Overlapping tile automata

David Janin

LaBRI, CNRS UMR 5800
Laboratoire Bordelais de Recherche en Informatique

Rapport de recherche RR-1465-12 (revised february 2013)

Overlapping tile automata

March 1, 2013

David Janin,
LaBRI, IPB, Université de Bordeaux

Overlapping tile automata

David Janin*

Université de Bordeaux, LaBRI UMR 5800,
351, cours de la Libération, F-33405 Talence, FRANCE
janin@labri.fr

Abstract. Premorphisms are monotonic mappings between partially ordered monoids where the morphism condition $\varphi(xy) = \varphi(x)\varphi(y)$ is relaxed into the condition $\varphi(xy) \leq \varphi(x)\varphi(y)$. Their use in place of morphisms has recently been advocated in situations where classical algebraic recognizability collapses. With languages of overlapping tiles, an extension of classical recognizability by morphisms, called quasi-recognizability, has already proved both its effectiveness and its power. In this paper, we complete the theory of such tile languages by providing a notion of (finite state) non deterministic tile automata that capture quasi-recognizability in the sense that quasi-recognizable languages correspond to finite boolean combinations of languages recognizable by finite state non deterministic tile automata. As a consequence, it is also shown that quasi-recognizable languages of tiles correspond to finite boolean combination of upward closed (in the natural order) languages of tiles definable in Monadic Second Order logic.

Introduction

Motivations and background.

There are many ways to describe one-dimensional overlapping tiles : the objects which are studied in this paper. Arising in inverse semigroup theory, they can be defined as (representations of) elements of a McAlister monoid [15], i.e. linear and unidirectional birooted trees. Then they are used in studies [11, 12] of the structure of tiling (in the usual sense with no overlaps) of the d-dimensional Euclidian space $\mathbb{R}^d$.

Overlapping tiles can also be seen as two way string objects extended by extra history recording capacities that prevent a new letter from being placed in a position where another distinct letter has already been positioned in the past.

For instance, starting from a given string object, say ab with two distinct letters a and b, one can remove letter b from the right of that object. The resulting object is denoted by abb^{-1}. If these objects are treated just as standard string objects, b^{-1} acts as the *group inverse* of b, and thus $bb^{-1} = 1$ henceforth $abb^{-1} = a$.

* partially funded by the project CONTINT 2012 - ANR 12 CORD 009 02 - INEDIT

If these string objects are treated as strings extended with recording capacity as mentioned above, then $abb^{-1} \neq a$. In that case, b^{-1} acts as the *pseudo-inverse* of b and bb^{-1} is seen rather as sort of a footprint of letter b that is kept on the right side of letter a.

Now if one adds that same letter b to the right of abb^{-1} again, this leads to build again the object $abb^{-1}b$ which, in both cases, equals the string object ab. Indeed, adding letter b to the right of its footprint bb^{-1} merely amounts to rebuilding b, i.e. we have $bb^{-1}b = b$.

On the contrary, if one tries to add letter a - distinct from b - to the right of abb^{-1} then the resulting object $abb^{-1}a$ equals 0: the undefined tile. Indeed, with standard string objects the resulting value would be aa but with overlapping tiles, no other letter than the original can be positioned on the right footprint bb^{-1} of b hence $bb^{-1}a = 0$ and thus $abb^{-1}a = 0$.

Surprisingly, this extension of the string data type turns out to be a mathematically well-founded and structurally rich extension of that type. Adding and removing letters to the right or to the left of an extended string induces an associative product: the underlying algebraic structure is the monoid of overlapping tiles. The history recording mechanism induces a partial order relation: the natural order defined on tiles.

Recent modeling experiments in computational music [1, 10], conducted with variants of overlapping tiles, further illustrate how the structural richness of the monoid of tiles can be used to great benefit. Indeed many derived operators, e.g. sequential or parallel compositions, can be defined from the inverse monoid structure.

There is thus a need to develop a language theory for overlapping tiles. Such a study has been initiated in [9].

Doing so, an immediate difficulty is that, as already observed for inverse monoids [16, 19], classical language theory tools are not directly applicable. Indeed, on birooted trees [19] or on positive overlapping tiles [9], the notion of recognizability defined via morphisms into finite monoids or, equivalently, defined by via classical finite state (two way) automata, collapses in terms of expressive power.

As a remedy, the use of premorphisms instead of morphisms has been successfully proposed [7]. Indeed, this variant of algebraic recognizability, called quasi-recognizability, is shown to *essentially* captures the expressive power of Monadic Second Order Logic (MSO) over tiles [9]. The purpose of the present paper is to extend and strengthen such an emerging algebraic theory by providing it with an automata theoretical counterpart.

Outline.

Overlapping tile automata are non deterministic finite state word automata with a semantics (acceptance condition) that is now defined in terms of overlapping tiles.

We first show that languages of tiles recognized by such finite state automata correspond to upward closed (in some natural order) languages definable in MSO. Then, we prove that they capture the notion of quasi-recognizable languages of tiles [7] in the sense that quasi-recognizable languages of tiles correspond to finite boolean combination of languages of tiles definable by finite state tile automata.

It must be mentioned that our former definition of recognizability by premorphisms was only defined for languages of positive tiles. This new automata theoretical approach induces a refined definition that, equivalent to our former proposal on positive tiles, is now applicable to languages of arbitrary positive and negative tiles.

