

Overlapping tile automata:

towards a language theory of overlapping structures

David Janin
LaBRI, Université de Bordeaux

Computer Science in Russia,
Ekaterinburg, june 2013

1. From strings to overlapping tiles

As an introduction

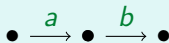
From strings to strings with memory

Alphabet $A = \{a, b, c, \dots\}$ and dual alphabet $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$

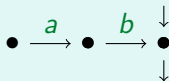
Right action sequence w with **add** (A) or **remove** (\bar{A})

1

Memoryless resulting value $s \cdot w$: **string**



Resulting value with **linear** memory: **overlapping tiles**



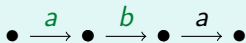
From strings to strings with memory

Alphabet $A = \{a, b, c, \dots\}$ and dual alphabet $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$

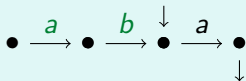
Right action sequence w with **add** (A) or **remove** (\bar{A})

a

Memoryless resulting value $s \cdot w$: **string**



Resulting value with **linear** memory: **overlapping tiles**



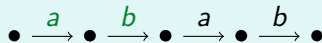
From strings to strings with memory

Alphabet $A = \{a, b, c, \dots\}$ and dual alphabet $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$

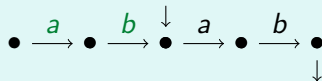
Right action sequence w with **add** (A) or **remove** (\bar{A})

ab

Memoryless resulting value $s \cdot w$: **string**



Resulting value with **linear** memory: **overlapping tiles**



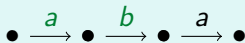
From strings to strings with memory

Alphabet $A = \{a, b, c, \dots\}$ and dual alphabet $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$

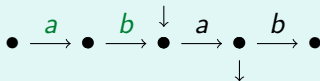
Right action sequence w with **add** (A) or **remove** (\bar{A})

$$ab\bar{b}$$

Memoryless resulting value $s \cdot w$: **string**



Resulting value with **linear** memory: **overlapping tiles**



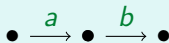
From strings to strings with memory

Alphabet $A = \{a, b, c, \dots\}$ and dual alphabet $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$

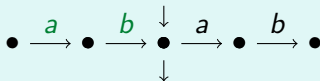
Right action sequence w with **add** (A) or **remove** (\bar{A})

$ab\bar{b}\bar{a}$

Memoryless resulting value $s \cdot w$: **string**



Resulting value with **linear** memory: **overlapping tiles**



From strings to strings with memory

Alphabet $A = \{a, b, c, \dots\}$ and dual alphabet $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$

Right action sequence w with **add** (A) or **remove** (\bar{A})

$$abb\bar{a}\bar{b}$$

Memoryless resulting value $s \cdot w$: **string**

$$\bullet \xrightarrow{a} \bullet$$

Resulting value with **linear** memory: **overlapping tiles**

$$\begin{array}{ccccccc} \bullet & \xrightarrow{a} & \bullet & \xrightarrow{b} & \bullet & \xrightarrow{a} & \bullet & \xrightarrow{b} & \bullet \\ & & \downarrow & & \downarrow & & & & \\ & & & & & & & & \end{array}$$

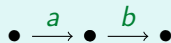
From strings to strings with memory

Alphabet $A = \{a, b, c, \dots\}$ and dual alphabet $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$

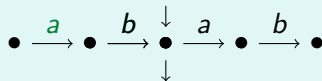
Right action sequence w with **add** (A) or **remove** (\bar{A})

$abb\bar{a}\bar{b}b$

Memoryless resulting value $s \cdot w$: **string**



Resulting value with **linear** memory: **overlapping tiles**



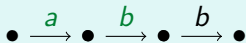
From strings to strings with memory

Alphabet $A = \{a, b, c, \dots\}$ and dual alphabet $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$

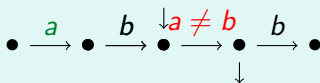
Right action sequence w with **add** (A) or **remove** (\bar{A})

$ab\bar{b}\bar{a}b\bar{b}b$

Memoryless resulting value $s \cdot w$: **string**



Resulting value with **linear** memory: **overlapping tiles**



Undefined !

