Simulations for a Class of Two-Dimensional Automata

Gérard Cécé, Alain Giorgetti

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Simulations for a Class of Two-Dimensional Automata

Gérard Cécé — Alain Giorgetti

N° 7425 — version 3

initial version October 2010 — revised version November 2012

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--7425--FR+ENG

Simulations for a Class of Two-Dimensional Automata

Gérard Cécé *†, Alain Giorgetti *‡

Thème : Programmation, vérification et preuves
Équipes-Projets CASSIS

Rapport de recherche n° 7425 — version 3 — initial version October 2010 — revised version November 2012 — 14 pages

Abstract: We study the notion of simulation over a class of automata which recognize 2D languages (languages of arrays of letters). This class of two-dimensional On-line Tessellation Automata (2OTA) accepts the same class of languages as the class of tiling systems, considered as the natural extension of classical regular word languages to the 2D case. We prove that simulation over 2OTA implies language inclusion. Even if the existence of a simulation relation between two 2OTA is shown to be an NP-complete problem in time, this is an important result since the inclusion problem is undecidable in general in this class of languages. Then we prove the existence in a given 2OTA of a unique maximal autosimulation relation, computable in polynomial time. ~~We also prove the existence of a unique minimal 2OTA which is simulation equivalent to a given 2OTA.~~ **(REVISION: Unfortunately, this last proof was false in the previous versions of this research report. Section 4.3 has consequently been revised.)**

Key-words: Simulation, Tiling, Picture languages, Picture automata

This project is supported by the ANR, the French national research agency (Agence Nationale de la Recherche), project ANR-06-ROBO-0009-03 and by the INRIA, CASSIS project.

* LIFC - EA 4269 - University of Franche-Comté - France
† Centre Numerica, 1 cours Leprince-Ringuet, BP 21126, 25201 MONTBELIARD Cedex
‡ INRIA Cassis Project, 16 route de Gray, 25030 BESANCON Cedex

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy
Téléphone : +33 3 83 59 30 00 — Télécopie : +33 3 83 27 83 19

Simulations pour une classe d'automates d'images

Résumé : Nous définissons une notion de simulation pour une classe d'automates qui reconnaissent des langages d'images (tableaux de lettres). Ces automates, connus sous le nom de "Two-dimensional On-line Tessellation Automata" (2OTA), acceptent la même classe de langages que les "tiling systems", qui sont considérés comme l'extension naturelle aux images de la classe des langages réguliers de mots. Nous démontrons que la simulation entre 2OTA implique l'inclusion entre langages reconnus. Quoique l'existence d'une relation de simulation entre deux 2OTA soit un problème NP-complet en temps, le résultat précédent est important car le problème général de l'inclusion dans cette classe de langages est indécidable. Nous démontrons ensuite l'existence dans un 2OTA donné d'une unique autosimulation maximale, calculable en temps polynomial. ~~Nous démontrons aussi l'existence d'un unique 2OTA minimal équivalent par simulation à un 2OTA donné.~~ **(RÉVISION: Malheureusement, cette dernière preuve d'existence n'était pas correcte dans les versions précédentes de ce rapport de recherche. La partie 4.3 a été révisée en conséquence.)**

Mots-clés : Simulation, pavage, langage d'images, automate d'images

Contents

1 Introduction

We are involved in the 'Smart Surface' project [sma10] whose aim is the realization of an active surface to automatically position and convey micro-items. This active surface is made of an array of smart micromodules. Under the abstraction that a micromodule can evolve only within a small number of states, we can consider those states as letters of a given alphabet. Then, what about representing a set of reachable configurations of the whole system as a recognizable two-dimensional (2D) language and using the regular model-checking (RMC) paradigm [ALdR06] on it? Let us recall that the RMC paradigm consists in representing infinite sets of configurations of a system by recognizable languages, and developing meta-transitions which can compute infinite sets of successors in one step.

To do this, we first need to clarify what could be recognizable 2D languages. The most accepted class is that recognized by tiling systems [GR97]. Unfortunately, an important property of classical regular languages is missing in this class, namely decidability of the inclusion problem, which is a necessary property in the RMC paradigm. This led us to seek sufficient conditions to decide inclusion. For words and trees, the existence of a simulation relation (in the sense of [Mil71]) between the underlying automata of two recognizable languages is such a sufficient condition. Moreover, the existence of an autosimulation relation, bigger than the identity, between the states of a finite automaton makes it possible to construct a smaller equivalent automaton by quotient. But tiling systems are not defined in terms of automata with straightforward notions of states and transitions. Fortunately, several kinds of automata recognizing the same class of 2D languages have been defined. Among them, there are two-dimensional On-line Tessellation Automata (2OTA) [IN77], Wang automata [LP10] and quadripolic automata [BG05]. This paper describes our results concerning simulations over 2OTA.[1]

Contributions We first define simulation relations between two 2OTA. We show that simulation implies language inclusion. From any given autosimulation relation – i.e. a simulation relation between the states of a given automaton – we construct a quotient automaton smaller than the given automaton, simulation equivalent to it (one simulates the other) and which therefore accepts the same language. In a 2OTA A we prove the existence of a unique maximal autosimulation relation. Then we show how to compute this maximal autosimulation in polynomial time. Unfortunately, we also show that the quotient automaton constructed from this maximal autosimulation relation is not always the smallest one which is simulation equivalent to A. We also prove that deciding the existence of a simulation relation between two different 2OTA is unfortunately NP-complete in time.

Related work The study of two-dimensional languages is an active field of research, see [CP09] for a recent overview. To our knowledge, this is the first work on simulations concerning 2D automata. In the last few years, several works have been done about simulations over tree automata [ALdR06, ABH$^+$08, ACH$^+$10] but mainly to reduce them. For example the complexity of the existence of an upward simulation between two different tree automata has not been investigated. Moreover, the search for a minimal automaton which is simulation equivalent to a given one has not been done for tree automata. For Kripke structures, and also for classical labelled transition systems, this search happened to be successful [BG03].

Outline The next section introduces pictures (two-dimensional arrays of letters), picture languages and tiling systems. Section 3 is dedicated to 2OTA and some of their properties. Then we define simulation relations over two 2OTA in Section 4, and give the first results of the paper: simulation implies language inclusion and there is a unique maximal autosimulation relation in a given 2OTA. We treat the algorithmic and complexity issues in Section 5. Section 6 is about backward simulations between 2OTA. We show that they do not imply language inclusion, contrarily to backward simulations between tree automata. Section 7 finally concludes the paper and suggests some future directions.