The paper is organized as follows. Monoids of overlapping tiles and the related notion of tile automata are presented in Section 1. They are shown to capture upward closed (in the natural order) MSO definable languages of tiles (Theorem 2).

Special classes of premorphisms and partially ordered monoids, referred to as adequate, are defined and studied in Section 2. They provide the appropriate concepts for defining an effective notion of quasi-recognizability (Lemma 6).

Tile automata and quasi-recognizable languages are then related in Section 3. It is shown that quasi-recognizable languages of tiles exactly correspond to finite boolean combinations of languages recognizable by finite tile automata (Theorem 9) or, equivalently, finite boolean combinations of upward MSO definable languages of tiles (Corollary 10).

1 Overlapping tiles and their automata

Here we briefly give a description of the McAlister monoid [15] on the alphabet A, or, as presented and studied in [9], the monoid of one-dimensional overlapping tiles. Then we define and study the notion of overlapping tile automata.

1.1 Preliminaries

Let A be a finite alphabet A and let A^* be the free monoid generated by A with neutral denoted by 1. The concatenation of two words u and v is denoted by $u \cdot v$

or simply uv. The monoid A_0^* is defined as the extension of the free monoid A^* with a zero with $0 \cdot u = u \cdot 0 = 0$ for every $u \in A_0^*$.

Let $\leq_p$ (resp. $\leq_s$) be the prefix (resp. the suffix) order over A_0^*, that is, for every u and $v \in A_0^*$, $u \leq_p v$ (resp. $u \leq_s v$) when there exists $w \in A_0^*$ such that $uw = v$ (resp. $wu = v$). Observe that for every $u \in A_0^*$, we have $u \leq_p 0$ (resp. $v \leq_s 0$). Let then $\vee_p$ be the *(prefix) join* operator defined, for every u and $v \in A_0^*$, by $u \vee_p v = v$ when $u \leq_p v$, by $u \vee_p v = u$ when $v \leq_p u$ and by $u \vee_p v = 0$ otherwise. One can check that $\vee_p$ is indeed the join for the set A_0^* ordered by the prefix order. The *(suffix) join* operator $\vee_s$ is defined symmetrically.

Given $\bar{A}$ a disjoint copy of A, let $u \mapsto \bar{u}$ be the syntactic dual mapping defined by $\bar{1} = 1$ and, for every every letter $a \in A$ and every $u \in (A + \bar{A})^*$, by $\overline{ua} = \bar{u} \cdot \bar{a}$ and $\overline{u\bar{a}} = \bar{u} \cdot a$. The *free group* $FG(A)$ generated by A is defined as the quotient of $(A + \bar{A})^*$ by the least congruence $\simeq$ such that, for every letter $a \in A$, $a\bar{a} \simeq 1$ and $\bar{a}a \simeq 1$. As usual, every element $[u] \in FG(A)$ is represented by the unique word $v \in [u]$ that contains no factors of the form $a\bar{a}$ or $\bar{a}a$.

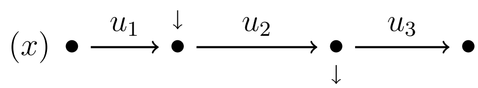
1.2 Overlapping tiles

A *one-dimensional overlapping tile*, or just *tile*, over the alphabet A is a triple of words $x = (u_1, u_2, u_3) \in A^* \times (A^* + \bar{A}^*) \times A^*$ such that, if $u_2 \in \bar{A}^*$, the syntactic inverse $\overline{u_2} \in A^*$ is a suffix of the word u_1 and a prefix of the word u_3.

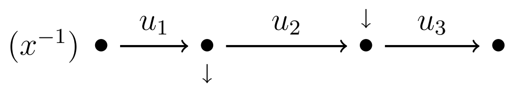
When $u_2 \in A^*$ we say that x is a *positive tile*. When $u_2 \in \bar{A}^*$ we say that x is a *negative tile*. When $u_2 = 1$, i.e. when x is both positive and negative, we say that x is a *context tile*. Sets T_A, T_A^+, T_A^- and C_A will respectively denote the set of tiles, the set of positive tiles, the set of negative tiles and the set of context tiles over A.

The *(syntactic) inverse* of a tile $x \in T_A$ is defined to be the tile $x^{-1} = (u_1 \cdot u_2, \bar{u}_2, u_2 \cdot u_3)$, with the product defined in the free group $FG(A)$. One can easily check that the induced mapping $x \mapsto x^{-1}$ is an involution that maps positive (resp. negative) tiles to negative (resp. positive) tiles and that, restricted to context tiles, is the identity mapping.

The *word domain* of a tile $x = (u_1, u_2, u_3)$ is the word $u_1 \cdot u_2 \cdot u_3 \in A^*$ with product performed in $FG(A)$. The *directed root* of the tile x is the word $u_2 \in A^* + \bar{A}^*$. A positive tile $x = (u_1, u_2, u_3)$ is conveniently drawn as a (linear, unidirectional and left to right) Munn's birooted word tree [17] with dangling arrows to identify the input vertex and the output vertex that marks the extremities of the directed root of x.