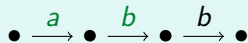
From strings to strings with memory

Alphabet $A = \{a, b, c, \dots\}$ and dual alphabet $\bar{A} = \{\bar{a}, \bar{b}, \bar{c}, \dots\}$

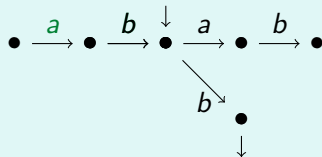
Right action sequence w with **add** (A) or **remove** (\bar{A})

$ab\bar{b}\bar{a}bbb$

Memoryless resulting value $s \cdot w$: **string**



Resulting value with **tree-shaped** memory: **birooted trees**



(see ICALP 2013)

(Towards) non classical language theory

What language theoretic tools ?

Automata, partial algebra, logic. . . for languages of sequences of actions ?

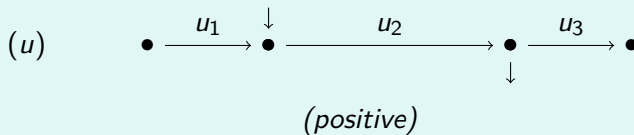
2. Overlapping tiles monoids

Towards McAlister (inverse) monoid [5]

The monoid T_A of overlapping tiles (McAlister 70's)

Given alphabet A :

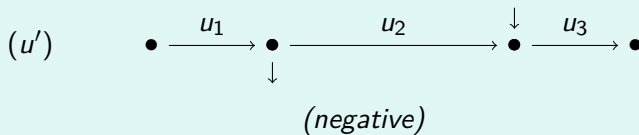
Overlapping tiles



The monoid T_A of overlapping tiles (McAlister 70's)

Given alphabet A :

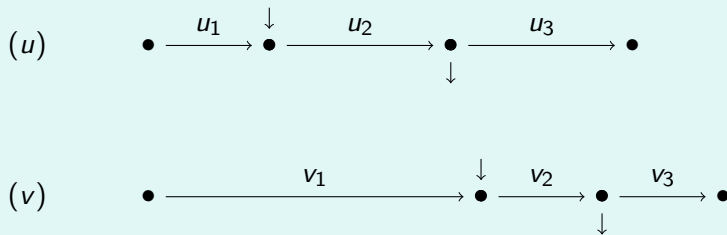
Overlapping tiles



The monoid T_A of overlapping tiles (McAlister 70's)

Given alphabet A :

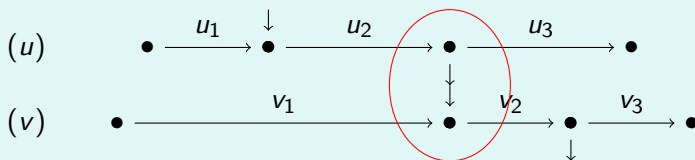
Overlapping tiles product



The monoid T_A of overlapping tiles (McAlister 70's)

Given alphabet A :

Overlapping tiles product

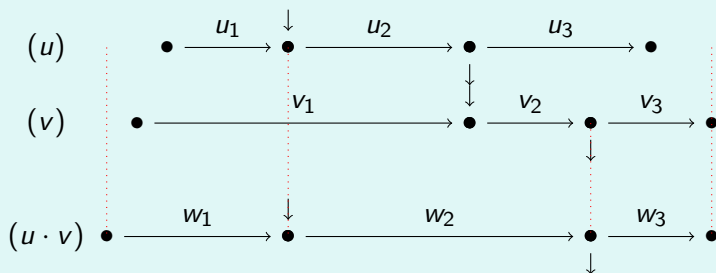


Synchronization

The monoid T_A of overlapping tiles (McAlister 70's)

Given alphabet A :

Overlapping tiles product

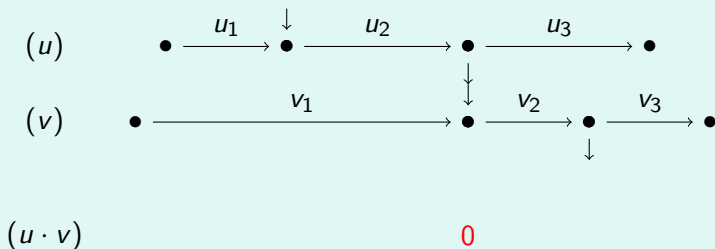


Fusion when $u_1 u_2 \vee_s v_1 \neq 0$ and $u_3 \vee_p v_2 v_3 \neq 0$

The monoid T_A of overlapping tiles (McAlister 70's)