[1] "2OTA" indifferently abbreviates the plural "two-dimensional on-line tessellation automata" and the singular "two-dimensional on-line tessellation automaton".

2 Picture Languages and Tiling Systems

A *picture* is a two-dimensional array of letters from a given finite alphabet Σ. The set of all pictures over Σ is noted Σ^{**}. The *size* of a picture p is a couple of integers, $size(p) = (m, n)$, where m is the number of rows and n is the number of columns. By convention, we note ε the empty picture, whose size is $(0, 0)$. There is no picture of size $(0, k)$ or $(k, 0)$ with k positive. For a given picture p, we note $p_{i,j}$ the letter found at the intersection of the i^{th} line and the j^{th} column and we note $\hat{p}$ the picture which consists of p surrounded with a special symbol $\# \notin \Sigma$.

In the following example we show a square picture p of size $(5, 5)$ made of b but the main diagonal which is made of a. The corresponding $\hat{p}$, which size is $(7, 7)$, is also given.

$$p = \begin{array}{ccccc} a & b & b & b & b \\ b & a & b & b & b \\ b & b & a & b & b \\ b & b & b & a & b \\ b & b & b & b & a \end{array} \text{ and } \hat{p} = \begin{array}{ccccccc} \# & \# & \# & \# & \# & \# & \# \\ \# & a & b & b & b & b & \# \\ \# & b & a & b & b & b & \# \\ \# & b & b & a & b & b & \# \\ \# & b & b & b & a & b & \# \\ \# & b & b & b & b & a & \# \\ \# & \# & \# & \# & \# & \# & \# \end{array} \qquad (1)$$

A *picture language* on Σ is a subset of Σ^{**}. For a language $L \subseteq \Sigma^{**}$, we define $\hat{L} = \{\hat{p} \mid p \in L\}$. A *tiling system (TS)* is a tuple $T = (\Sigma, \Gamma, \Theta, \pi)$ such that Σ and Γ are finite alphabets, $\pi : \Gamma \to \Sigma$ is a mapping and Θ, the set of *tiles*, is a finite set of pictures of size $(2, 2)$ on the alphabet Γ. A language $L \subseteq \Sigma^{**}$ is said *recognized* by T if there exists a language $L' \subseteq \Gamma^{**}$ such that $L = \pi(L')$ and all sub-pictures of $\hat{L}'$ of size $(2, 2)$ belong to Θ. If the tile $\begin{array}{|c|c|}\hline \# & \# \\ \hline \# & \# \\ \hline\end{array}$ belongs to Θ we consider by convention that the empty picture ε belongs to L. Given a tiling system T, we note $\mathcal{L}(T)$ the language recognized by T. The family of languages recognized by tiling systems is noted $\mathcal{L}(TS)$ and is called the class of *recognizable picture languages*.

As an example, consider the tiling system $T = (\Sigma, \Gamma, \Theta, \pi)$ with: $\Sigma = \{a, b\}$, $\Gamma = \{0, 1, 2\}$, $\pi(0) = \pi(2) = b$, $\pi(1) = a$, and Θ the set of all the seventeen sub-pictures of size $(2, 2)$ of the following picture:

$$\begin{array}{ccccccc} \# & \# & \# & \# & \# & \# & \# \\ \# & 1 & 0 & 0 & 0 & 0 & \# \\ \# & 2 & 1 & 0 & 0 & 0 & \# \\ \# & 2 & 2 & 1 & 0 & 0 & \# \\ \# & 2 & 2 & 2 & 1 & 0 & \# \\ \# & 2 & 2 & 2 & 2 & 1 & \# \\ \# & \# & \# & \# & \# & \# & \# \end{array}$$

It is easy to show that the picture p in (1) belongs to $\mathcal{L}(T)$ and furthermore that $\mathcal{L}(T)$ is the set of all non empty square pictures whose main diagonal is made of a while the other positions are labeled by b.

3 Two-dimensional On-line Tessellation Automata

We consider 2OTA as an extension of classical finite automata from words to pictures. The intuition is as follows. In a finite automaton over words, a transition goes from one state to another state while reading a letter. In the 2D case, two directions have to be taken in account: downward and rightward. A transition in a 2OTA goes from two states to a third state while reading a letter, moving at the same time downward from the first state and rightward from the second state. In [IN77, GR97] a 2OTA is considered as a cellular automaton where cells change state in a synchronous way, diagonally across the array. This constraint is not necessary. Therefore, we relax it, and consider a run in a 2OTA like a run in a non-deterministic word automaton or tree automaton: a state is non deterministically associated to each position in the picture and we verify afterwards that this association satisfies the transition relation. Moreover, we do not force the set of initial states to be a singleton.

A (non-deterministic) *two-dimensional on-line tessellation automaton* (2OTA) is a tuple $A = (\Sigma, Q, I, F, \delta)$ where Σ is a finite alphabet, Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final, or accepting, states, and $\delta \subseteq Q^2 \times \Sigma \times Q$ is the transition relation. Given three states q_1, q_2, $q_3 \in Q$ and a letter $a \in \Sigma$, we can note more graphically the transition (q_1, q_2, a, q_3) by $\begin{array}{ccc} & q_1 & \\ q_2 & a & q_3 \end{array}$. This emphasizes the fact that q_1 and q_2 are respectively above q_3 and to the left of q_3.

Let $p \in \Sigma^{**}$ be a nonempty picture of size (m, n) over the alphabet Σ. A *run* of the automaton A on the picture p is a sequence of states $q_{i,j}$ for (i, j) in $\{0, \dots, m\} \times \{0, \dots, n\} \setminus \{(0, 0)\}$ such that there exists $q_0 \in I$ and for all valid i and j: $q_{i,0} = q_{0,j} = q_0$, $q_{m,n} \in F$ and $\begin{array}{ccc} & q_{i-1,j} & \\ q_{i,j-1} & p_{i,j} & q_{i,j} \end{array} \in \delta$. A run of the automaton A on the empty picture ε is a state q in $I \cap F$.

A two-dimensional on-line tessellation automaton A *accepts* a picture p if and only if there exists a run of A on p. The language *recognized by* A is the set $\mathcal{L}(A)$ of pictures accepted by A. The family of picture languages recognized by 2OTA is denoted $\mathcal{L}$(2OTA).

