

Musique computationnelle : Des arabesques de Debussy aux monoïdes quasi-inversifs de Lawson

David Janin,
Université de Bordeaux
LaBRI

Séminaire MaMuX, 30 mars 2012

La musique est le seul langage qui permette de comprendre plusieurs personnes qui parlent en même temps. . .

Amadeus, XVIIIe siècle,

(d'après M. Forman dans le film éponyme)

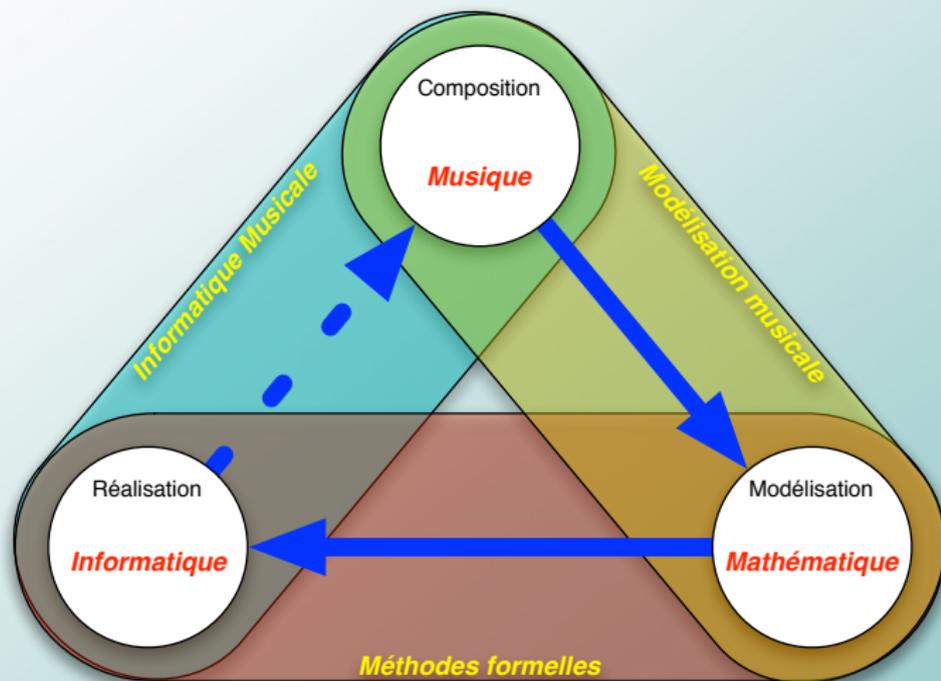
Aujourd'hui, deux siècles plus tard, on a bien compris le multiplexage fréquentiel, temporel, spatiale.

La musique est aussi, depuis des siècles, le langage du temps, de l'espace, du parallélisme, de l'interaction, . . .

La modélisation formelle du langage musical peut-elle nous conduire à de nouvelles *métaphores* ou *paradigmes* utiles à la modélisation des systèmes informatique ?

Research field :

Theoretical foundation of computational music



1. Arabesques

La structure complexe des arabesques...

Un exemple initiatique

The image displays two systems of musical notation for the piece 'Arabesques' by Maurice Ravel. The first system consists of a piano (p) part on the left and a violin part on the right. The piano part features a melodic line with triplets and slurs, while the violin part provides a harmonic accompaniment with sustained notes and slurs. The second system continues the musical development, showing the piano part with more complex rhythmic patterns and the violin part with sustained chords and melodic fragments. The tempo is marked 'a Tempo' at the beginning of the first system.