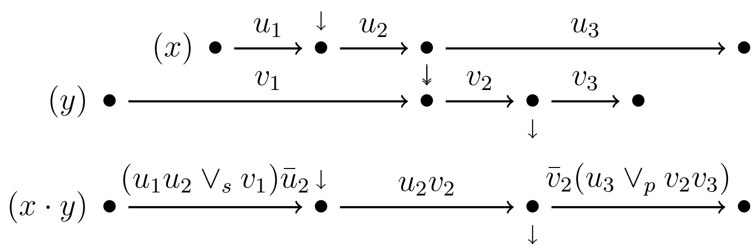


The negative tile $x^{-1} = (u_1u_2, \overline{u_2}, u_2u_3)$ is obtained from x by just inverting the input and the output vertices.



The product $x \cdot y$ of two tiles $x = (u_1, u_2, u_3)$ and $y = (v_1, v_2, v_3) \in T_A$ is defined in two steps: the output vertex of x is first positioned (or synchronized) with the input vertex of y, then, if possible, the word domains of x and y are merged letter by letter. The input vertex (resp. output vertex) of the resulting product $x \cdot y$ is then defined to be the input vertex of x (resp. the output vertex of y).

In the case $x = (u_1, u_2, u_3) \in T_A^+$ and $y = (v_1, v_2, v_3) \in T_A^+$, these two phases can be depicted as follows:



Formally, the (partial) *product* of two non-zero tiles $x = (u_1, u_2, u_3)$ and $y = (v_1, v_2, v_3)$ is defined as

$$x \cdot y = ((u_1u_2 \vee_s v_1)\overline{u_2}, u_2v_2, \overline{v_2}(u_3 \vee_p v_2v_3))$$

when both pattern matching conditions, $u_1u_2 \vee_s v_1 \neq 0$ and $u_3 \vee_p v_2v_3 \neq 0$, when evaluated in $FG(A)$, are satisfied.

Adding an undefined tile denoted by 0, this product is completed by $x \cdot y = 0$ when any of the pattern matching condition is not satisfied with $x \cdot 0 = 0 = 0 \cdot x$ for every $x \in T_A$.

For instance, with $A = \{a, b, c\}$ we have $(a, b, a) \cdot (b, a, c) = (a, ba, c)$ while $(a, b, a) \cdot (a, a, c) = 0$ since $a \neq b$. Another example is given by $(1, a, 1) \cdot (ba, c, ab) = (b, ac, ab)$. It illustrates the fact that the left and the right parts of the resulting product can arbitrarily come from any of the two product components.

As a special case of product, when $x = (u_1, u_2, 1)$ and $y = (1, v_2, v_2)$ the product $x \cdot y$ is always non zero. In that case, we say that the product $x \cdot y$ is a *disjoint product*.

The resulting set T_A^0 equipped with the above product is a *monoid* with unit $1 = (1, 1, 1)$ that is shown [9, 4] to be isomorphic to the McAlister monoid [15] generated by A. Extending the inverse mapping to 0 by taking $0^{-1} = 0$, for every tile $x \in T_A^0$, the tile x^{-1} is the unique tile such that both $xx^{-1}x = x$ and $x^{-1}xx^{-1} = x^{-1}$. The monoid T_A^0 is thus an *inverse* monoid [14].

As such, the idempotent elements are the elements of the form xx^{-1} (equivalently $x^{-1}x$) and the *natural order* associated with the inverse monoid T_A^0 is defined by $x \leq y$ when $x = xx^{-1}y$ (or, equivalently $x = yx^{-1}x$). One can easily check that for every non zero tile $x = (u_1, u_2, u_3)$ and $y = (v_1, v_2, v_3)$, we have $x \leq y$ if and only if $u_1 \geq_s v_1$, $u_2 = v_2$ and $u_3 \geq_p v_3$. One can also check that for every $x \in T_A^0$, $x \leq 1$ if and only if $x \cdot x = x$. In the sequel, idempotent tiles are thus also called *subunits*.

For every non zero tile $x = (u_1, u_2, u_3) \in T_A$, we define the *left projection* $x^L = x^{-1}x = (u_1u_2, 1, u_3)$ and the *right projection* $x^R = xx^{-1} = (u_1, 1, u_2u_3)$. One can check that $x^R \cdot x = x = x \cdot x^L$. Even more, the tile x^R (resp. x^L) is the least (in the natural order) idempotent tile $z \in T_A^0$ such that $z \cdot x = x$ (resp. $x = x \cdot z$). As an immediate consequence of the definition, for every x and $y \in T_A^0$, the product $x \cdot y$ is a disjoint product if and only if $x \cdot y \neq 0$ and 1 is the unique idempotent element that is both above x^L and y^R in the natural order. These notions of projections and disjoint products play a central role in the notion of adequately ordered monoid and adequate premorphism that are presented below.

Last, one can observe that the mapping $u \mapsto (1, u, 1)$ from A^* to T_A is a one-to-one morphism. In other words, the free monoid A^* is embedded into the McAslister monoid T_A^0 of overlapping tiles. In the remainder of the text we may use the same notation for words of A^* and their images in T_A^0.

1.3 Tile automata

Definition 1. A non deterministic (finite) *overlapping tile automaton* is a triple $\mathcal{A} = \langle Q, \delta, K \rangle$ with a (finite) set of states Q, a non deterministic transition function $\delta : A \to \mathcal{P}(Q \times Q)$ and an accepting set $K \subseteq Q \times Q$.

An run of the automaton $\mathcal{A}$ on a word $u = a_1 \cdots a_n \in A^*$ from state p to state q, which is denoted by $p \xrightarrow{u} q$, is a sequence of $n + 1$ states $q_0 = p$, q_1, q_2, $\ldots$, $q_n = q \in Q$ such that for every $1 \leq i \leq n$, we have $(q_{i-1}, q_i) \in \delta(a_i)$.