Given alphabet A :

Overlapping tiles

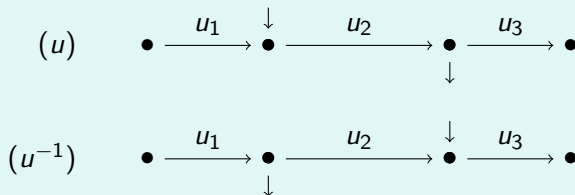


and **undefined** otherwise

Inverses

Monoid T_A^0 is an inverse monoid [6, 4]

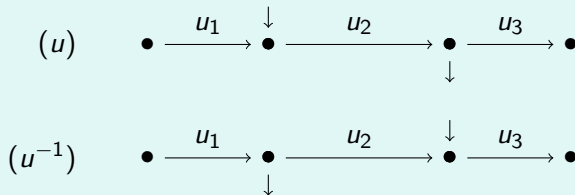
Inverses : $u \cdot u^{-1} \cdot u = u$ and $u^{-1} \cdot u \cdot u^{-1} = u^{-1}$ (unique)



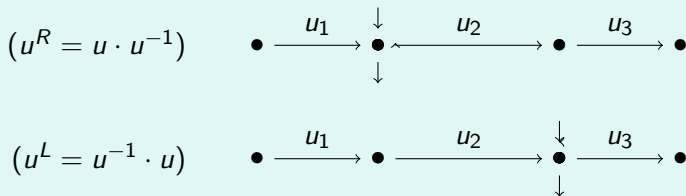
Inverses and idempotents

Monoid T_A^0 is an inverse monoid [6, 4]

Inverses : $u \cdot u^{-1} \cdot u = u$ and $u^{-1} \cdot u \cdot u^{-1} = u^{-1}$ (unique)

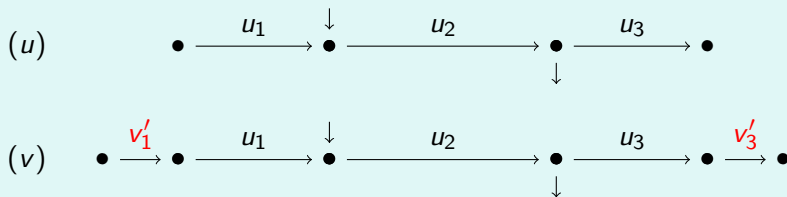


Idempotents : $e \cdot e = e$ (or projections uu^{-1} and $u^{-1}u$)



Natural order

Definition: $0 \leq v \leq u$



Lemma

$$v \leq u \text{ iff } v = v^R u \text{ iff } v = uv^L$$

Lemma (subunits vs idempotents vs self inverse)

$$e \leq 1 \text{ iff } e \cdot e = e \text{ iff } e^{-1} = e$$

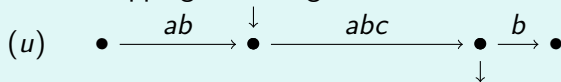
3. Overlapping tile automata

Automata for overlapping tiles

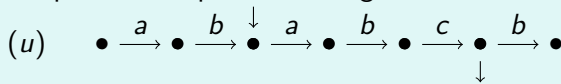
Overlapping tiles as FO-structures

Induced FO-structure

Every non zero overlapping tile u , e.g.



is just seen as a linear directed graph with labeled edge and distinguished input and output vertex, e.g.



with :

- domain $dom(u)$ as set of vertices,
- relations $\xrightarrow{a} \subseteq dom(u) \times dom(u)$,
one for every letter $a \in A$, as edge relations,
- and input in_u and output $out_u \in dom(u)$.

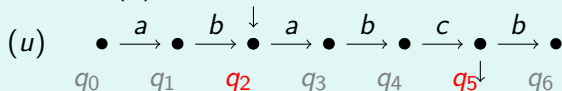
Overlapping tile automata

Automaton : $\mathcal{A} = \langle Q, \delta, W \rangle$

with a (finite) set of **states** Q , a **transition table** $\delta : A \rightarrow \mathcal{P}(Q \times Q)$
and an **acceptance pairs** $W \subseteq Q \times Q$.