Extrait des arabesques : *andantino con moto*

Play

The image displays a musical score for 'Arabesques' by Debussy. The score is in G major and 4/4 time. The tempo is marked 'a Tempo'. The piano part features triplet patterns. The soprano line is marked with 'p' and features a series of motifs highlighted by red boxes. The score is presented in two systems, each with a soprano line and a piano accompaniment line.

une **succession de motifs** avec entrelacs, en voix de soprane

Play

The image displays a musical score for 'Arabesques' by Debussy, featuring two systems of staves. The top system includes a treble clef staff with the tempo marking 'a Tempo' and a piano staff. The bottom system includes a treble clef staff and a bass clef staff. The score is marked with 'p' (piano) and contains various musical notations such as triplets and slurs. A red box highlights a specific melodic line in the treble clef of the top system, which is further emphasized by an orange box. This highlighted line represents the 'trame sous-jacente' (underlying structure) mentioned in the text below.

une **trame sous-jacente** dans les médiums,

Play

The image displays two systems of musical notation for Chopin's Arabesque No. 1. The top system features a treble clef staff with a melodic line and a bass clef staff with a piano accompaniment. The bottom system shows a different section of the piece. In both systems, the bass clef staff contains arpeggiated chords, which are highlighted with green boxes. The melodic lines in both systems are highlighted with red boxes. The tempo marking 'a Tempo' is visible at the beginning of the first system. The piano marking 'p' is also present in the first system.

des arpegges pour l'harmonie,

Play

The image displays a musical score for 'Arabesques' in G major (one sharp) and 4/4 time. The score is divided into two systems. The first system consists of two staves: a treble clef staff and a bass clef staff. The tempo is marked 'a Tempo' and the dynamics are marked 'p'. The first system includes a triplet of eighth notes in the treble staff and a triplet of eighth notes in the bass staff. The second system also consists of two staves. The treble staff contains a melodic line with a long note, and the bass staff contains a rhythmic accompaniment. Annotations include red boxes highlighting specific melodic phrases in the treble staff and green boxes highlighting corresponding phrases in the bass staff. A blue horizontal line is drawn across the bass staff in the second system, and a yellow horizontal line is drawn across the treble staff in the first system.

une ligne de basse...

Play

Quel langage de modélisation ?

Un langage pour l'analyse

partition \implies *structure* \implies *intention*

Un langage pour la composition (ou programmation musicale)

partition \longleftarrow *structure* \longleftarrow *intention*

Un langage expressif

doté d'opérateurs de compositions séries, parallèles,
avec *recouvrements partiels*, et descriptions hiérarchiques. . .

Un langage utilisable

doté de bonnes propriétés informatique telles que la
compositionnalité, la *réutilisabilité*, etc. . .

Quel langage de modélisation ?

Un langage pour l'analyse

partition \implies *structure* \implies *intention*

Un langage pour la composition (ou programmation musicale)

partition \longleftarrow *structure* \longleftarrow *intention*

Un langage expressif

doté d'opérateurs de compositions séries, parallèles,
avec recouvrements partiels, et descriptions hiérarchiques...

Un langage utilisable

doté de bonnes propriétés informatique telles que la
compositionnalité, la réutilisabilité, etc...

Quel langage de modélisation ?

Un langage pour l'analyse

partition \implies *structure* \implies *intention*

Un langage pour la composition (ou programmation musicale)

partition \longleftarrow *structure* \longleftarrow *intention*

Un langage expressif

doté d'opérateurs de compositions séries, parallèles,
avec recouvrements partiels, et descriptions hiérarchiques...

Un langage utilisable

doté de bonnes propriétés informatique telles que la
compositionnalité, la réutilisabilité, etc...

Quel langage de modélisation ?

Un langage pour l'analyse

partition \implies *structure* \implies *intention*

Un langage pour la composition (ou programmation musicale)

partition \longleftarrow *structure* \longleftarrow *intention*

Un langage expressif

doté d'opérateurs de compositions **séries**, **parallèles**,
avec **recouvrements partiels**, et **descriptions hiérarchiques**...

Un langage utilisable

doté de bonnes propriétés informatique telles que la
compositionnalité, la réutilisabilité, etc. . .

Quel langage de modélisation ?