A run of the automaton $\mathcal{A}$ on a positive tile $x = (u, v, w) \in T_A^+$ (resp. a negative tile $x = (uv, \bar{v}, vw) \in T_A^-$) is quadruple of states $(s, p, q, e) \in Q \times Q \times Q \times Q$: a start state s, an input state p, an output state q and an end state e, such that $s \xrightarrow{u} p$, $p \xrightarrow{v} q$ and $q \xrightarrow{w} e$ (resp. $s \xrightarrow{u} q$, $q \xrightarrow{v} p$ and $p \xrightarrow{w} e$).

Such a run is an accepting run when $(p, q) \in K$. The set of tiles over which there is an accepting run of the automaton $\mathcal{A}$ is denoted by $L(\mathcal{A}) \subseteq T_A$. It is the language of tiles *recognized* by the automaton $\mathcal{A}$.

Examples. Let us consider the tile automaton (graph) defined with the set of states $Q = \{1, 2, 3\}$ and transitions $\delta(a) = \{(1, 1), (1, 2), (2, 3), (3, 3)\}$ for every $a \in A$. With $K^+ = \{(1, 2)\}$ or $K^+ = \{(2, 3)\}$ we recognize the language of all strictly positive tiles. With $K^- = \{(2, 1)\}$ or $K^- = \{(3, 2)\}$ we recognize the language of all strictly negative tiles. With $K^0 = \{(2, 2)\}$ we recognize the language of all idempotent tiles.

As an immediate consequence of tile automata semantics, every language of tiles recognized by a tile automaton is upward closed in the natural order. The following theorem shows that this is actually the characteristic property of these languages.

Theorem 2. *For every language of tiles $L \subseteq T_A$, the language L is recognizable by a finite state overlapping tile automaton if and only if the language L is upward closed (in the natural order) and definable in monadic second order logic.*

Proof. This essentially follows from the characterization of MSO definable languages that is provided by Theorem 4 in [9]. □

As far as complexity issues are concerned, one can easily check that, as for a non deterministic word automaton, deciding if a tile $x \in T_A$ belongs to the language $L(\mathcal{A})$ for some finite state automaton can be done in time $2n \cdot 2^p$ with n the size of x and p the number of states in $\mathcal{A}$. Similarly, the language emptyness problem can be solved in linear time in the size of automaton $\mathcal{A}$. Indeed, this just amounts to check that there is a path in the automaton graph from a state p to a state q with $(p, q) \in K$.

2 Quasi-recognizable languages of tiles

We define in this section a notion of quasi-recognizability extending the one proposed in [7]. The major difference is that in [7] our proposal was only defined for languages of positive tiles.

2.1 Adequately ordered monoids

Let S be a monoid partially ordered by a relation $\leq_S$ (or just $\leq$ when there is no ambiguity). We always assume that the order relation $\leq$ is stable under product, i.e. if $x \leq y$ then $xz \leq yz$ and $zx \leq zy$ for every x, y and $z \in S$. The set $U(S)$ of *subunits* of the partially ordered monoid S is defined by $U(S) = \{y \in S : y \leq 1\}$.

Definition 3 (Adequately ordered monoid). A partially ordered monoid S is an *adequately ordered monoid* when all subunits of S are idempotents, and, for every $x \in S$, both $x^L = \bigwedge\{y \in U(S) : xy = x\}$ and $x^R = \bigwedge\{y \in U(S) : yx = x\}$ exist in $U(S)$ with $x^R x = x x^L = x$. The subunit x^L (resp. x^R) is called the *left projection* (resp. the *right projection*) of the element x.

Since subunits are assumed to be idempotents, one can check that they commute and thus, ordered by the monoid order, form a meet semilattice with the product as the meet operator. It follows that when x is itself a subunit, we have $x = x^L = x^R$, i.e. both left and right projection mappings are indeed projection mappings from S onto $U(S)$.

Examples. Every monoid S extended with the trivial order $x \leq y$ only when $x = y$ is a adequately ordered monoid with $x^L = x^R = 1$ for every $x \in S$. Every inverse monoid S ordered by the natural order relation defined by $x \leq y$ when $x = xx^{-1}y$ (or equivalently $y = yx^{-1}x$) for every x and $y \in S$ is a adequately ordered monoid with $x^L = x^{-1}x$ and $x^R = xx^{-1}$. Especially, the monoid T_A^0 is an adequately ordered monoid. As shown in the next section, the relation monoid $\mathcal{P}(Q \times Q)$ ordered by inclusion is also an adequately ordered monoid.

Remark. For the reader familiar with the work initiated by Fountain [5], an ordered monoid S is adequately ordered exactly when it is $U(S)$-semiadequate in the sense of [13]. This suggests that, conversely, a U-semiadequate can be called adequately ordered when its *natural order* defined by $x \preceq y$ when $x = x^R y x^L$ can be *extended* into a partial order $\leq$ that is stable under product with $U = U(S)$. This is not necessarily the case. However, when such a extension exists, both orders $\preceq$ and $\leq$ coincide on subunits.

In [7], only U-semiadequate monoids with stable natural order were considered and shown to suffice for languages of positive tiles. Of course, every such monoid is also an adequately ordered monoid. The more relaxed definition proposed here copes with the fact that the natural order on the relation monoid $\mathcal{P}(Q \times Q)$ is not stable under product while the inclusion order, that extends the natural order, is stable.

