

From Parker's bebop to Lawson quasi-inverse monoids

David Janin,
Université de Bordeaux
LaBRI

Séminaire LACL - Juin 2012

Play

Music is the unique language where many people can talk at the same time understanding each other. . . .

Amadeus, XVIIIe siècle,

(from M. Forman movie)

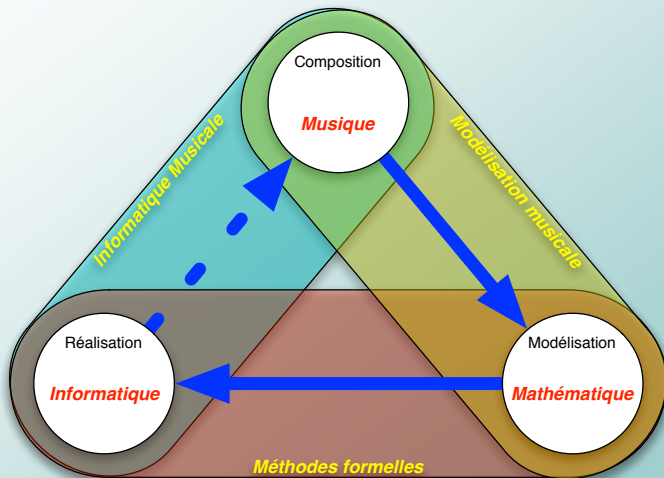
Nowadays, two centuries later, we eventually formalized what composers were doing every day : multiplexing.

For centuries, music is also a language of time, space, parallelism and interaction,...

Shall music modeling lead us to the discovery of relevant and new *metaphors* or *paradigms* useful for modeling computerized systems?

Research field :

Theoretical foundation of computational music



1. Overlapping tiles

From Bebop modeling [Jan11]...

My little blue suede shoes (Ch. Parker)



The image displays two staves of musical notation for the song "My little blue suede shoes" by Charles Parker. The music is in the key of B-flat major (two flats) and 4/4 time. The first staff shows three measures of music, each starting with a red label (a) under a bracketed group of notes. The second staff shows two measures of music, each starting with a red label (a) and (b) under a bracketed group of notes. The notation includes treble clefs, a key signature of two flats, and a common time signature. The notes are primarily eighth and quarter notes, with some rests and ties.

Analysis

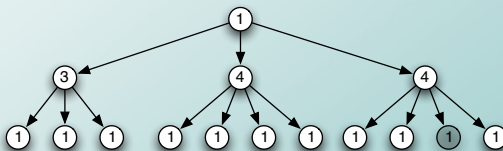
Three exposures a given pattern (a) followed by a variation (b) of the same pattern.

Play

Modeling pattern (a)



modeled as:

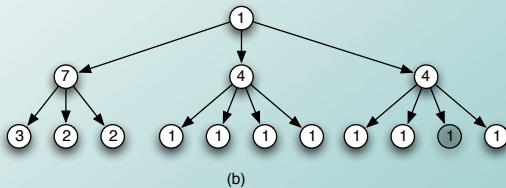


(a)

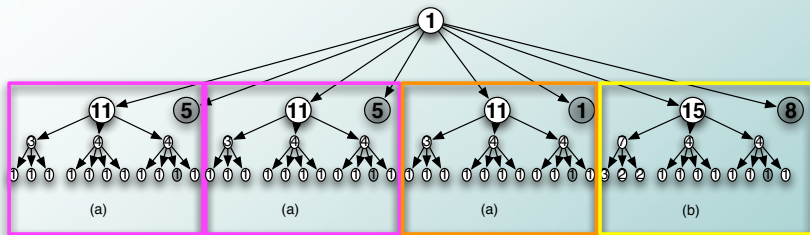
Modeling pattern (b)



modeled as:



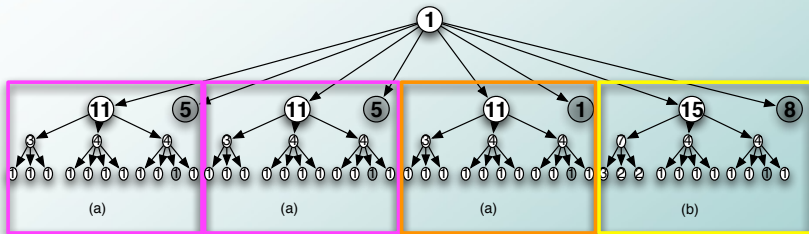
Resulting modeling



Drawbacks:

- inserted silences of many lengths,
- logical structure $(3 \times (a) + (b))$ lost.