As an example (inspired from one in [GR97]), a 2OTA recognizing square pictures with a in the main diagonal and b elsewhere is $A = (\Sigma, Q, I, F, \delta)$ with $\Sigma = \{a, b\}$, $Q = \{0, 1, 2\}$, $I = \{0\}$, $F = \{2\}$ and $\delta = \left\{ \begin{array}{ccc} & 0 & \\ 0 & a & 2 \end{array}, \begin{array}{ccc} & 0 & \\ 2 & b & 1 \end{array}, \begin{array}{ccc} & 0 & \\ 1 & b & 1 \end{array}, \begin{array}{ccc} & 2 & \\ 0 & b & 1 \end{array}, \begin{array}{ccc} & 1 & \\ 1 & a & 2 \end{array}, \begin{array}{ccc} & 1 & \\ 2 & b & 1 \end{array}, \begin{array}{ccc} & 1 & \\ 1 & b & 1 \end{array}, \begin{array}{ccc} & 1 & \\ 0 & b & 1 \end{array}, \begin{array}{ccc} & 2 & \\ 1 & b & 1 \end{array} \right\}$. A visual way to represent a run of A on a picture p is to surround the letters in p with states such that the three surrounding states of a letter form with this letter a transition in δ. For instance:

$$\begin{array}{ccccc} & 0 & 0 & 0 & 0 \\ 0 & a_{2} & b_{1} & b_{1} & b_{1} \\ 0 & b_{1} & a_{2} & b_{1} & b_{1} \\ 0 & b_{1} & b_{1} & a_{2} & b_{1} \\ 0 & b_{1} & b_{1} & b_{1} & a_{2} \end{array} \text{ represents a run of } A \text{ on } p = \begin{array}{cccc} a & b & b & b \\ b & a & b & b \\ b & b & a & b \\ b & b & b & a \end{array}.$$

Let $A = (\Sigma, Q, I, F, \delta)$ be a 2OTA. We define the following sets of states: $\mathit{Reach}^0(A) = I$, $\mathit{Reach}^{n+1}(A) = \mathit{Reach}^n(A) \cup \{q \in Q \mid \exists q_1, q_2 \in \mathit{Reach}^n(A), \exists a \in \Sigma, (q_1, q_2, a, q) \in \delta\}$ for $n \geq 0$ and $\mathit{Reach}(A) = \bigcup_{n \geq 0} \mathit{Reach}^n(A)$. The set $\mathit{Reach}(A)$ describes the *reachable states* of A. A state in $\mathit{Reach}^n(A)$ and not in $\mathit{Reach}^{n-1}(A)$ when $n \geq 1$ is said at *distance* n of the initial set of states I. It is an undecidable problem to know whether a state is useful (whether it appears in a run of a recognized picture) or not, see the next proposition. However, the set $\mathit{Reach}(A)$ is easily computable as the sets I, Q and δ are finite. Furthermore, given a 2OTA, we can obviously restrict its set of states to its reachable states without changing its recognized picture language. We call the resulting automaton the *restriction* of the given 2OTA.

The following proposition summarizes some of the principal properties concerning 2OTA and tiling systems.

Proposition 1.

1. *The class of recognizable picture languages is closed under union, intersection and projection, but not under complement.*
2. $\mathcal{L}$*(2OTA)* $= \mathcal{L}(TS)$.
3. *From a 2OTA, a tiling system recognizing the same language is computable in polynomial time (and vice versa).*
4. *The membership problem for the language of some 2OTA is NP-complete.*
5. *The inclusion problem for recognizable picture languages is undecidable.*

6. *Knowing whether a given state belongs to a run of a given 2OTA is an undecidable problem.*

The proofs are given in [GR97] for (1), (2), and (3). In [LMN98] it is shown that the membership problem in $\mathcal{L}(TS)$ is NP-complete. With (2) and (3) we therefore have (4). In [GR97] it is shown that the universality problem (whether a picture language is indeed the set of all pictures) is undecidable in $\mathcal{L}(TS)$, we therefore deduce (5). In [GR97] it is shown that the emptiness problem (whether a picture language is empty) is undecidable in $\mathcal{L}(TS)$ and thus also in $\mathcal{L}(2OTA)$. Since it is easy to transform a 2OTA such that it has a single accepting state, we therefore deduce (6).

4 Simulations

The first motivation of this paper is to obtain a test of inclusion between the languages accepted by two 2OTA. This is done by the possible existence of a simulation between them. The following definition is therefore an extension of the definition of simulations from the case of classical finite word automata.

Definition 1. *Let $A = (\Sigma, Q, I, F, \delta)$ and $A' = (\Sigma, Q', I', F', \delta')$ be two 2OTA. A relation $S \subseteq Q \times Q'$ is a* simulation *over $A \times A'$, and A' is said to* simulate *A if:*

1. *for all $q \in I$ there exists $r \in I'$ such that $(q, r) \in S$,*
2. *for all $(q_1, r_1), (q_2, r_2) \in S$ and $(q_1, q_2, a, q_3) \in \delta$ there exists $r_3 \in Q'$ such that $(r_1, r_2, a, r_3) \in \delta'$ and $(q_3, r_3) \in S$, and*
3. *$(q, r) \in S$ and $q \in F$ imply $r \in F'$.*

For a simulation S we will occasionally note xSy for $(x, y) \in S$. A and A' are said simulation equivalent *if there exist a simulation over $A \times A'$ and a simulation over $A' \times A$.*

From this definition, we get the following expected theorem.

Theorem 1. *Let A, A' be two 2OTA and S a simulation over $A \times A'$. Then $\mathcal{L}(A) \subseteq \mathcal{L}(A')$.*

Proof. If A accepts the empty picture ε there is a state q in $I \cap F$. By Condition (1) there is a state r in I' such that $(q, r) \in S$. Then by Condition (3) the state r is also in F' and therefore A' also accepts the empty picture.

Now, let $(q_{i,j})_{(i,j)\in\{0,\ldots,m\}\times\{0,\ldots,n\}\setminus\{(0,0)\}}$ be a run of A on a nonempty picture p of size (m, n) $(m, n \geq 1)$. We define a run $(r_{i,j})_{(i,j)\in\{0,\ldots,m\}\times\{0,\ldots,n\}\setminus\{(0,0)\}}$ of A' on p.

Let $q_0 \in I$ be the initial state such that $q_{i,0} = q_{0,j} = q_0$ for all valid i and j. By (1) in Definition 1 there exists a state r_0 in I' such that $q_0 S r_0$. Let $r_{i,0} = r_{0,j} = r_0$ for all valid i and j.