Lemma 4. *Let S be an adequately ordered monoid. For every x and $y \in S$, if x and y are $\mathcal{R}$-equivalent (i.e. if $x \cdot S = y \cdot S$) then $x^R = y^R$ and, symmetrically, if x and y are $\mathcal{L}$-equivalent (i.e. if $S \cdot x = S \cdot y$) then $x^L = y^L$.*

In other words, left and right canonical local identities of a given element can be seen as some approximation of its left and right Green's classes.

Remark. We prove in [7] that every monoid S can be embedded into an adequately ordered monoid $\mathcal{Q}(S)$: the *quasi-inverse expansion* of S, in such a way that, for every (images of) two elements x and $y \in S$, we have $x^L = y^L$ (resp. $x^R = y^R$) in $\mathcal{Q}(S)$ if and only if x and y are $\mathcal{L}$-equivalent (resp. $\mathcal{R}$-equivalent). We will use a similar idea in the proof of Theorem 9 below.

2.2 Premorphisms and adequate premorphisms

A mapping $\varphi : S \to T$ between two adequately ordered monoids is a *premorphism* (or $\vee$-*premorphism* in [6]) when $\varphi(1) = 1$ and, for every x and $y \in S$, we have $\varphi(xy) \leq_T \varphi(x)\varphi(y)$ and if $x \leq_S y$ then $\varphi(x) \leq_T \varphi(y)$.

Definition 5 (Adequate premorphisms). A premorphism φ is an *adequate premorphism* when, for every $x \in S$, we have $\varphi(x^L) = (\varphi(x))^L$ and $\varphi(x^R) = (\varphi(x))^R$, and for every x and $y \in S$, if $xy \neq 0$ and $x^L \vee y^R = 1$ then $\varphi(xy) = \varphi(x)\varphi(y)$. In that latter case we say that the product xy is disjoint.

Lemma 6. *Let $\varphi : T_A^0 \to S$ be an adequate premorphism. The restriction of φ to (the overlapping tile image of) A^* is a morphism and, for every positive tile $(u, v, w) \in T_A^+$ one has $\varphi((u, v, w)) = (\varphi(u))^L \cdot \varphi(v) \cdot (\varphi(w))^R$.*

Symmetrically, the restriction of φ to the inverses of (the overlapping tile image of) A^* is also a morphism and, for every negative tile $(uv, \bar{v}, vw)$ one has $\varphi((uv, \bar{v}, vw)) = (\varphi(w))^R \cdot \varphi(v^{-1}) \cdot (\varphi(u))^L = (\varphi(w^{-1}))^L \cdot \varphi(v^{-1}) \cdot (\varphi(u^{-1}))^R$.

As a consequence, when S is finite, for every tile $x \in T_A^0$, the image $\varphi(x)$ of x by φ is effectively computable, in time linear in the size of the tile x, from the images of $\varphi(A)$, $\varphi(\bar{A})$ combined by product and right (or left) projection in S.

Proof. For every $u \in A^*$ we have $\varphi((1, u, 1))$ inductively computable by $\varphi(1) = 1$ and, for every $v \in A^*$ and $a \in A$, $\varphi((1, av, 1)) = \varphi((1, a, 1)) \cdot \varphi((1, v, 1))$ since the product $(1, av, 1) = (1, a, 1) \cdot (1, v, 1)$ is a disjoint product. By symmetry, the same holds for inverses. Indeed, for every tile x and y, if xy is a disjoint product then so is $(xy)^{-1} = y^{-1}x^{-1}$. It follows that $\varphi((u, \bar{u}, u))$ is also computable for every $u \in A^*$, Then, one can observe that for every tile $x = (u, v, w)$ we have $x = (u, 1, 1) \cdot (1, v, 1) \cdot (1, 1, w)$ with disjoint products and $(u, 1, 1) = (1, u, 1)^L$ and $(1, 1, w) = (1, w, 1)^R$. The adequacy assumption enables us to conclude. that $\varphi((u, v, w)) = (\varphi((1, u, 1)))^L \cdot \varphi((1, v, 1)) \cdot (\varphi((1, w, 1)))^R$ wich is thus computable. By symmetry, we also have $x^{-1} = (1, 1, w) \cdot (v, \bar{v}, v) \cdot (u, 1, 1)$ with disjoint products and $(1, 1, w) = (w, \bar{w}, w)^L$ and $(u, 1, 1) = (u, \bar{u}, u)^R$ hence we conclude similarly.

In these computations, right projections (or left projections) suffice since $(u, \bar{u}, u)^L = (1, u, 1)^R$ and $(u, \bar{u}, u)^R = (1, u, 1)^L$ for every $u \in A^*$. □

2.3 Quasi-recognizable languages

Definition 7 (Quasi-recognizable languages). A language of tiles $L \subseteq T_A$ is *quasi-recognizable* when there exists an adequate premorphism $\varphi : T_A^0 \to S$ in a finite adequately ordered monoid S such that $L = \varphi^{-1}(\varphi(L))$.

As far as computability and complexity issues are concerned, the Lemma 6 ensures that this notion of quasi-recognizability is effective in the sense that membership in a quasi-recognizable language L of tiles is computable. Deciding if a tile $x \in T_A$ belongs to the language L can even be done in bilinear time in the size of the tile x and in the size of the ordered monoid that quasi-recognizes the language L.