Resulting modeling



Drawbacks:

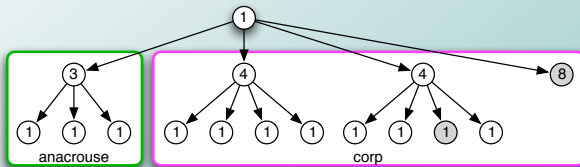
- inserted silences of many lengths,
- logical structure $(3 \times (a) + (b))$ lost.

A better approach

Make anticipation of the first logical beat explicit (anacrusis):



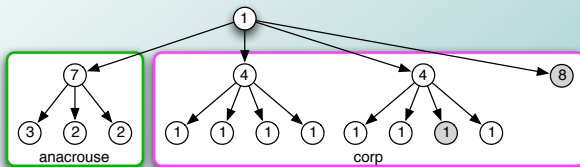
and make explicit the silence till the end of the next bar:



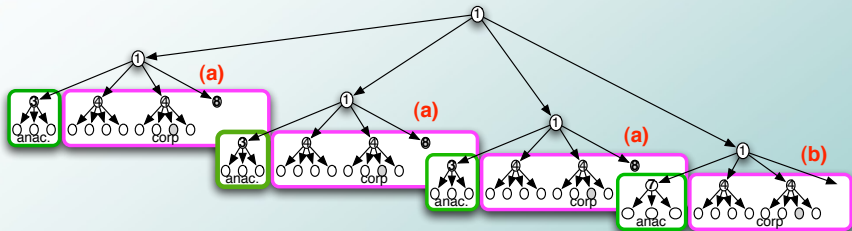
Do the same for the second pattern:



which give:



with resulting *sequential composition*:



defined with *local overlapping* of anacrusis and preceding pattern body.

And there is our logical structure : $3x(a) + (b) !$

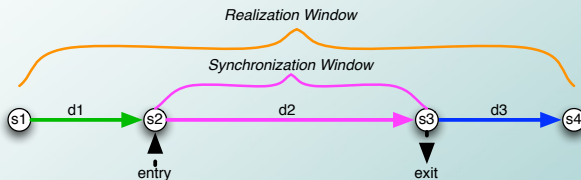
2. Formalization

Towards a algebraic structuration of time...

Synchronization window vs realization window

Hint [Jan11]

Distinguish, in every pattern:



introduction, development and conclusion.

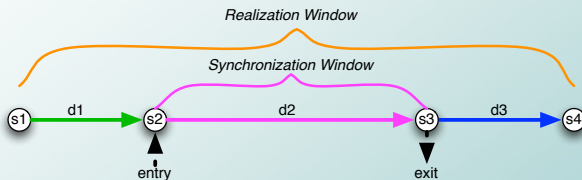
Old idea

Already occurring (though in an ad hoc way) in musical modeling LOCO [DH88], but also in automation or for process modeling in software system

Synchronization window vs realization window

Hint [Jan11]

Distinguish, in every pattern:

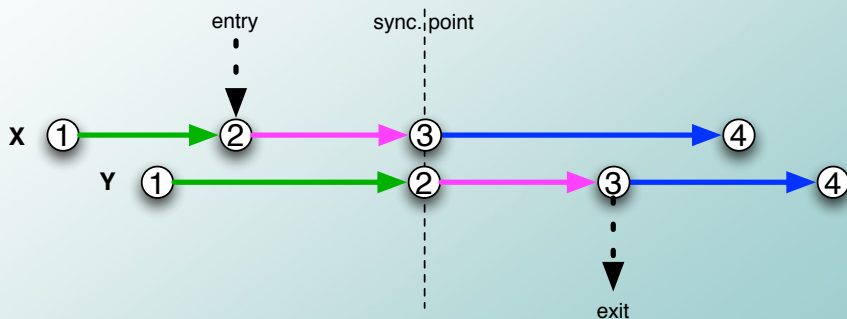


introduction, development and conclusion.