Un langage pour l'analyse

$partition \implies structure \implies intention$

Un langage pour la composition (ou programmation musicale)

$partition \longleftarrow structure \longleftarrow intention$

Un langage expressif

doté d'opérateurs de compositions **séries**, **parallèles**,
avec **recouvrements partiels**, et **descriptions hiérarchiques**...

Un langage utilisable

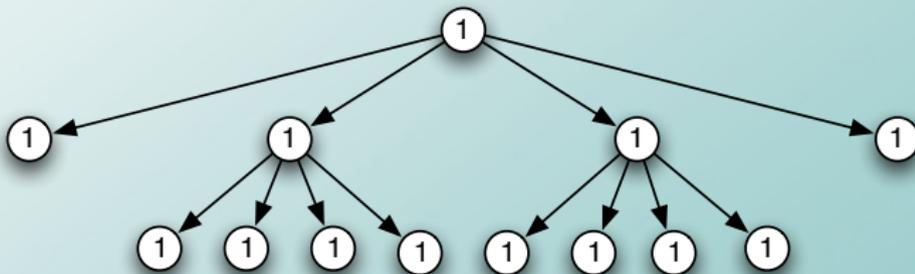
doté de bonnes propriétés informatique telles que la
compositionnalité, la **réutilisabilité**, etc. . .

Quel modèle pour ce langage ?

Un exemple de modélisation: la (structure rythmique de la) partition



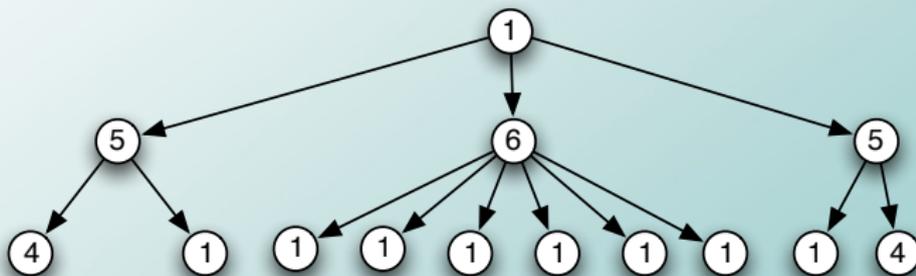
est (typiquement) modélisée à l'aide d'un arbre pondéré par les durées (relatives) de ses sous-arbres :



Le même motif



peut aussi se coder



en rendant compte du *phrasé musical* c'est à dire sa *structure* .

Quel modèle ?

Les arbres pondérés ?

pas si sûr...

Quel modèle ?

Les arbres pondérés ?

pas si sûr...

2. Composition Séquentielle

Le problème du Bebop...
et la solution du Bebop...

My little blue suede shoes (Ch. Parker)



The image displays two staves of musical notation in G-flat major (two flats) and common time (C). The first staff contains three measures of music. The first two measures are marked with a red '(a)' and are connected by a slur, representing the motif. The third measure is a whole rest. The second staff contains four measures. The first measure is marked with a red '(a)' and is connected by a slur to the second measure, which is marked with a red '(b)'. The third and fourth measures of the second staff are also connected by a slur, representing the conclusive variation.

Analyse

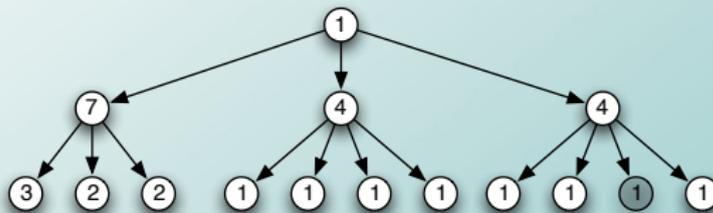
Trois expositions d'un même motif (a) suivi d'une variation conclusive (b).