It is also quite an immediate consequence of Lemma 6 above that the quasi-recognizable languages of tiles are definable in MSO (see [9] for a definition of MSO logic over tiles). What about the converse ? It is a consequence of [7] that quasi-recognizability *essentially* capture MSO in the following sense. Given a new letter $\# \notin A$, given $\#(L) \subseteq T_{A+\#}$ defined for every language $L_\subseteq T_A$ by $\#(L) = \{(\#u, v, w\#) : (u, v, w) \in L\}$, we have: language $\#(L)$ is quasi-recognizable if and only if L is definable in MSO. Corollary 10, proved in the next section, provides a complete logical characterization of quasi-recognizable languages of tiles.

3 Tile automata vs quasi-recognizability

In this section, we relate the notions of finite state tile automata and the notion of quasi-recognizable languages of tiles.

3.1 From tile automata to quasi-recognizability

Let $\mathcal{A} = \langle Q, \delta, K \rangle$ be a finite non deterministic tile automaton. The run image of the positive tile $x = (u, v, w) \in T_A^+$ is defined as the set of pairs of states

$\varphi_{\mathcal{A}}((u,v,w)) = \{(p,q) \in Q \times Q : \exists s, e \in Q, s \xrightarrow{u} p, p \xrightarrow{v} q, q \xrightarrow{w} e\}$, i.e. the set of all runs of the tile automaton $\mathcal{A}$ over u.

Theorem 8. *Every language of tiles definable by a finite state tile automaton is quasi-recognizable.*

Proof. Let $\mathcal{A} = \langle Q, \delta, K \subseteq Q \times Q \rangle$ be a tile automaton and let $\varphi_{\mathcal{A}} : T_A \to \mathcal{P}(Q \times Q)$ be the run mapping induced by $\mathcal{A}$. We essentially have to prove that $\mathcal{P}(Q \times Q)$ equipped with the product

$$X \cdot Y = \{(p,q) \in Q \times Q : \exists r \in Q, (p,r) \in X, (r,q) \in Y\}$$

and ordered by inclusion is an adequately ordered monoid and that $\varphi_{\mathcal{A}}$ extended to 0 by taking $\varphi_{\mathcal{A}}(0) = \emptyset$ is an adequate premorphism.

The fact that $\mathcal{P}(Q \times Q)$ equipped with the relation product is a stable ordered monoid with neutral element $I_Q = \{(q,q) \in Q \times Q : q \in Q\}$ and inclusion ordered is a classical result [18]. Since subunits are obviously idempotents, it suffices thus to prove that canonical left and right identities exist.

Let $X \in \mathcal{P}(Q \times Q)$ and let $X^R = \{(p,p) \in Q \times Q : \exists q \in Q, (p,q) \in X\}$ and let $X^L = \{(q,q) \in Q \times Q : \exists p \in Q, (p,q) \in X\}$. One can easily check that $X = X^R \cdot X = X \cdot X^L$ for every $X \subseteq Q \times Q$. Let then $Y \subseteq I_Q$ such that $Y \cdot X = X$ (resp. $X \cdot Y = X$). It is an immediate observation that this implies $X^R \subseteq Y$ (resp. $X^L \subseteq Y$). In other words, X^L (resp. X^R) is indeed the least right (left) local unit for X.

The fact that $\varphi_{\mathcal{A}}$ extended to zero as defined above is an premorphism raises no real difficulty. By definition, we have $\varphi_{\mathcal{A}}(1) = I_Q$ and it is rather immediate that $\varphi_{\mathcal{A}}(u) \subseteq \varphi_{\mathcal{A}}(v)$ whenever $u \leq v$ in T_A. The fact that we also have $\varphi_{\mathcal{A}}(uv) \leq \varphi_{\mathcal{A}}(u)\varphi_{\mathcal{A}}(v)$ for every u and $v \in T^0_A$ is a little more complex to check but with no special difficulty.

The fact that $\varphi_{\mathcal{A}}$ is also adequate is somehow simpler and essentially follows from the definition. In particular, the disjoint product case just mimics the classical case where $\varphi_{\mathcal{A}}$ is defined over words only. □

As an immediate consequence of the definition of $\varphi_{\mathcal{A}}$, writing $X^{-1} = \{(q,p) \in Q \times Q : (p,q) \in X\}$ for every relation $X \subseteq Q \times Q$, we have $\varphi_{\mathcal{A}}(u^{-1}) = (\varphi_{\mathcal{A}}(u))^{-1}$ and thus we also have $L^-(\mathcal{A}) = \{u \in T^-_A : (\varphi_{\mathcal{A}}(u^{-1}))^{-1} \in K\}$. In general, this property is not satisfied by an arbitrary adequate premorphism. However, we prove in Theorem 9 below that every adequate premorphism can still be translated into an equivalent premorphism of the form $\varphi_{\mathcal{A}}$ for some finite state automaton $\mathcal{A}$.