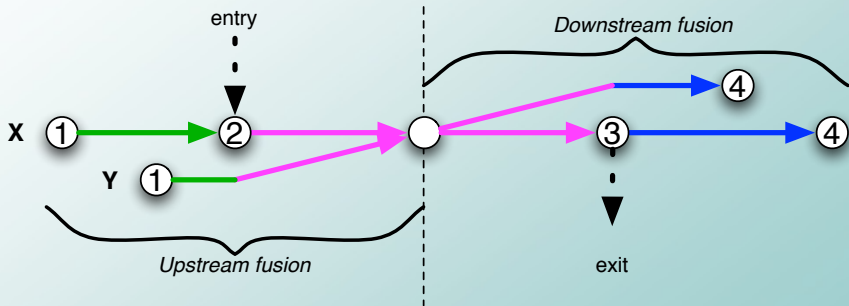
Old idea

Already occurring (though in an ad hoc way) in musical modeling [LOCO](#) [DH88], but also in [automation](#) or for process modeling in software system.

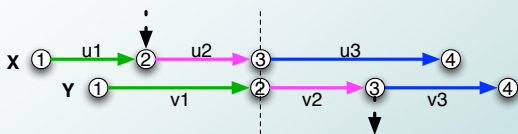
Sequential product : 1. synchronisation



Sequential product : 2. fusion



Induced algebra: triples of durations



Triples of duration : $D_A = \mathbb{R} \times \mathbb{R} \times \mathbb{R}$, with product:

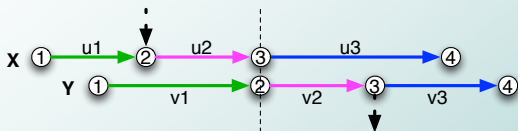
$$\underbrace{(u_1, u_2, u_3)}_U \cdot \underbrace{(v_1, v_2, v_3)}_V = \underbrace{(\max(u_1, v_1 - u_2), u_2 + v_2, \max(v_3, u_3 - v_2))}_{U.V}$$

and fusion defined by **mixing and crossfade** of underlying audio patterns.

Remark

This model is actually used for audio pattern reconstruction and control live-looping performance [BJM12].

Induced algebra: triples of words



Triples of words : $T_A = 0 + A^* \times A^* \times A^*$ with product:

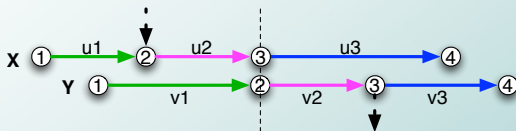
$$\underbrace{(u_1, u_2, u_3)}_X \cdot \underbrace{(v_1, v_2, v_3)}_Y = \underbrace{((u_1 u_2 \vee_s v_1) u_2^{-1}, u_2 v_2, v_2^{-1} (u_3 \vee_p v_2 v_3))}_{X.Y}$$

and fusion defined by **letter to letter unification** and 0 if incompatible.

with $u \vee_s v = \min_s \{w \in A^* : u \leq_s w, v \leq_s w\}$ and

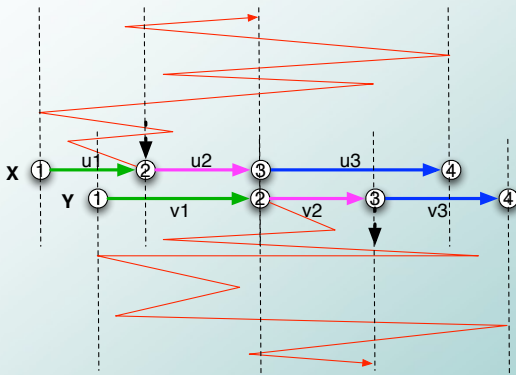
$u \vee_p v = \min_p \{w \in A^* : u \leq_p w, v \leq_p w\}$.

Product examples



- $(a, b, c).(b, c, d) = (a, bc, d)$,
- $(a, b, c).(da, 1, bcd) = (da, b, cd)$,
- $(a, b, c).(a, b, c) = 0$.

A link with 2W-automata run ?



Remark

Overlapping tiles of words can be seen as *domains of partial runs of two way automata* [Péc85] with tiles product seen as *two-way automata domains composition*.