The remaining part of the run of A' is defined by induction on $k = i + j$, from $k = 1$ to $k = m + n$. For $k = 1$, $r_{0,1}$ and $r_{1,0}$ have already been defined. They are the same initial state of A', $q_{0,1} S r_{0,1}$ and $q_{1,0} S r_{1,0}$. Let $k \geq 1$ be fixed and assume that all the states $r_{i,j}$ have been defined for $i + j \leq k$, such that $q_{i,j} S r_{i,j}$ and $\begin{smallmatrix} & r_{i-1,j} & \\ r_{i,j-1} & p_{i,j} & r_{i,j} \end{smallmatrix} \in \delta'$ whenever the indexes are valid. For any valid i and j such that $i + j = k + 1$, the transition $\begin{smallmatrix} & q_{i-1,j} & \\ q_{i,j-1} & p_{i,j} & q_{i,j} \end{smallmatrix}$ is in δ. Since $q_{i-1,j} S r_{i-1,j}$ and $q_{i,j-1} S r_{i,j-1}$ there exists by (2) in Definition 1 a state $r_{i,j}$ in Q' such that $\begin{smallmatrix} & r_{i-1,j} & \\ r_{i,j-1} & p_{i,j} & r_{i,j} \end{smallmatrix} \in \delta'$ and $q_{i,j} S r_{i,j}$. We choose this state to complete the family $(r_{i,j})_{(i,j)\in\{0,\ldots,m\}\times\{0,\ldots,n\}\setminus\{(0,0)\}}$. This family of states is by construction a run of A' on p. $\square$

4.1 Autosimulations

The second motivation of this study on simulations over 2OTA is to reduce them thanks to an autosimulation relation.

Definition 2. *Let $A = (\Sigma, Q, I, F, \delta)$ be a 2OTA. A relation $S \subseteq Q \times Q$ is a* simulation*, or more precisely an* autosimulation*, over A if:*

1. *S is reflexive,*
2. *for all $(q_1, r_1), (q_2, r_2) \in S$ and $(q_1, q_2, a, q_3) \in \delta$ there exists a state r_3 such that $(r_1, r_2, a, r_3) \in \delta$ and $(q_3, r_3) \in S$, and*
3. *$(q, r) \in S$ and $q \in F$ imply $r \in F$.*

From this definition, autosimulations and simulations over 2OTA are related as follows.

Proposition 2. *Let $A = (\Sigma, Q, I, F, \delta)$ be a 2OTA. A relation $S \subseteq Q \times Q$ is an autosimulation over A iff S is a reflexive simulation over $A \times A$.*

The finite set of autosimulations over a given 2OTA A is partially ordered by inclusion. It consequently admits maximal elements. The following lemma addresses the question of their uniqueness.

Theorem 2. *For any 2OTA A there exists a unique maximal autosimulation, denoted $\preccurlyeq_A$, over A. This maximal autosimulation is furthermore reflexive and transitive.*

Proof. Let S_1 and S_2 be two autosimulations over $A = (\Sigma, Q, I, F, \delta)$. We first show that the reflexive and transitive closure S^* of $S = S_1 \cup S_2$ is also an autosimulation. This is done by proving the property

$$P(n) := \forall n_1 \leq n, n_2 \leq n \left(\begin{array}{l} (q_1, q_2, a, q) \in \delta \wedge q_1 S^{n_1} r_1 \wedge q_2 S^{n_2} r_2 \\ \Rightarrow \exists r \ ((r_1, r_2, a, r) \in \delta \wedge q S^* r) \end{array} \right)$$

by induction on $n \geq 0$. The base case for $n = 0$ is straightforward since $S^0 \subseteq S^*$ is the identity. Therefore $P(0)$ is true. For the induction case, one of n_1 and n_2 can be assumed to be strictly greater than 0. Assume without loss of generality that it is n_1 and that $(q_1, q_2, a, q) \in \delta \wedge q_1 S^{n_1} r_1 \wedge q_2 S^{n_2} r_2$. Then there exists a state r'_1 such that $q_1 S^{n_1 - 1} r'_1$ and $r'_1 S r_1$. By the induction hypothesis, there exists a state r' such that $(r'_1, r_2, a, r') \in \delta \wedge q S^* r'$. The fact that $r'_1 S r_1$ implies that either $r'_1 S_1 r_1$ or $r'_1 S_2 r_1$. Assume (again without loss of generality) that it is $r'_1 S_1 r_1$. Since S_1 is reflexive, we also have $r_2 S_1 r_2$. From the fact that S_1 is an autosimulation, there exists a state r such that $(r_1, r_2, a, r) \in \delta \wedge r' S_1 r$. So we have $q S^* r'$ and $r' S_1 r$, which imply that $q S^* r$ and complete the inductive proof. The formula $P(n)$ is thus true for all n, implying Condition (2) of Definition 2. Condition (3) is also trivially true on S since S_1 and S_2 are autosimulations. From all of this, the relation S^* is a reflexive and transitive autosimulation which includes S_1 and S_2. Since identity is an autosimulation and from any two autosimulations we can construct a bigger reflexive and transitive autosimulation we get the unicity of the maximal autosimulation and its transitivity. □

We will henceforth only consider transitive autosimulations, i.e. autosimulations which are preorders.

4.2 Quotienting 2OTA

From an autosimulation S (that we assume transitive), we can define the equivalence relation (reflexive, symmetric and transitive) $S \cap S^{-1}$. We note $[q]_S$, or simply $[q]$ if S is obvious from the context, the class of the state q by the equivalence relation $S \cap S^{-1}$. We extend S on equivalence classes such that $([q], [r]) \in S$ iff $(q, r) \in S$.

Definition 3. *Let $A = (\Sigma, Q, I, F, \delta)$ be a 2OTA and S a (transitive) autosimulation over A. The* quotient automaton *$A_{/S} = (\Sigma, Q_{/S}, I_{/S}, F_{/S}, \delta_{/S})$, of A by S, is such that:*

1. $Q_{/S} = \{[q] \mid q \in Q\}$ *is the set of equivalent classes of* $S \cap S^{-1}$*,*
2. $I_{/S} = \{[q] \mid q \in I \wedge \forall r \in I, (qSr \Rightarrow rSq)\}$*,*
3. $F_{/S} = \{[q] \mid q \in F\}$*,*
4. $$\delta_{/S} = \left\{ \begin{smallmatrix} & [q_1] \\ [q_2] & a_{[q]} \end{smallmatrix} \;\middle|\; \begin{array}{l} \begin{smallmatrix} & q_1 \\ q_2 & a_{q} \end{smallmatrix} \in \delta \wedge \forall q_1', q_2', r \in Q, \\ \left([q_1'] = [q_1] \wedge [q_2'] = [q_2] \wedge qSr \wedge \begin{smallmatrix} & q_1' \\ q_2' & a_{r} \end{smallmatrix} \in \delta\right) \Rightarrow rSq \end{array} \right\}.$$

Our definition of the quotient automaton is not the classical one. The idea is to forget unnecessary transitions. In the case of a classical word automaton, this amounts at forgetting a transition $q' \xrightarrow{a} q$ if there already exists a transition $q' \xrightarrow{a} r$ with qSr. In this case, q is said to be a *little brother* of r. We have also adapted the initial set with the same idea: we keep only maximal initial states, maximality being defined with respect to the preorder S. From a 2OTA A and a simulation S over A the quotient $A_{/S}$ can be computed in polynomial time.