Run of \mathcal{A} on a tile $u : \rho : \text{dom}(u) \rightarrow Q$

such that, if $x \xrightarrow{a} y$ then $(\rho(x), \rho(y)) \in \delta(a)$
for every $x, y \in \text{dom}(u)$, every $a \in A$,



with $(q_0, q_1), (q_2, q_3) \in \delta(a)$, $(q_1, q_2), (q_3, q_4), (q_5, q_6) \in \delta(b)$ and
 $(q_4, q_5) \in \delta(c)$.

Accepting : $(\rho(\text{in}_u), \rho(\text{out}_u)) \in W$

Tile automata languages

Recognized languages $L(\mathcal{A})$

defined as the set of (non zero) overlapping tile u such that there is a an accepting run of \mathcal{A} on u .

Remark

Every language L recognized by an automaton is upward closed in the natural order and definable in MSO.

Theorem (MSO[↑])

A language of overlapping tiles $L \subseteq T_{\mathcal{A}} - 0$ is recognized by a finite state tile automaton if and only if L is definable in MSO and upward closed.

Corollary

*This class of languages is closed under union, intersection, projection, inverse, and . . . **upward closed** product or star.*

4. Quasi-recognizability

Algebras for the boolean closure of upward closed MSO definable languages

Towards an associated algebra

Induced premorphism

Let $\mathcal{A} = \langle Q, \delta, \Delta, W \rangle$ be an automaton and let

$$\varphi : T_{\mathcal{A}} \rightarrow \mathcal{P}(Q \times Q)$$

be the mapping defined by $\varphi(0) = \emptyset$ and, for every non zero tile u .

$$\varphi(u) = \{(\rho(in_u), \rho(out_u)) \in Q \times Q : \text{run } \rho \text{ of } \mathcal{A} \text{ on } u\}$$

Lemma

The mapping φ recognizes $L(\mathcal{A})$ in the sense that $L(\mathcal{A}) = \varphi^{-1}(\varphi(L(\mathcal{A})))$.

Remark

With $X \subseteq \mathcal{P}(Q \times Q)$ the language $\varphi^{-1}(X) - 0 \subseteq T_{\mathcal{A}}$ is $Bool_f(MSO^\uparrow)$.

Towards an associated algebra

Lemma

The sub-monoid induced by $\varphi(T_A)$ ordered by inclusion is a partially ordered monoid.

Lemma

The mapping φ is a \vee -premorphisms, i.e. for every B and $C \in T_A$:

- *if $B \leq C$ then $\varphi(B) \subseteq \varphi(C)$*
- *$\varphi(B \cdot C) \subseteq \varphi(B) \cdot \varphi(C)$*

Question

Find a more complete axiomatization of these ordered monoids and premorphisms ?

Algebras for languages of overlapping tiles

Definition (Adequately ordered monoid)

A monoid S such that:

- **the order is stable**, i.e. for every x, y and $z \in S$, if $x \leq y$ then $z \cdot x \leq z \cdot y$ and $x \cdot z \leq x \cdot y$,
- **subunits are idempotents**, i.e. for every $x \in U(S) = \{x \in S : x \leq 1\}$ we have $xx = x$,
- for every $x \in S$, both the **left projection** $x^L = \min\{z \in U(S) : x \cdot z = x\}$ and the **right projection** $x^R = \min\{z \in U(S) : z \cdot x = x\}$ exists (in $U(S)$).

Examples

- (1) Every trivially ordered monoid with $x^L = x^R = 1$.
- (2) Every naturally ordered inverse monoid with $x^L = x^{-1}x$ and $x^R = xx^{-1}$.
- (3) Every finite partially ordered monoid with idempotent subunits.

Algebras for languages of overlapping tiles

Definition (Adequate premorphism)

A mapping $\varphi : S \rightarrow T$ with S and T two adequately ordered monoids such that:

- φ is **monotonic**, **submultiplicative** with $\varphi(1) = 1$.
- φ preserves **disjoint product**, i.e. for every x and $y \in S$ such that $x \cdot y \neq 0$ and $x^L \vee y^R = 1$ we have $\varphi(x \cdot y) = \varphi(x) \cdot \varphi(y)$,
- φ preserves **left** and **right projections**, i.e. for every $x \in S$ we have $\varphi(x^L) = (\varphi(x))^L$ and $\varphi(x^R) = (\varphi(x))^R$.