Play

Modélisation des motifs (b)

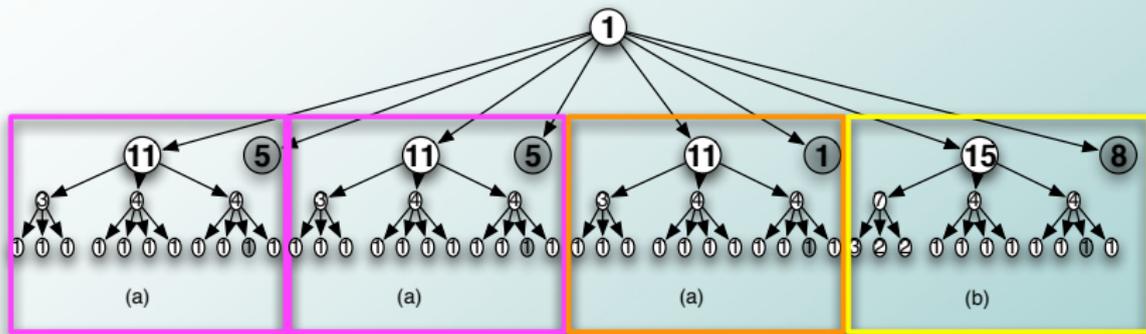


modélisé par:



(b)

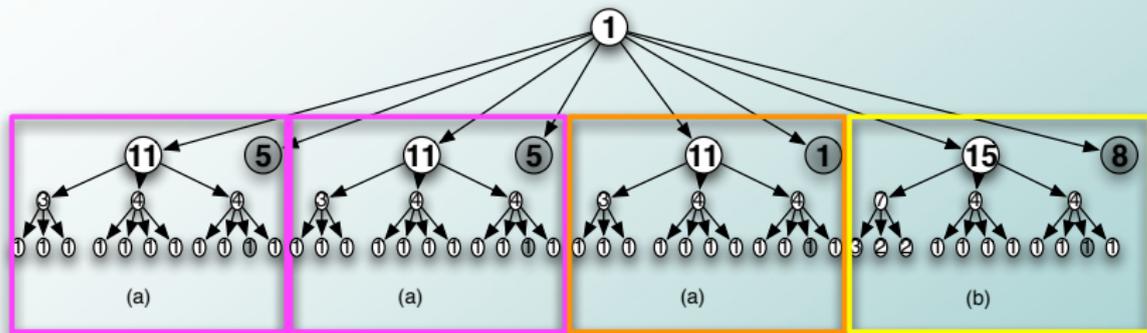
Modélisation résultante



Défauts de cette modélisation:

- insertion de silences de longueurs variables,
- structuration logique $(3 \times (a) + (b))$ perdue.

Modélisation résultante



Défauts de cette modélisation:

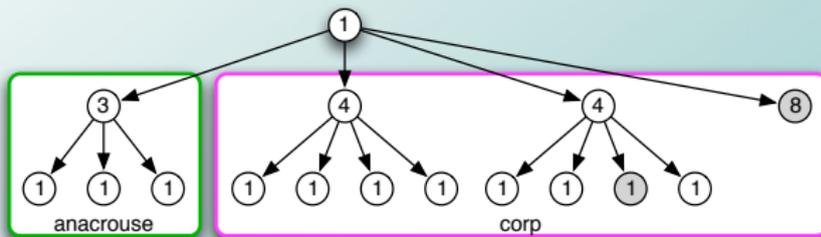
- insertion de silences de longueurs variables,
- structuration logique ($3 \times (a) + (b)$) perdue.

Autre approche

Expliciter l'anticipation du premier temps (anacrouse) :



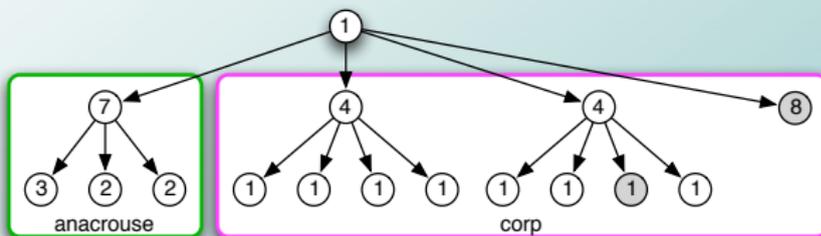
et la synchronisation implicite jusqu'à la fin de la seconde mesure :