Lemma 1. *Let* $(\alpha_1, \alpha_2, a, \alpha_3) \in \delta_{/S}$ *be a transition in the quotient automaton. Then for all* $r_1 \in \alpha_1$*,* $r_2 \in \alpha_2$ *there exists* $r_3 \in \alpha_3$ *such that* $(r_1, r_2, a, r_3) \in \delta$*.*

Proof. By definition of $\delta_{/S}$ there exist three states q_1, q_2 and q_3 such that $[q_1] = \alpha_1$, $[q_2] = \alpha_2$, $[q] = \alpha_3$, $(q_1, q_2, a, q_3) \in \delta$ and

$$\forall q_1', q_2', r \in Q, ([q_1'] = [q_1] \wedge [q_2'] = [q_2] \wedge q_3Sr \wedge (q_1', q_2', a, r) \in \delta) \Rightarrow rSq_3. \tag{2}$$

Let $r_1 \in \alpha_1 = [q_1]$ and $r_2 \in \alpha_2 = [q_2]$. Then we have q_1Sr_1 and q_2Sr_2. From $(q_1, q_2, a, q_3) \in \delta$ and the definition of an autosimulation, there exists a state r_3 such that $(r_1, r_2, a, r_3) \in \delta$ and q_3Sr_3. Finally, applying (2) when q_1' is r_1, q_2' is r_2 and r is r_3 leads to r_3Sq_3. Consequently $r_3 \in \alpha_3$ completes the proof. □

Lemma 2. *Let* $(q_1, q_2, a, q_3) \in \delta$ *be a transition in* A*. Then there exists* q *such that* q_3Sq *and* $([q_1], [q_2], a, [q]) \in \delta_{/S}$*.*

Proof. Let

$$R = \left\{r \mid q_3Sr \wedge \exists r_1, r_2 \in Q, [r_1] = [q_1] \wedge [r_2] = [q_2] \wedge (r_1, r_2, a, r) \in \delta\right\}.$$

Let q be such a maximal element of R for the preorder S, i.e. an element of R such that $\forall q' \in R, qSq' \Rightarrow q'Sq$. The state q always exists since R is a subset of the finite set Q. By definition of R and q, we have q_3Sq. Let r_1 and r_2 be such that $[r_1] = [q_1]$, $[r_2] = [q_2]$ and $(r_1, r_2, a, q) \in \delta$. It remains to prove that $([q_1], [q_2], a, [q]) \in \delta_{/S}$.

Let q_1', q_2' and r be any three states in Q such that qSr and

$$[q_1'] = [r_1] \wedge [q_2'] = [r_2] \wedge (q_1', q_2', a, r) \in \delta. \tag{3}$$

By transitivity of S, q_3Sr holds and means together with (3) that r is an element of R. Consequently rSq. By definition of $\delta_{/S}$ it results from $(r_1, r_2, a, q) \in \delta$ that $([r_1], [r_2], a, [q])$ is in $\delta_{/S}$. The equalities $[r_1] = [q_1]$ and $[r_2] = [q_2]$ complete the proof. □

Theorem 3. *Let* $A = (\Sigma, Q, I, F, \delta)$ *be a 2OTA and* S *be a simulation over* A*. Then* A *and* $A_{/S}$ *are simulation equivalent.*

Proof. Let S' and S'' be the binary relations respectively defined over $A \times A_{/S}$ and $A_{/S} \times A$ by $S' = \{(q, [r]) \mid qSr\}$ and $S'' = \{([q], q) \mid q \in Q\}$. We separately prove that S' and S'' are simulations.

The following three conditions altogether prove that S' is a simulation.

1. Let $q \in I$ be an initial state of A. Let r be a maximal element of $J = \{r \mid r \in I \wedge qSr\}$, for the preorder S. Let $q' \in I$ be such that rSq'. By transitivity of S, we have qSq', thus $q' \in J$ and $q'Sr$ since r is maximal in J, which proves that $[r] \in I_{/S}$. Finally the state r is also such that $(q, [r]) \in S'$, thus S' satisfies Condition (1) in Definition (1).

2. Let q_1, q_2, q_3, r_1 and r_2 be states such that $(q_1, [r_1]) \in S'$, $(q_2, [r_2]) \in S'$ and $(q_1, q_2, a, q_3) \in \delta$.

   By Lemma 2 there exists $q \in Q$ such that q_3Sq and $([q_1], [q_2], a, [q]) \in \delta_{/S}$. Since q_3Sq the state $\alpha = [q]$ of $A_{/S}$ is such that $(q_3, \alpha) \in S'$ and $([q_1], [q_2], a, \alpha) \in \delta_{/S}$. Thus Condition (2) in Definition (1) is satisfied by S'.

3. For any final state $q \in F$ and any pair $(q, [r]) \in S'$, it results from qSr that $r \in F$ is also a final state. $[r] \in F_{/S}$ and Condition (3) in Definition (1) is satisfied by S'.

The following three conditions altogether prove that S'' is a simulation.

1. For any initial state $\alpha \in I_{/S}$ there is a state $q \in I$ such that $\alpha = [q]$, i.e. $(\alpha, q) \in S''$. Thus S'' satisfies Condition (1) in Definition (1).

2. Let α_1, α_2, α_3, r_1 and r_2 be states such that $(\alpha_1, r_1) \in S''$, $(\alpha_2, r_2) \in S''$ and $(\alpha_1, \alpha_2, a, \alpha_3) \in \delta_{/S}$. By Lemma 1 there exists $r_3 \in \alpha_3$ such that $(r_1, r_2, a, r_3) \in \delta$. Since $r_3 \in \alpha_3$ we have $(\alpha_3, r_3) \in S''$ and Condition (2) in Definition (1) is satisfied for S''.