Example

The canonical mapping $\varphi : T_A \rightarrow \mathcal{P}(Q \times Q)$ induced by an automaton \mathcal{A} .

Quasi-recognizable languages

Definition

A language $L \subseteq T_A$ is **quasi-recognizable** (QREC) when there exists a finite adequately ordered monoid S and an adequate premorphism $\varphi : T_A \rightarrow S$ such that $L = \varphi^{-1}(\varphi(L))$.

Lemma (Effectiveness)

Let $\varphi : T_A \rightarrow S$ be an adequate premorphism with finite S . For every $B \in T_A$, the image $\varphi(B)$ of B is computable in linear time in B .

Proof.

Thanks to some canonical disjoint decomposition property. □

Expressiveness of QREC

Theorem

A language L is QREC if and only if L is a finite boolean combination of languages recog. by finite state automata, i.e. $QREC = \text{BOOL}(\text{MSO}^\uparrow)$.

proof.

$(QREC) \Rightarrow \text{MSO}$: simulate disjoint decomposition.

$(QREC + \text{MSO}) \Rightarrow \text{BOOL}(\text{MSO}^\uparrow)$: let $\varphi : T_A \rightarrow S$ an adequate premorphism with finite S . For every $x \in S$ we have:

$$\varphi^{-1}(x) = \underbrace{\left(\bigcup_{y \in x^\uparrow} \varphi^{-1}(y) \right)}_{\text{MSO}^\uparrow} \cap \underbrace{\left(\bigcup_{y \in x^\downarrow} \varphi^{-1}(y) \right)}_{\text{MSO}^\downarrow}$$

Expressiveness of QREC

proof (continued).

$\text{BOOL}(\text{MSO}^\uparrow) \Rightarrow \text{QREC}$: since $\text{MSO}^\uparrow \subseteq \text{QREC}$ and QREC is closed under complement, it suffices to prove closure under union. Let $\varphi_1 : T_A \rightarrow S_1$ and $\varphi_2 : T_A \rightarrow S_2$ two adequate premorphisms with finite S_1 and S_2 .

Then $S_1 \times S_2$ is adequately ordered and

$$\varphi = \langle \varphi_1, \varphi_2 \rangle : T_A \rightarrow S_1 \times S_2$$

recognizes the union of any languages L_1 and L_2 resp. recognized by φ_1 and φ_2 .



5. Beyond automata

Where we eventually show that QREC is strictly included in MSO
(this result is obtained after the presentation, following Marc
Zeitoun remark)

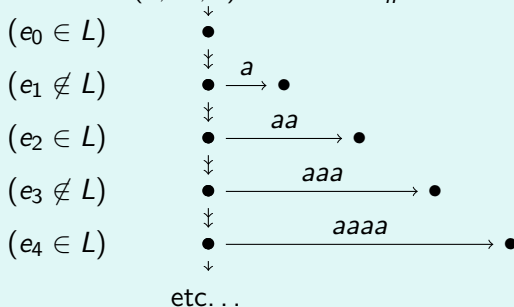
BOOL(MSO[↑]) vs MSO

Theorem

There are MSO definable languages of tiles that are not QREC henceforth, equivalently, not BOOL(MSO[↑]).

Proof.

Take $u_n = (1, a^n, 1)$, $e_n = u_n \cdot u_n^{-1}$ and $L = \{e_{2n} : n \in \omega\}$.



6. Conclusion

Works in progress

- QREC and languages of finite trees [1].
- Tile programming [2].
- Musical tiles [3]
- “Syntactic” adequately ordered monoid ?
- Birooted infinite words or even trees ?

[1] D. Janin.

Algebras, automata and logic for languages of labeled birooted trees.

In *Int. Col. on Aut., Lang. and Programming (ICALP)*, volume 7966 of *LNCS*, pages 318–329. Springer, 2013.

[2] 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.

[3] D. Janin, F. Berthaut, and M. DeSainteCatherine.

Multi-scale design of interactive music systems : the libTuiles experiment.

In *Sound and Music Computing (SMC)*, 2013.

[4] M. V. Lawson.

Inverse Semigroups : The theory of partial symmetries.

World Scientific, 1998.

[5] M. V. Lawson.

McAlister semigroups.

Journal of Algebra, 202(1):276 – 294, 1998.

[6] M. Pietrich.

Inverse semigroups.

Wiley, 1984.