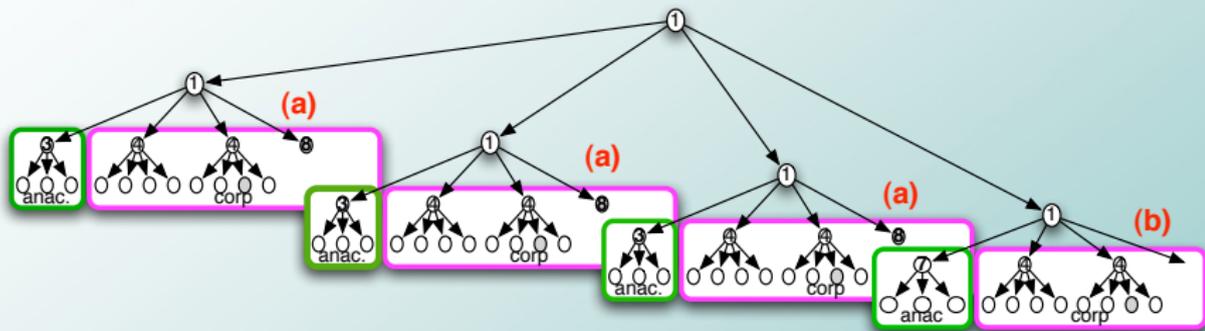
idem pour le deuxième motif le deuxième motif :



ce qui donne:



avec la *composition séquentielle* résultante:



définit avec *superpositions locales* des anacrouses avec les corps des motifs qui précèdent.

On retrouve ici notre structure logique : $3 \times (a) + (b)$!

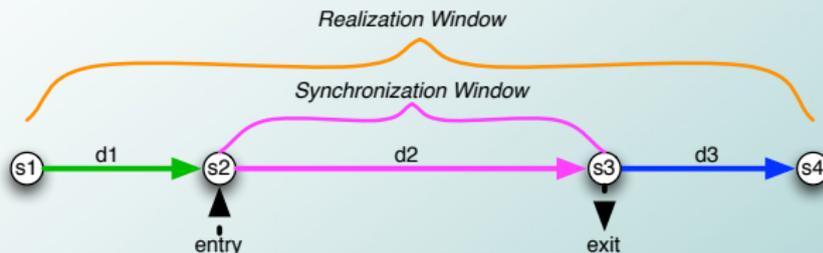
3. Formalisation

Une structuration algébrique du temps...

Fenêtres de synchronisation vs fenêtres de réalisation

Principe [Jan11]

Distinguer pour chaque motif musical:



Idée ancienne

Implicitement présente en modélisation musicale dans LOCO [DH88], mais aussi en productique ou en système d'exploitation.

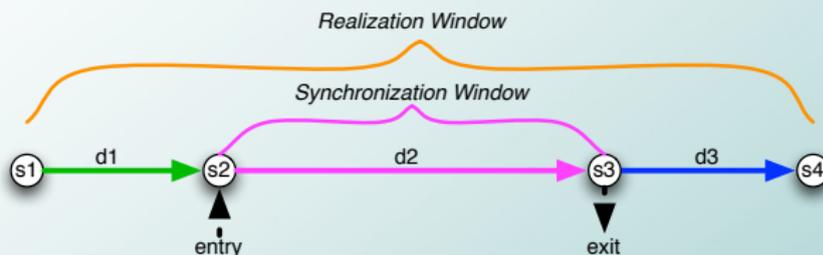
Remarque

En théorie de la musique : cette distinction semble modéliser et généraliser la notion d'*appui* (idée à développer ?).

Fenêtres de synchronisation vs fenêtres de réalisation

Principe [Jan11]

Distinguer pour chaque motif musical:



Idée ancienne

Implicitement présente en modélisation musicale dans LOCO [DH88], mais aussi en **productique** ou en **système d'exploitation**.