3. For any final state $\alpha \in F_{/S}$ there is a state $q \in F$ such that $\alpha = [q]$. Then, for any state r in relation with α by S'', i.e. such that $\alpha = [r]$, $[r] = [q]$ and $r \in F$ since $q \in F$ and S is a simulation. Condition (3) in Definition (1) is satisfied for S''. $\square$

As an immediate consequence, doing such a quotient does not modify the recognized language.

Corollary 1. *Let $A = (\Sigma, Q, I, F, \delta)$ be a 2OTA and S be a simulation over A. Then A and $A_{/S}$ recognize the same picture language.*

4.3 Minimality and simulation equivalent 2OTA

We have proved the existence of a maximal autosimulation over any 2OTA A. Let $\preccurlyeq_A$ denote this maximal autosimulation over A. In the previous sections we have used this maximal autosimulation $\preccurlyeq_A$ to reduce a 2OTA A more than by doing a classical quotient. We could expect, as for Kripke structures [BG03], that the restriction of this reduced 2OTA $A_{/\preccurlyeq_A}$ to its reachable part is the smallest 2OTA which is simulation equivalent to the given 2OTA. Unfortunately it is not always the case, as shown by the following proposition.[2]

Proposition 3. *Let $A = (\Sigma, Q, I, F, \delta)$ be the 2OTA defined by $\Sigma = \{a\}$, $Q = \{q_0, q_0', q\}$, $I = \{q_0, q_0'\}$, $\delta = \{(q_0, q_0, a, q), (q_0', q_0', a, q)\}$ and $F = \{q\}$. Then the restriction of $A_{/\preccurlyeq_A}$ to its reachable part is not the smallest 2OTA which is simulation equivalent to A.*

Proof. In the transition relation δ, there is no transition of the form $(q_1, q_2, a, \_)$ with q_1 different from q_2, furthermore, q is the only final state of A. Consequently, there is no pair (q_1, q_2) with q_1 different from q_2 in any autosimulation over A. Thus, the maximal autosimulation $\preccurlyeq_A$ over A is the identity $\{(q_0, q_0), (q_0', q_0'), (q, q)\}$ over Q. Since all the states of A are reachable, the restriction of $A_{/\preccurlyeq_A}$ to its reachable part is A itself.

Let $A' = (\Sigma, Q', I', F', \delta')$ be the 2OTA defined by $Q' = \{r_0, r\}$, $I' = \{r_0\}$, $F' = \{r\}$, and $\delta' = \{(r_0, r_0, a, r)\}$. Let $S = \{(q_0, r_0), (q_0', r_0), (q, r)\}$ and $S' = \{(r_0, q_0), (r, q)\}$. Then, S is a simulation of A by A' and S' is a simulation of A' by A. Thus, A and A' are simulation equivalent, but A' is strictly smaller than the restriction of $A_{/\preccurlyeq_A}$ to its reachable part. $\square$

[2]This new Section 4.3 fixes an error in [CG11] and in the previous versions of this research report: their Theorem 4 and Corollary 2 were false.

5 How to Compute Simulations

A transition in a 2OTA resembles a transition in a tree automaton. Indeed, the transition (q_1, q_2, a, q) can be viewed as the rule $(q_1, q_2) \xrightarrow{a} q$ in a tree automaton. In [ABH+08] a polynomial time algorithm is given to compute what is called the maximal *upward simulation* in a tree automaton. In this section we reduce computation of maximal simulations in 2OTA to computation of maximal upward simulations in tree automata. Before that we shortly recall useful results about upward simulations in binary tree automata. In particular we do not define the semantics of tree automata which is not related to the present subject.

Definition 4. *A* binary Tree Automaton *(bTA) is a tuple $T = (\Sigma, Q, F, \delta)$ where Σ is a finite alphabet, Q is a finite set of states, $F \subseteq Q$ is a set of final states and $\delta \subseteq Q^2 \times \Sigma \times Q$ is a finite set of transitions.*

As usual, we can note $(q_1, q_2) \xrightarrow{a} q$ whenever $(q_1, q_2, a, q) \in \delta$. If we forget that 2OTA recognize pictures and bTA recognize binary trees, their definitions are similar, up to an extra set of initial states for 2OTA. This similarity is used in the remainder of this section. An *upward simulation* over a bTA $T = (\Sigma, Q, F, \delta)$ is a relation $S \subseteq Q \times Q$ such that $(q_1, q_2) \xrightarrow{a} q$ and $q_i S r_i$ for a given $i \in \{1, 2\}$ imply the existence of a state r such that qSr and $(r_1, q_2) \xrightarrow{a} r$ if $i = 1$ and $(q_1, r_2) \xrightarrow{a} r$ if $i = 2$. Note that the set of final states is not present in this definition. So let us call a *simulation without final states* (wfs-simulation) a relation $S \subseteq Q \times Q$ such that $(q_1, q_2) \xrightarrow{a} q$ and $q_i S r_i$ for all $i \in \{1, 2\}$ imply the existence of a state r such that $(r_1, r_2) \xrightarrow{a} r$ and qSr. We immediately get the following lemma.

Lemma 3. *Let T be a bTA and S a reflexive and transitive relation over T. Then S is a wfs-simulation over T iff it is an upward simulation over T.*

Proof. The fact that a reflexive wfs-simulation is also an upward simulation is obvious. Now let us consider a transitive upward simulation S over $T = (\Sigma, Q, F, \delta)$, a transition $(q_1, q_2) \xrightarrow{a} q$ in δ and two states $r_1, r_2 \in Q$ such that $q_i S r_i$ for all $i \in \{1, 2\}$. As S is an upward simulation there exists a state r' and a transition $(r_1, q_2) \xrightarrow{a} r'$ such that qSr'. By the same argument on this new transition there also exists a state r and a transition $(r_1, r_2) \xrightarrow{a} r$ such that $r'Sr$. By transitivity of S we get qSr, which concludes the proof. □

In [ALdR06], the existence and the uniqueness of a maximal upward simulation over a bTA T are shown. This maximal upward simulation, noted $\preccurlyeq_T$, is furthermore shown reflexive and transitive. As argued in [ABH+08], given a preorder R, it can still be shown that there is a unique maximal upward simulation included in R, noted $\preccurlyeq_T^R$, over a given bTA T. This relation $\preccurlyeq_T^R$ is reflexive and transitive. Still in [ABH+08], a polynomial time algorithm is given to compute $\preccurlyeq_T$. But indeed, a straightforward extension of the construction used in their proof (adding R as a constraint on the initial partition-relation pair) leads to the following stronger result.