Resulting structures

Theorem

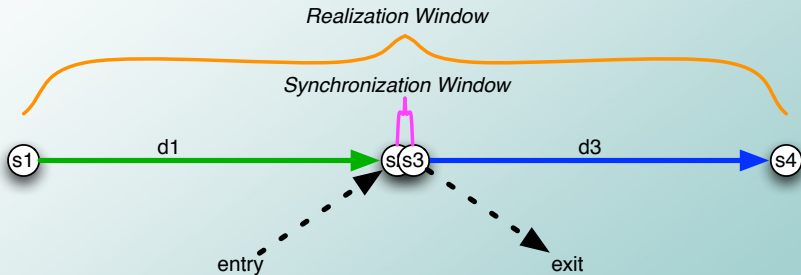
In all above cases, we obtain **monoids**, i.e. sets equipped with an **associative product** with a **neutral element**.

3. Properties

Monoids with remarkable properties. . .

Context elements

When input and output sync match, i.e. with empty sync window, elements are called **contexts**



Properties of context elements

Lemma

Context elements are **idempotents**, i.e. for all context element E ,
 $E.E = E$.

Lemma

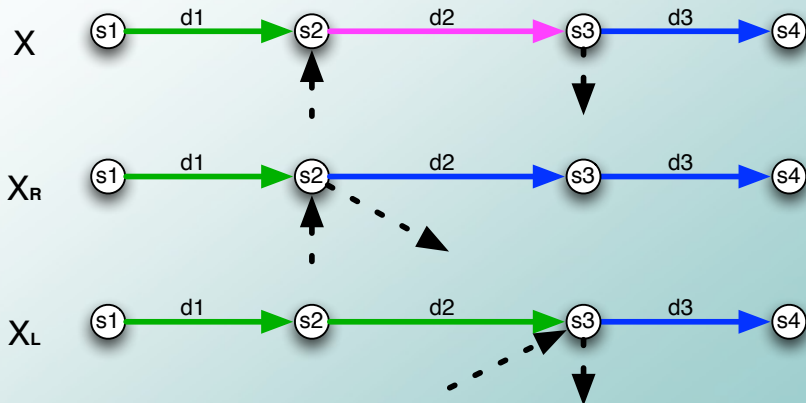
Context elements **commute**, i.e. for all context elements E and F ,
 $E.F = F.E$.

Corollary

The set U of context elements is a **\wedge -semilattice** with, for all E
and $F \in U$, $E \wedge F = E.F$.

Left and right canonical contexts

Every element is **structurally** associated to a **right context** and a **left context**:



Properties of canonical context elements

Given the left and right context operators $X \mapsto X_L$ and $X \mapsto X_R$:

Lemma

for all patterns X and Y :

- (0) $(X_L)_R = (X_L)_L = X_L$ and $(Y_R)_R = (Y_R)_L = Y_R$ (*projection*),
- (1) $X_L X_L = X_L$ and $Y_R Y_R = Y_R$ (*idempotence*),
- (2) $X_L Y_L = Y_L X_L$ and $X_R Y_R = Y_R X_R$ (*commutation*),
- (3) $X_R X = X X_L = X$ (*local neutrality*),
- (4) $X_R (XY)_R = (XY)_R$ (resp. $(XY)_L Y_L = (XY)_L$) (*\mathcal{L} and \mathcal{R} -compatibility*).

Natural order over patterns

Definition

$X \leq Y$ when $\text{sync}(X) = \text{sync}(Y)$ and $\text{real}(X) \supseteq \text{real}(Y)$.

Lemma

Relation \leq is an *order* relation, *stable* under product.

Moreover, for all patterns X and Y the following properties are equivalent:

- $X \leq Y$,
- $X = X_R \cdot Y \cdot X_L$,
- $X = E \cdot Y \cdot F$ for two context elements E and F .

i.e. bi-lateral version of Nambooripad's *natural order* [Nam80].

Syntax vs semantics

Left and right canonical context elements have a **semantics characterization** ;

Lemma

For all element X ,

- (1) $X_L = \min\{E \leq 1 : X.E = X\}$ (*canonical left stab.*),
- (2) $X_R = \min\{E \leq 1 : E.X = X\}$ (*canonical right stab.*).

