c_3(d-1, d-1) = 0 \vee \sigma_{c_3(d-1, d-1)} \in \Delta_{\mathrm{br}}^{(d)}), \\ c_3(1, d-1) \neq 0 \Rightarrow \sigma_{c_3(1, d-1)} \in \Delta_{\mathrm{r}}^{(d)} \cup \Delta_{\mathrm{br}}^{(d)}, \text{ and} \\ c_3(1, j) = 0 \text{ for } j < d-1. \end{cases}$$

$\Delta_{\mathrm{b}}^{(2)}$ is defined analogously to $\Delta_{\mathrm{r}}^{(2)}$, whereas $\Delta_{\mathrm{tr}}^{(2)}$, $\Delta_{\mathrm{bl}}^{(2)}$, and $\Delta_{\mathrm{br}}^{(2)}$ are defined analogously to $\Delta_{\mathrm{tl}}^{(2)}$, with appropriate interchanges between rows and columns, as well as positions $(1, d)$, $(d, 1)$, and (d, d) respectively, instead of $(1, 1)$. ∎

4. CONCLUDING REMARKS AND OPEN PROBLEMS

In this paper we have proved that the family REC of picture languages recognized by finite tiling systems coincides with the family EMSO of picture languages definable in existential monadic second-order logic. This result indicates that the notion of recognizability presented in this paper is a robust one in the sense that at the same time it has a natural logic meaning and generalizes in a straightforward way the automaton recognizability for sets of strings.

In developing a theory of recognizability for picture languages a general task is to identify interesting subfamilies of recognizable picture languages and to state relationships between them. A first possible approach takes into account the closure operations of recognizable picture languages. The family REC contains all finite sets (cf. [9]) and is closed under row and column concatenation and under row and column Kleene closure (see Section 2.1 for the definitions), union and intersection (cf. Theorems 2.4 and 2.5).

However, REC is not closed under complementation (cf. Theorem 2.6). So one can introduce some regular-like expressions and define subfamilies of REC. In more specific way, given the set

$$\Theta = \{ \ominus, \oplus, *\ominus, *\oplus, \cup, \cap, {}^{c} \}$$

of operations and a subset $X \subseteq \Theta$, the family REG(X) is defined as the smallest family of picture languages containing all finite sets and closed under the operations of X. If X does not contain the complement, REG(X) is a subfamily of REC. The cases in which X contains the complement lead to some interesting open problems, concerning the inclusion of REG(X) in REC. Two cases play a special role: for $X = \Theta$, REG(Θ), denoted shortly by REG, defines the family of (generalized) *regular picture languages*; for $X = \{ \ominus, \oplus, \cup, \cap, {}^{c} \}$ REG(X) defines the family of *star-free picture languages*.

In the logical approach, one can introduce subfamilies of REC = EMSO by considering specific sublogics of existential monadic second-order logic. An interesting example is first-order logic where the two successor relations S_1 and S_2 are replaced by the corresponding orderings $\leqslant_1$ and $\leqslant_2$ (defined by $(i, j) \leqslant_1 (i', j)$ iff $i \leqslant i'$, resp. $(i, j) \leqslant_2 (i, j')$ iff $j \leqslant j'$). Concerning picture languages beyond the class EMSO, it is open whether the alternation of negation and existential set quantification in formulas (or equivalently: the alternation between projection and complement for picture languages) leads to an infinite hierarchy of classes of picture languages when starting with EMSO.

A third approach is to study and compare restricted versions of recognizability. Examples are recognizability by "nonambiguous tiling systems" and (possibly different versions of) "deterministic tiling systems". A tiling system is called *nonambiguous* if any tilable picture admits exactly one tiling. One may ask under which circumstances recognizability implies nonambiguous recognizability. We do not know, for example, if this is true for co-recognizable picture languages (i.e., for picture languages whose complement is also recognizable). A natural definition of *deterministic* tiling system requires that there is at most one tile for the top left corner of a picture, given this tile there is at most one continuation along the top row and along the leftmost column, and there is always at most one tile given its left and top neighbour. (These deterministic tiling systems are a version of the tessellation automata of [12]; other nonequivalent definitions require, for instancce, a unique tiling row by row or column by column.) Again, the question arises under which conditions we can infer that a recognizable picture language is deterministic in the above sense. Since there are picture languages which are unambiguous but not deterministic [16], these conditions should be more restrictive than those for nonambiguity.