Theorem 4. *Let $T = (\Sigma, Q, F, \delta)$ be a bTA and $R \subseteq Q \times Q$ be a reflexive and transitive relation (preorder). The maximal upward simulation $\preccurlyeq_T^R$ included in R is reflexive, transitive and computable in polynomial time.*

Corollary 2. *The maximal autosimulation over a 2OTA A is computable in polynomial time.*

Proof. Let $A = (\Sigma, Q, I, F, \delta)$ be a 2OTA. Then $T = (\Sigma, Q, F, \delta)$ is a bTA. Let $R \subseteq Q \times Q$ be the preorder such that qRr iff $(q \in F \Rightarrow r \in F)$. This preorder simply defines a partition of Q in two blocks: F and $Q \setminus F$, with $(Q \setminus F) \times F \subseteq R$. By Condition (3) of Definition 2 any autosimulation over A is included in R. This means that the maximal autosimulation $\preccurlyeq_A$ over A is also the maximal reflexive and transitive wfs-simulation over T included in R. By Lemma 3 it is also the maximal upward simulation $\preccurlyeq_T^R$ included in R, since $\preccurlyeq_T^R$ is reflexive and transitive. By Theorem 4 it is computable in polynomial time. □

Unfortunately, unlike in word automata, deciding the existence of a simulation between two 2OTA is not feasible in polynomial time.

Theorem 5. *Deciding whether a 2OTA is simulated by another 2OTA is an NP-complete problem.*

Proof. Let $A_1 = (\Sigma, Q_1, I_1, F_1, \delta_1)$ and $A_2 = (\Sigma, Q_2, I_2, F_2, \delta_2)$. First, this problem is in NP because we can non deterministically choose a relation in $Q_1 \times Q_2$ and then verify in polynomial time whether it satisfies the conditions of a simulation or not. To show the NP-completeness, we reduce the membership problem in 2OTA, known to be NP-complete (item (4) of Proposition 1), to our problem. For any picture p of size (m, n) we construct A_1 such that $p \in \mathcal{L}(A_2)$ iff A_1 is simulated by A_2.

Let $Q_1 = \{(0,0)\} \cup \{1,\dots,m\} \times \{1,\dots,n\}$, $I_1 = \{(0,0)\}$, $F_1 = \{(m,n)\}$ and

$$\delta_1 = \left\{ \begin{array}{cc} & {\scriptstyle (0,0)} \\ {\scriptstyle (0,0)} & p_{1,1\;(1,1)} \end{array} \right\} \cup \bigcup_{j \in \{2,\dots,m\}} \left\{ \begin{array}{cc} & {\scriptstyle (0,0)} \\ {\scriptstyle (1,j-1)} & p_{1,j\;(1,j)} \end{array} \right\} \cup \bigcup_{i \in \{2,\dots,n\}} \left\{ \begin{array}{cc} & {\scriptstyle (i-1,1)} \\ {\scriptstyle (0,0)} & p_{i,1\;(i,1)} \end{array} \right\} \cup \bigcup_{(i,j) \in \{2,\dots,m\} \times \{2,\dots,n\}} \left\{ \begin{array}{cc} & {\scriptstyle (i-1,j)} \\ {\scriptstyle (i,j-1)} & p_{i,j\;(i,j)} \end{array} \right\}.$$

By construction, we clearly have $\mathcal{L}(A_1) = \{p\}$. If A_1 is simulated by A_2, by Theorem 1, we have $\mathcal{L}(A_1) \subseteq \mathcal{L}(A_2)$, and thus $p \in \mathcal{L}(A_2)$. Conversely, suppose $p \in \mathcal{L}(A_2)$. By definition, there exists a run $(q_{i,j})_{(i,j) \in \{0,\dots,m\} \times \{0,\dots,n\} \setminus \{(0,0)\}}$ of A_2 on p. Consider the relation $S = \{((0,0), q_{1,0})\} \cup \bigcup_{(i,j) \in \{1,\dots,m\} \times \{1,\dots,n\}} \{((i,j), q_{ij})\}$. It is easy to show that S is a simulation relation over $A_1 \times A_2$. □

However, in 2OTA the inclusion problem is undecidable. In this perspective, the simulation test is a sufficient condition of inclusion which does not have a worst time complexity than the one of the membership problem.

6 The Case of Backward Simulations

In word automata or tree automata, there are two different simulations: a *forward simulation*, from initial states to final states, and a *backward simulation*, from final states to initial states. They both imply language inclusion. The simulations in 2OTA considered so far in this paper are forward simulations. This section establishes the noticeable fact that what could correspond to backward simulation in the case of 2OTA does not imply language inclusion.

Definition 5. *Let $A = (\Sigma, Q, \{q_0\}, F, \delta)$ and $A' = (\Sigma, Q', \{q'_0\}, F', \delta')$ be two 2OTA. A relation $S \subseteq Q \times Q'$ is a* backward simulation *over $A \times A'$ if:*

1. $(q_0, q'_0) \in S$,
2. *for all $(q_3, r_3) \in S$ and $(q_1, q_2, a, q_3) \in \delta$ there exist $r_1, r_2 \in Q'$ such that $(r_1, r_2, a, r_3) \in \delta'$, $(q_1, r_1) \in S$ and $(q_2, r_2) \in S$,*
3. *for all $q \in F$ there exists $r \in F'$ such that $(q, r) \in S$.*

In order to simplify the problem, initial sets are restricted to singletons (this is implicitly done in the case of tree automata where the initial state indeed corresponds to rules with an empty left hand side).

Proposition 4. *In 2OTA, backward simulation does not imply language inclusion.*

Proof. Consider the two automata $A = (\Sigma, Q, \{q_0\}, \{q_d\}, \delta)$ and $A' = (\Sigma, Q', \{q'_0\}, \{q'_d\}, \delta')$ such that $Q = \{q_0, q_a, q_b, q_c, q_d\}$, $\delta = \left\{ \begin{array}{cc} & {\scriptstyle q_0} \\ {\scriptstyle q_0} & a_{\,q_a} \end{array}, \begin{array}{cc} & {\scriptstyle q_0} \\ {\scriptstyle q_a} & b_{\,q_b} \end{array}, \begin{array}{cc} & {\scriptstyle q_a} \\ {\scriptstyle q_0} & c_{\,q_c} \end{array}, \begin{array}{cc} & {\scriptstyle q_b} \\ {\scriptstyle q_c} & d_{\,q_d} \end{array} \right\}$, $Q' = \{q'_0, q'_a, q''_a, q'_b, q'_c, q'_d\}$ and $\delta' = \left\{ \begin{array}{cc} & {\scriptstyle q'_0} \\ {\scriptstyle q'_0} & a_{\,q'_a} \end{array}, \begin{array}{cc} & {\scriptstyle q'_0} \\ {\scriptstyle q'_0} & a_{\,q''_a} \end{array}, \begin{array}{cc} & {\scriptstyle q'_0} \\ {\scriptstyle q''_a} & b_{\,q'_b} \end{array}, \begin{array}{cc} & {\scriptstyle q'_a} \\ {\scriptstyle q'_0} & c_{\,q'_c} \end{array}, \begin{array}{cc} & {\scriptstyle q'_b} \\ {\scriptstyle q'_c} & d_{\,q'_d} \end{array} \right\}$.