*Also, context elements can be defined as the set of **sub-units** $U = \{X : X \leq 1\}$.*

Remark

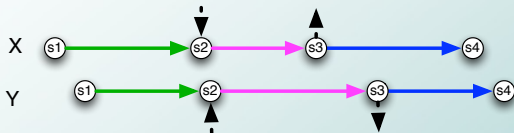
The semigroup version of these structures are already known, in inverse semigroup theory, as Lawson's **(stable) U -semiadequate monoids** [Law91].

4. Modeling

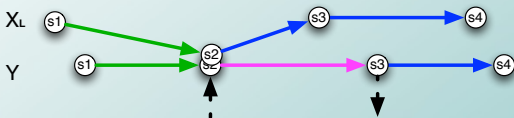
Sequential or parallel product ? Inverse elements ?

Induced partial parallelism : fork and join

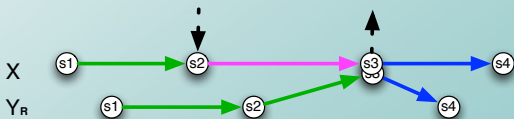
- Given two patterns X and Y :



- Fork:** start two patterns X and Y at the same time : $X_L.Y$,

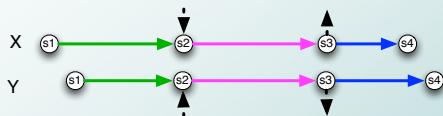


- Join:** Stop two patterns X and Y at the same time : $X.Y_R$.



Parallel composition

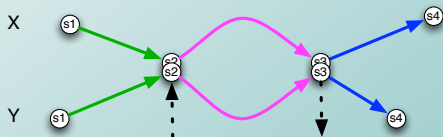
When $\text{sync}(X) = \text{sync}(Y)$,



we define:

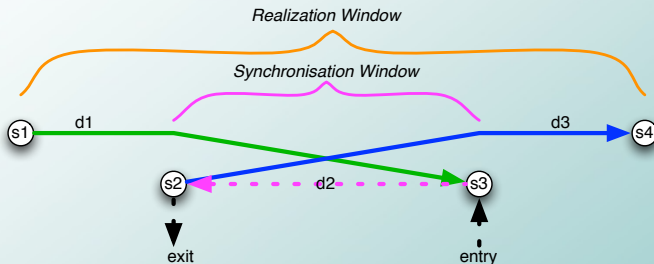
$$X \parallel Y = X_R.Y = Y_R.X = X.Y_L = Y.X_L$$

or, in picture:



Inverse patterns

Completing the set of models with backward sync window:



Though this may not have meaning in music, it makes not difficulty as a model of backward two-way automata partial run.

Induced inverse monoid

Lemma

For all pattern X there is a **unique** pattern X^{-1} (just switching input and output of X) such that

$$XX^{-1}X = X \text{ et } X^{-1}XX^{-1} = X^{-1}$$

with $X_L = X^{-1}X$ and $X_R = XX^{-1}$, i.e. the resulting (completed) monoid is an **inverse monoid** [Law98a].

Theorem

The monoid of tiles completed with inverse tiles is isomorphic to McAlister monoid [Law98b].

5. Langages

Langages of positives tiles

Classical classes of langages

Definition

For all langage $L \subseteq T_A = 0 + A^* \times A^* \times A^*$:

- L is **REC** when $L = \varphi^{-1}(\varphi(L))$ for some morphism $\varphi : T_A \rightarrow S$ and finite monoid S ,
- L is **RAT_R** resp. **RAT_C** when L is definable by means of finite sets, product, star and (R) residuals or (C) canonical context operators,
- L is **MSO** when L is definable by means of an MSO formulae.

Theorem ([Jan12a])

$$REC \subset\subset RAT_R \stackrel{?}{\subseteq} RAT_C = MSO$$

Classical classes of langages

Definition

For all langage $L \subseteq T_A = 0 + A^* \times A^* \times A^*$:

- L is **REC** when $L = \varphi^{-1}(\varphi(L))$ for some morphism $\varphi : T_A \rightarrow S$ and finite monoid S ,
- L is **RAT_R** resp. **RAT_C** when L is definable by means of finite sets, product, star and (R) residuals or (C) canonical context operators,
- L is **MSO** when L is definable by means of an MSO formulae.