3.2 From quasi-recognizability to tile automata

Theorem 9. *For every quasi-recognizable language of tiles $L \subseteq T_A$ there exists a finite state non deterministic tile automaton $\mathcal{A}$ such that $L = \varphi_{\mathcal{A}}^{-1}(\varphi_{\mathcal{A}}(L))$, i.e. the adequate premorphism induced by $\mathcal{A}$ recognizes the language L.*

Proof. Let $\psi : T_A^0 \to S$ be an adequate premorphism into a finite adequately ordered monoid S let $K_\psi \subseteq S$ and let $L = \psi^{-1}(K_\psi)$. We want to build a finite state automaton $\mathcal{A} = \langle Q, \delta, K \rangle$ such that $\varphi_{\mathcal{A}}^{-1}(K) = \psi^{-1}(K_\psi)$.

To achieve this goal it is sufficient to define an automaton $\mathcal{A}$ that, given any positive tile $x = (u, v, w)$ (resp. negative tile $x^{-1} = (uv, \bar{v}, vw)$) as input, computes, via $\varphi_{\mathcal{A}}(x)$ (resp. $\varphi_{\mathcal{A}}(x^{-1})$), the *left ideal* $S \cdot \psi(u)$ associated to $\psi(u)$, the *right ideal* $\psi(w) \cdot S$ associated to $\psi(w)$ and the *image* $\psi(v)$ of the root v of x (resp. the *image* $\psi(v^{-1})$ of the root $\bar{v}$ of x^{-1}).

Indeed, by Lemma 4, computing these left and right ideals is enough to compute the expected left and right canonical identities $\psi(u^L)$ and $\psi(w^R)$. Then, by applying Lemma 6, together with the value of $\psi(v)$ (resp. $\psi(v^{-1})$), we can compute the value $\psi(x)$ of x (resp. $\psi(x^{-1})$ of x^{-1}) by the premorphism ψ.

Extending the idea described in the automata section in order to distinguish positive from negative tiles, the automaton $\mathcal{A}$ is defined as follows.

The set of states Q is defined to be $Q = S \times S \times S \times M$ with set of modes $M = \{P, S, PR, NR, c, p_1, p_2, n_1, n_2\}$ where P, S, PR and NR respectively stand for the "stable" automaton mode prefix, suffix, positive root and negative root automaton modes, and c, p_1, p_2, n_1 and n_2 stand for the "frontier" modes that occur at most once in between stable modes.

For every $a \in A$, the set $\delta(a)$ of transitions labeled by a is defined to be the union of the following sets of transitions:

- ▷ "prefix" transitions, from states in mode P to states in mode $m \in \{P, c, p_1, n_1\}$, of the form $\{((x, y, z, P), (x \cdot \varphi(a), y, z, m) : x, y, z \in Q\}$,
- ▷ "positive root" transitions, from states in mode $m \in \{PR, p_1\}$ to states in mode $k \in \{PR, p_2\}$, of the form $\{((x, y, z, m), (x, y \cdot \varphi(a), z, k)) : x, y, z \in S\}$,
- ▷ "negative root" transitions, from states in mode $m \in \{PN, n_1\}$ to states in mode $k \in \{PN, n_2\}$, of the form $\{((x, y, z, m), (x, \varphi(a^{-1}) \cdot y, z, k)) : x, y, z \in S\}$,
- ▷ "suffix" transitions, from states in mode $m \in \{S, c, p_2, n_2\}$ to states in mode S, of the form $\{((x, y, \varphi(a) \cdot z, m), (x, y, z, S)) : x, y, z \in S\}$.

Of course, such an automaton will run freely on tiles regardless of whether it is running on the prefix, the root or the suffix part of a tile. However, by watching the states in frontier modes occurring at the extremities of the root, we can collect all the information we need.

More precisely, the next step is then to keep from the set of all runs $\varphi_{\mathcal{A}}(x)$ of $\mathcal{A}$ on any given input tile $x \in T_{\mathcal{A}}$ only the relevant data. This is achieved by the following mapping. For every $X \subseteq \mathcal{P}(Q \times Q)$, we define the relevant image $f(X) \subseteq S \times S \times S$ of X "in" S by taking:

$$\begin{aligned} f(X) = \{(x,1,z) \in S \times S \times S : ((x,1,z,c),(x,1,z,c)) \in X\} \quad (1) \\ \cup\{(x,y,z) \in S \times S \times S : ((x,1,z,p_1),(x,y,z,p_2)) \in X\} \quad (2) \\ \cup\{(x,y,z) \in S \times S \times S : ((x,1,z,n_2),(x,y,z,n_1)) \in X\} \quad (3) \end{aligned}$$

where line (1) treats the case of context tiles, line (2) treats the case of (strictly) positive tiles and line (3) treats the case of (strictly) negative tiles.

With this construction, one can show that for every $x = (u,v,w) \in T_A^+$ we have: $f(\varphi_{\mathcal{A}}(x)) = S \cdot \psi(u) \times \{\psi(v)\} \times \psi(w) \cdot S$ and, for every $x^{-1} = (uv, \bar{v}, vw) \in T_A^-$ we have $f(\varphi_{\mathcal{A}}(x^{-1})) = S \cdot \psi(u) \times \{\psi(v^{-1})\} \times \psi(w) \cdot S$. In other words, for every $x \in T_A$, the finite value of $f(\varphi_{\mathcal{A}}(x))$ completely characterizes $\psi(x)$ thus we conclude the proof by taking

$$\begin{aligned} K = f^{-1}\left(\{S \cdot \psi(u) \times \{\psi(v)\} \times \psi(w) \cdot S : \psi(u^L) \cdot \psi(v) \cdot \psi(w^R) \in K_\psi\}\right) \\ f^{-1}\left(\{S \cdot \psi(u) \times \{\psi(v^{-1})\} \times \psi(w) \cdot S : \psi(w^R) \cdot \psi(v^{-1}) \cdot \psi(u^L) \in K_\psi\}\right) \end{aligned}$$