Received March 29, 1994; final manuscript received October 2, 1995

REFERENCES

1. Blum, B., and Hewitt, C. (1967), Automata on a two-dimensional tape, *in* "IEEE Symposium on Switching and Automata Theory," pp. 155–160.
2. Bossut, F., Dauchet, M., and Warin, B. (1995), A Kleene Theorem for a class of planar acyclic graphs, *Inform. and Comput.* **117**, 251–265.
3. Courcelle, B. (1990), The monadic second-order theory of graphs. I. Recognizable sets of finite graphs, *Inform. and Comput.* **85**, 12-75.
4. Eilenberg, S. (1974), "Automata, Languages and Machines," Vol. A, Academic Press, New York.
5. Ebbinghaus, H. D., Flum, J., and Thomas, W. (1994), "Mathematical Logic" 2nd ed., Springer-Verlag/Berlin/Heidelberg/New York.
6. Fagin, R. (1975), Monadic generalized spectra, *Z. math. Logik Grundl. Math.* **21**, pp. 89–96.
7. Fagin, R., Stockmeyer, L., and Vardi, M. Y. (1993), On monadic NP vs. monadic co-NP, *in* "Proc. 8th IEEE Conf. on Structure in Complexity Theory," pp. 19–30.
8. Giammarresi, D., and Restivo, A. (1992), Recognizable picture languages, *in* "Proc. First International Colloquium on Parallel Image Processing, 1991," *Internat. J. Pattern Recognition and Artificial Intelligence*, **6**, Nos. 2 & 3, 241–256; also in the special issue "Parallel Image Processing" (M. Nivat, A. Saoudi, and P. S. P. Wang, Eds.), World Scientific, Series in Machine Perception and Artificial Intelligence, Singapore, pp. 31–46.
9. Giammarresi, D., and Restivo, A., Two-dimensional finite state recognizability. *Fund. Inform.*, to appear.
10. Hanf, W. P. (1965), Model-theoretic methods in the study of elementary logic, *in* "The Theory of Models" (J. W. Addison, L. Henkin, and A. Tarski, Eds.), pp. 132–145, North-Holland, Amsterdam.
11. Hopcroft, J. E., and Ullman, J. D. (1979), "Introduction to Automata Theory, Languages and Computation," Addison–Wesley, Reading, MA.
12. Inoue, K., and Nakamura, A. (1977), Some properties of two-dimensional on-line tessellation acceptors. *Inform. Sci.* **13**, 95–121.
13. Inoue, K., and Nakamura, A. (1977), Nonclosure properties of two-dimensional on-line tessellation acceptors and one-way parallel/sequential array acceptors. *Trans. of IECE of Japan* **6**, 475–476.

14. Inoue, I., and Takanami, I. (1990), A survey of two-dimensional automata theory, *in* "Proc. 5th Int. Meeting of Young Computer Scientists" (J. Dassow and J. Kelemen, Eds.), pp. 72–91, Lecture Notes in Computer Science, Vol. 381, Springer-Verlag, Berlin.
15. Inoue, K., and Takanami, I. (1992), A characterization of recognizable picture languages, *in* "Proc. Second International Colloquium on Parallel Image Processing" (A. Nakamura *et al*., Eds.), Lecture Notes in Computer Science, Vol. 654, Springer-Verlag, Berlin.
16. Potthoff, A., Seibert, S., and Thomas, W. (1994), Nondeterminism versus determinism of finite automata over directed acyclic graphs, *Bull. Belg. Math. Soc.* **1**, 285–298.
17. Straubing, H. (1994), "Finite Automata, Formal Logic, and Circuit Complexity," Birkhäuser, Boston.
18. Thomas, W. (1991), On logics, tilings, and automata, *in* "Proc. 18th Int. Colloquium on Automata, Languages and Programming," Lecture Notes in Computer Science, Vol. 510, pp. 441–453, Springer-Verlag, Berlin.