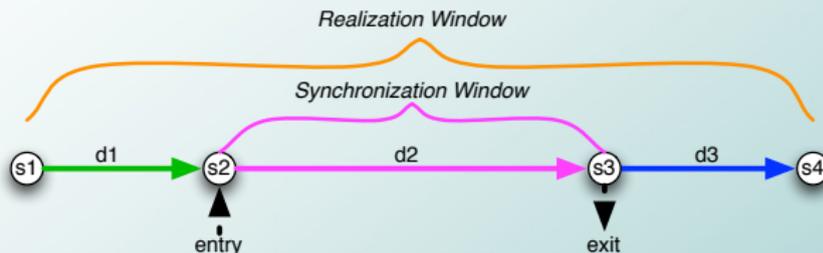
Remarque

En théorie de la musique : cette distinction semble modéliser et généraliser la notion d'*appuis* (idée à développer ?).

Fenêtres de synchronisation vs fenêtres de réalisation

Principe [Jan11]

Distinguer pour chaque motif musical:



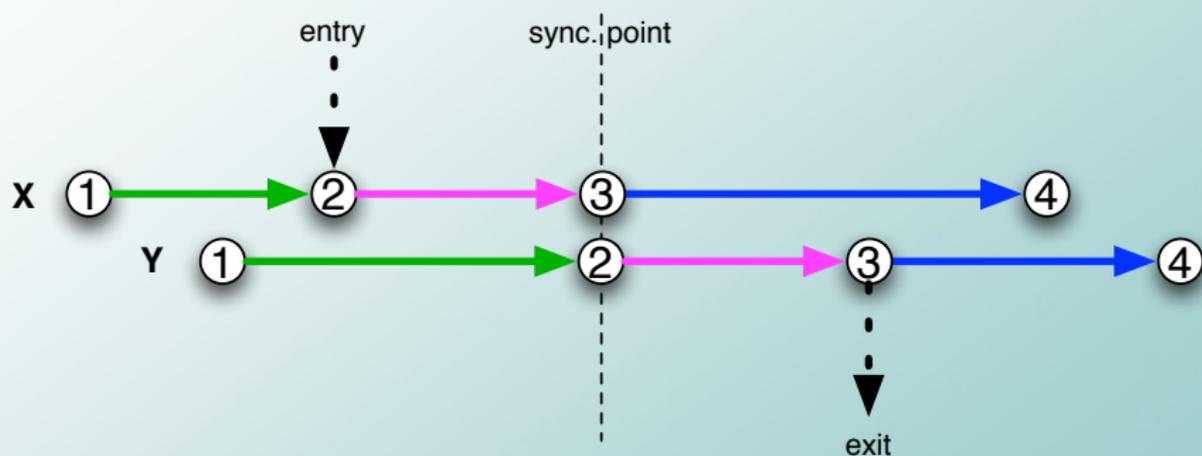
Idée ancienne

Implicitement présente en modélisation musicale dans LOCO [DH88], mais aussi en **productique** ou en **système d'exploitation**.