By construction, for every tile $x \in T_A$, we indeed have $x \in \varphi_{\mathcal{A}}^{-1}(K)$ if and only if $x \in \psi^{-1}(K_\psi)$. □

Observe that if we consider a language $L \subseteq T_A^+$ of *positive* tiles that is recognizable by a premorphism ψ from the monoid of positive tiles T_A^+ into an adequately ordered monoid S then the above proof can easily be adapted so that $\varphi_{\mathcal{A}} : T_A^0 \to \mathcal{P}(Q \times Q)$ still recognizes L. In other words, a quasi-recognizable language of positive tiles is also quasi-recognizable as a language of arbitrary tiles. This proves that the work presented here indeed generalizes the results formerly obtained in [7].

Corollary 10. *For every language of tiles $L \subseteq T_A$, the language $L \subseteq T_A$ is quasi-recognizable if and only if L is a finite boolean combination of upward closed MSO-definable languages of tiles.*

Proof. Immediate consequence of Theorem 2 and Theorem 9. □

Conclusion

We have shown that the emerging notion of quasi-recognizability, defined in [7] as a remedy to the collapse of classical recognizability by monoid morphisms, can be equipped with quite a simple notion of finite state automata that, applied to languages of overlapping tiles, captures its expressiveness. Extending the theory of languages of words, this new theory of languages of tiles inherits several and significantly robust features of that classical language theory.

Compared to our former proposal [7], the notion of quasi-recognizability has also been refined - especially via the notion of disjoint products - and can now be applied to more general settings. Forthcoming studies [8] even show that the notion of recognizability by adequate premorphisms and the related notion of non deterministic finite state automata can even be extended, with similar logical characterizations, to languages of labeled birooted trees generalizing thus the known algebraic characterizations of regular languages of finite trees.

Based on the notion of U-semiadequate monoids [13], the present work also provides a rather unexpected application of the study of non regular semigroups initiated by Fountain [5] in the 70's. Our proposal still need to be further related with the research lines developed in that field such as, for instance, the notion of partial actions [2, 3].

References

1. F. Berthaut, D. Janin, and B. Martin. Advanced synchronization of audio or symbolic musical patterns: an algebraic approach. *International Journal of Semantic Computing*, 6(4):1–19, 2012.
2. C. Cornock. *Restriction Semigroups: Structure, Varieties and Presentations.* PhD thesis, Departement of Mathematics University of York, 2011.
3. C. Cornock and V. Gould. Proper two-sided restriction semigroups and partial actions. *Journal of Pure and Applied Algebra*, 216:935–949, 2012.
4. A. Dicky and D. Janin. Two-way automata and regular languages of overlapping tiles. Technical Report RR-1463-12, LaBRI, Université de Bordeaux, 2012.
5. J. Fountain. Adequate semigroups. *Proc. Edinburgh Math. Soc.*, 22(2):113–125, 1979.
6. C. D. Hollings. The Ehresmann–Schein–Nambooripad Theorem and its successors. *European Journal of Pure and Applied Mathematics*, 5(4):414–450, 2012.
7. D. Janin. Quasi-recognizable vs MSO definable languages of one-dimensional overlaping tiles. In *Mathematical Foundations of computer Science (MFCS)*, volume 7464 of *LNCS*, pages 516–528. Springer-Verlag, 2012.
8. D. Janin. Algebras, automata and logic for languages of labeled birooted trees. Technical Report RR-1467-13, LaBRI, Université de Bordeaux, 2013.
9. D. Janin. On languages of one-dimensional overlapping tiles. In *International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, volume 7741 of *LNCS*, pages 244–256. Springer-Verlag, 2013.

10. D. Janin, F. Berthaut, M. DeSainte-Catherine, Y. Orlarey, and S. Salvati. The T-calculus : towards a structured programming of (musical) time and space. Technical Report RR-1466-13, LaBRI, Université de Bordeaux, 2013 (to appear).
11. J. Kellendonk. The local structure of tilings and their integer group of coinvariants. *Comm. Math. Phys.*, 187:115–157, 1997.
12. J. Kellendonk and M. V. Lawson. Universal groups for point-sets and tilings. *Journal of Algebra*, 276:462–492, 2004.
13. M. V. Lawson. Semigroups and ordered categories. i. the reduced case. *Journal of Algebra*, 141(2):422 – 462, 1991.
14. M. V. Lawson. *Inverse Semigroups : The theory of partial symmetries*. World Scientific, 1998.
15. M. V. Lawson. McAlister semigroups. *Journal of Algebra*, 202(1):276 – 294, 1998.
16. S. W. Margolis and J.-E. Pin. Languages and inverse semigroups. In *ICALP*, volume 172 of *Lecture Notes in Computer Science*, pages 337–346. Springer, 1984.
17. W. D. Munn. Free inverse semigroups. *Proceeedings of the London Mathematical Society*, 29(3):385–404, 1974.
18. J-.E. Pin. Chap. 10. Syntactic semigroups. In *Handbook of formal languages, Vol. I*, pages 679–746. Springer Verlag, 1997.
19. P. V. Silva. On free inverse monoid languages. *ITA*, 30(4):349–378, 1996.