Clearly, we have $\mathcal{L}(A) = \left\{ \begin{array}{|c|c|} \hline a & b \\ \hline c & d \\ \hline \end{array} \right\}$. Now, consider the relation $S = \{(q_0, q'_0), (q_a, q'_a), (q_a, q''_a), (q_b, q'_b), (q_c, q'_c), (q_d, q'_d)\}$. The reader can easily check that S is a backward simulation over $A \times A'$ but that $\mathcal{L}(A') = \emptyset$, which concludes the proof. □

7 Conclusion and Future Work

In this paper, we have applied the notion of simulation to a class of 2D automata, namely the one of two dimensional on-line tessellation automata. We have obtained many desired results. A first result is a non trivial sufficient condition to decide the inclusion problem between the languages recognized by two 2OTA. This is an important result since this problem is undecidable in general. The fact established here that this sufficient condition is decidable in NP-complete time is the trade-off to pay. We also show how to reduce the size of a 2OTA into a smaller automaton which is equivalent to it by simulation. This is also an important result since, although decidable, the membership problem in all comparable classes of 2D automata is NP-complete in time, and reducing the size of a given automaton dramatically reduces the time to obtain a response to the membership question (less states have to be tried). We have shown that this reduction can fortunately be done in polynomial time.

Now that we have a first test for inclusion we can come back to our initial motivation: the extension of the regular model-checking paradigm to two dimensional languages. This will probably require to relax our definition of simulation and take into account the scanning strategy to recognize a picture [AGM09, LP10] and identify meta-transitions. This will be done in accordance with the examples we will treat.

The structure of 2OTA is similar to the one of binary tree automata. Indeed, both can be viewed as relational structures with two successors relations S_1 and S_2 [Tho03]. The difference lies in the fact that in 2OTA, we necessarily have that the composition of the two relations commutes, i.e. $S_1 \circ S_2 = S_2 \circ S_1$. With this point of view, our work can be seen as the extension of the notion of simulation from trees to tree structures with a constraint. Another noticeable difference between the two models is that in 2OTA what corresponds to downward simulation in trees does not imply language inclusion. The similarity with tree automata has allowed us to use the algorithm of [ABH$^+$08] for the computation of the maximal autosimulation in a 2OTA.

We have also begun to confront the present simulation notion to other devices accepting 2D recognizable languages such as Wang systems [dPV97], quadripolic automata [BG05] and even tiling systems.

References

[ABH$^+$08] Parosh Aziz Abdulla, Ahmed Bouajjani, Lukás Holík, Lisa Kaati, and Tomás Vojnar. Computing Simulations over Tree Automata. In C. R. Ramakrishnan and J. Rehof, editors, *TACAS*, volume 4963 of *LNCS*, pages 93–108. Springer, 2008.

[ACH$^+$10] Parosh Aziz Abdulla, Yu-Fang Chen, Lukás Holík, Richard Mayr, and Tomás Vojnar. When Simulation Meets Antichains. In J. Esparza and R. Majumdar, editors, *TACAS*, volume 6015 of *LNCS*, pages 158–174. Springer, 2010.

[AGM09] Marcella Anselmo, Dora Giammarresi, and Maria Madonia. A computational model for tiling recognizable two-dimensional languages. *Theor. Comput. Sci.*, 410(37):3520–3529, 2009.

[ALdR06] Parosh Aziz Abdulla, Axel Legay, Julien d'Orso, and Ahmed Rezine. Tree regular model checking: A simulation-based approach. *J. Log. Algebr. Program.*, 69(1-2):93–121, 2006.

[BG03] Doron Bustan and Orna Grumberg. Simulation-based minimization. *ACM Trans. Comput. Logic*, 4(2):181–206, 2003.

[BG05] Symeon Bozapalidis and Archontia Grammatikopoulou. Recognizable Picture Series. *Journal of Automata, Languages and Combinatorics*, 10(2/3):159–183, 2005.

[CG11] Gérard Cécé and Alain Giorgetti. Simulations over two-dimensional on-line tessellation automata. In Giancarlo Mauri and Alberto Leporati, editors, *Developments in Language Theory*, volume 6795 of *Lecture Notes in Computer Science*, pages 141–152. Springer, 2011.

[CP09] Alessandra Cherubini and Matteo Pradella. Picture Languages: From Wang Tiles to 2D Grammars. In S. Bozapalidis and G. Rahonis, editors, *CAI*, volume 5725 of *LNCS*, pages 13–46. Springer, 2009.

[dPV97] Lucio de Prophetis and Stefano Varricchio. Recognizability of Rectangular Pictures by Wang Systems. *Journal of Automata, Languages and Combinatorics*, 2(4):269–288, 1997.

[GR97] Dora Giammarresi and Antonio Restivo. Two-Dimensional Languages. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, pages 215–267. Springer, Berlin, 1997.

[IN77] Katsushi Inoue and Akira Nakamura. Some properties of two-dimensional on-line tessellation acceptors. *Inf. Sci.*, 13(2):95–121, 1977.

[LMN98] Kristian Lindgren, Cristopher Moore, and Mats Nordahl. Complexity of Two-Dimensional Patterns. *Journal of Statistical Physics*, 91(5-6):909–951, 1998.

[LP10] Violetta Lonati and Matteo Pradella. Deterministic recognizability of picture languages with Wang automata. *Discrete Math. & Theor. Comput. Sci.*, 12(4):73–94, 2010.

[Mil71] Robin Milner. An Algebraic Definition of Simulation Between Programs. In *Proc. of the 2nd Int. Joint Conf. on Artificial intelligence*, pages 481–489, San Francisco, CA, USA, 1971. Morgan Kaufmann Publishers Inc.

[sma10] The Smart Surface Project. http://www.smartsurface.cnrs.fr/, 2010.

[Tho03] Wolfgang Thomas. Uniform and nonuniform recognizability. *Theor. Comput. Sci.*, 292(1):299–316, 2003.

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399