Theorem ([Jan12a])

$$REC \subset\subset RAT_R \stackrel{\subset?}{\subseteq} RAT_C = MSO$$

Tile languages vs word languages

Theorem ([Jan12a])

Language $L \subseteq T_A - 0$ is MSO definable if and only if

$$L = \sum_{i \in I} (L_i \times C_i \times R_i)$$

with *finite* I and, for all $i \in I$, *regular* L_i , C_i and $R_i \subseteq A^*$.

Remark

Class MSO is thus both *simple* (above Theorem) and *robust* (previous theorem).

Tile languages vs word languages

Theorem ([Jan12a])

A language $L \subseteq T_A - 0$ is REC if and only if

$$L = \sum_{i \in I} (L_i \times C_i \times R_i)$$

with *finite* I and, for all $i \in I$, finite or co-finite $L_i \subseteq \text{Suff}({}^\omega(x_i y_i))$, $C_i \subseteq x_i y_i^k (x_i y_i)^*$ and $R_i \subseteq \text{Pref}((x_i y_i)^\omega)$ for some given words x_i and $y_i \in A^*$.

Remark

The situation is even worse than for languages recognizable by inverse monoids [MP84] or recognizable subsets of $FIM(A)$ [Sil96].

6. Quasi-rec.

A remedy to the collapse of REC

Why REC collapse ?

Lemma

Let morphism $\varphi : T_A \rightarrow S$ and let $x = (1, u, 1)$ and $y = (1, v, 1) \in T_A$ such that $\varphi(u) = \varphi(v) \neq 0$. Then u and v are ordered both under prefix and suffix order.

Proof.

We have $x_{RX} = x$ hence $\varphi(x_{RX}) = \varphi(x_R)\varphi(x) \neq 0$ hence $\varphi(x_R)\varphi(y) = \varphi(x_{RY}) \neq 0$ hence $x_{RY} \neq 0$.

But then $x_{RY} = (1, 1, u)(1, v, 1) = (1, v, v^{-1}(v \vee_p u))$ and thus $v \vee_p u \neq 0$.



Relaxing morphism to premorphism

Over (stable) ordered monoids:

Definition ([MR77])

Mapping $\varphi : (M, \leq) \rightarrow (N, \leq)$ is a **premorphisme** when φ is **monotonic**, and for all x and $y \in M$, $\varphi(xy) \leq \varphi(x)\varphi(y)$.

Definition ([Jan12c])

Language $L \subseteq T_A$ is **quasi-recognizable** (QREC) when $L = \varphi^{-1}(\varphi(L))$ for some **premorphisme** $\varphi : T_A \rightarrow N$ and finite (stable ordered) monoid N .

Tuning for $QREC \subseteq MSO$

Remark

At first sight MSO definability is lost.

Definability of $\varphi^{-1}(xy)$ in terms of $\varphi^{-1}(x)$ and $\varphi^{-1}(y)$?

Theorem ([Jan12c])

We can restrict to some subclass of quasi-inverse monoids such that $QREC \subseteq MSO$.

Hint: given $\varphi : T_A \rightarrow S$, assume that, in S ,

$$x = x_R \widehat{x} x_L$$

for every $x \in S$ with

$$\varphi((u_1, u_2, u_3)) = \varphi((1, u_1, 1))_R \varphi((1, u_2, 1)) \varphi((1, u_3, 1))_L$$

Checking when $MSO \subseteq QREC$

Theorem ([Jan12c])

Under coherent context closure (CCC) condition $MSO \subseteq QREC$.

Remark

Given a premorphism $\varphi : T_A \rightarrow S$ with quasi-inverse S , for all x and y , if $x\mathcal{R}y$ then $x_R = y_R$, if $x\mathcal{L}y$ then $x_L = y_L$.

Hence for all $u = (u_1, r, u_2)$ and $v = (v_1, r, v_2) \in T_A$ if $\varphi(u_1)\mathcal{L}\varphi(v_1)$ and $\varphi(u_3)\mathcal{R}\varphi(v_3)$ then $\varphi(u) = \varphi(v)$.

Word congruences associated to tile languages

Given $L \subseteq T_A - 0$ let \simeq_L the relation over **words** defined by $u \simeq_L v$ when u and v can be interchanged in triples of L preserving membership.

Lemma

L is MSO definable if and only if \simeq_L is of finite index.

Question

Can we build a premorphism recognizing L shifting up
 $\varphi : A^* \rightarrow A^* / \simeq_L$?

The quasi-inverse expansion

Definition

Let S be a monoid with \mathcal{L}_S and \mathcal{R}_S the monoid of left and right ideals.

Let $Q(S) = \mathcal{L}_S \times S \times \mathcal{R}_S + 0$ with product defined by

$$(P, x, Q) \cdot (L, y, R) = (P \cap (L)x^{-1}, xy, y^{-1}(Q) \cap Y)$$

when non empty context and zero otherwise.

One can check that $T_A \subseteq Q(A^*)$.

Lemma ([Jan12c])

Mapping $Q(\varphi) : T_A \rightarrow Q(A^* / \simeq_L)$ defined by

$$Q(\varphi)((u_1, u_2, u_3)) = (S \cdot \varphi(u_1), \varphi(u_2), \varphi(u_3) \cdot S)$$

is a premorphism that, under CCC hypothesis, recognizes L .

A Birget-Rhodes expansion

In the category of quasi-inverse monoids and premorphisms:

Lemma ([Jan12b])

\mathcal{Q} is a natural transformation such that, for all monoid morphism $\varphi : M \rightarrow N$, there are surjective premorphisms $\sigma_M : \mathcal{Q}(M) \rightarrow M$ and $\sigma_N : \mathcal{Q}(N) \rightarrow N$ with commuting diagram

$$\begin{array}{ccc}
 \mathcal{Q}(M) & \xrightarrow{\mathcal{Q}(\varphi)} & \mathcal{Q}(N) \\
 \downarrow \sigma_M & & \downarrow \sigma_N \\
 M^0 & \xrightarrow{\varphi} & N^0
 \end{array}$$

and $T_A \subseteq \mathcal{Q}(A^*)$.

7. Conclusion

- Sémantique «algébrique» pour les 2-way automata ?
- Langages de tuilages temporisés ?
- Théorie algébrique des langages d'arbres ?
- Développement en théorie des semigroupes ?
- Alternative aux calculs de processus... ?



F. Berthaut, D. Janin, and B. Martin.

Advanced synchronization of audio or symbolic musical patterns.

Technical Report RR1461-12, LaBRI, Université de Bordeaux, 2012.



P. Desain and H. Honing.

Loco: a composition microworld in logo.

Computer Music Journal, 12(3):30–42, 1988.



D. Janin.

Vers une modélisation combinatoire des structures rythmiques simples.

Technical Report RR-1455-11, LaBRI, Université de Bordeaux, August 2011.



D. Janin.

On languages of one-dimensional overlapping tiles.

Technical Report RR-1457-12, LaBRI, Université de Bordeaux, January 2012.



D. Janin.

Quasi-inverse monoids (and premorphisms).

Technical Report RR-1459-12, LaBRI, Université de Bordeaux,
March 2012.



D. Janin.

Quasi-recognizable vs MSO definable languages of
one-dimensionnal overlapping tiles.

Technical Report RR-1458-12, LaBRI, Université de Bordeaux,
February 2012.



Mark V. Lawson.

Semigroups and ordered categories. i. the reduced case.

Journal of Algebra, 141(2):422 – 462, 1991.



Mark V. Lawson.

Inverse Semigroups : The theory of partial symmetries.

World Scientific, 1998.



Mark V. Lawson.

McAlister semigroups.

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



Stuart W. Margolis and Jean-Eric Pin.

Languages and inverse semigroups.

In *ICALP*, volume 172 of *Lecture Notes in Computer Science*, pages 337–346. Springer, 1984.



D.B. McAlister and N. R. Reilly.

E-unitary convers for inverse semigroups.

Pacific Journal of Mathematics, 68:178–206, 1977.



K. S. S. Nambooripad.

The natural partial order on a regular semigroup.

Proc. Edinburgh Math. Soc., 23:249–260, 1980.



Jean-Pierre Pécuchet.

Automates boustrophedon, semi-groupe de birget et monoïde inversif libre.

ITA, 19(1):71–100, 1985.



Pedro V. Silva.

On free inverse monoid languages.

ITA, 30(4):349–378, 1996.