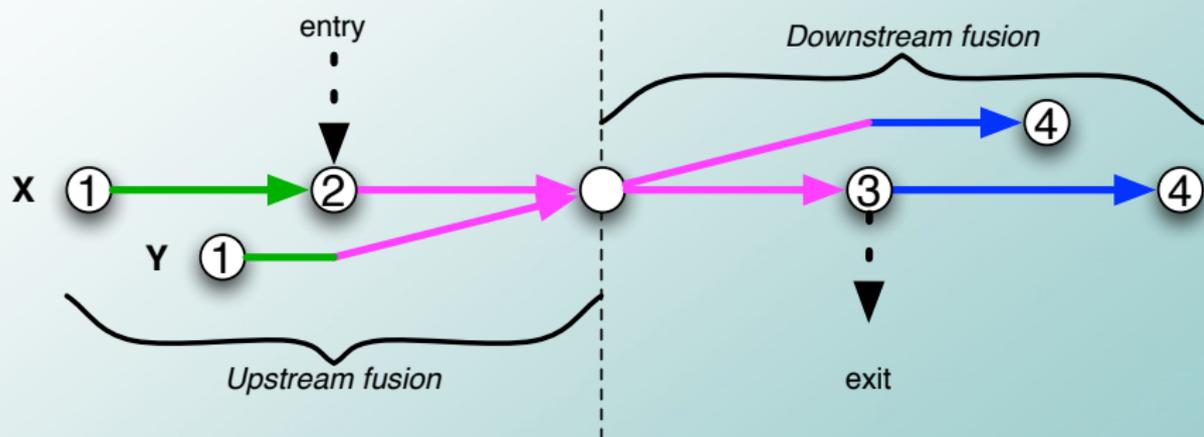
Remarque

En théorie de la musique : cette distinction semble modéliser et généraliser la notion d'*appuis* (idée à développer ?).

Produit séquentiel : 1. synchronisation



Produit séquentiel : 2. fusion



Algèbre induite : le cas discret

Triplets de mots : $T_A = 0 + A^* \times A^* \times A^*$ équipé du produit:

$$(u_1, u_2, u_3) \cdot (v_1, v_2, v_3) = ((u_1 u_2 \vee_s v_1) u_2^{-1}, u_2 v_2, v_2^{-1} (u_3 \vee_p v_2 v_3))$$

avec fusion défini par **unification lettre à lettre** (0 si incompatible)
avec

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

et

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

Exemple : $(ab, c, de) \cdot (b, c, d) = (ab, cd, e)$.

Algèbre induite : le cas continu

Triplets de durées : $D_A = \mathbb{R} \times \mathbb{R} \times \mathbb{R}$, équipé du produit:

$$(x_1, x_2, x_3) \cdot (y_1, y_2, y_3) = (\max(x_1, y_1 - x_2), x_2 + y_2, \max(y_3, x_3 - y_2))$$

avec fusion défini sur les motifs audio sous-jacents par **mixage et crossfading**.

Structure algébrique résultante

Théoreme

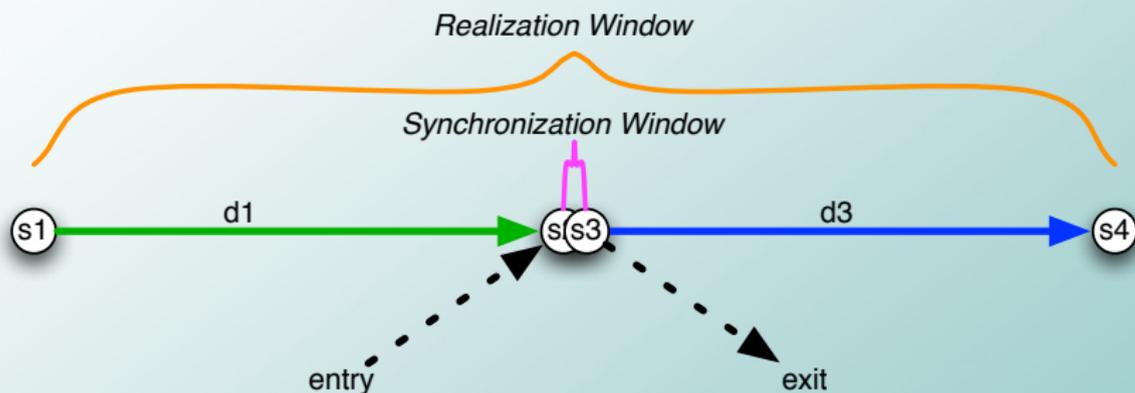
Dans tous les cas, on obtient des **monoïdes**, i.e. des ensembles équipés d'un **produit associatif** avec un **élément neutre**.

4. Monoïde quasi-inversif

Une structure algébrique particulièrement riche...

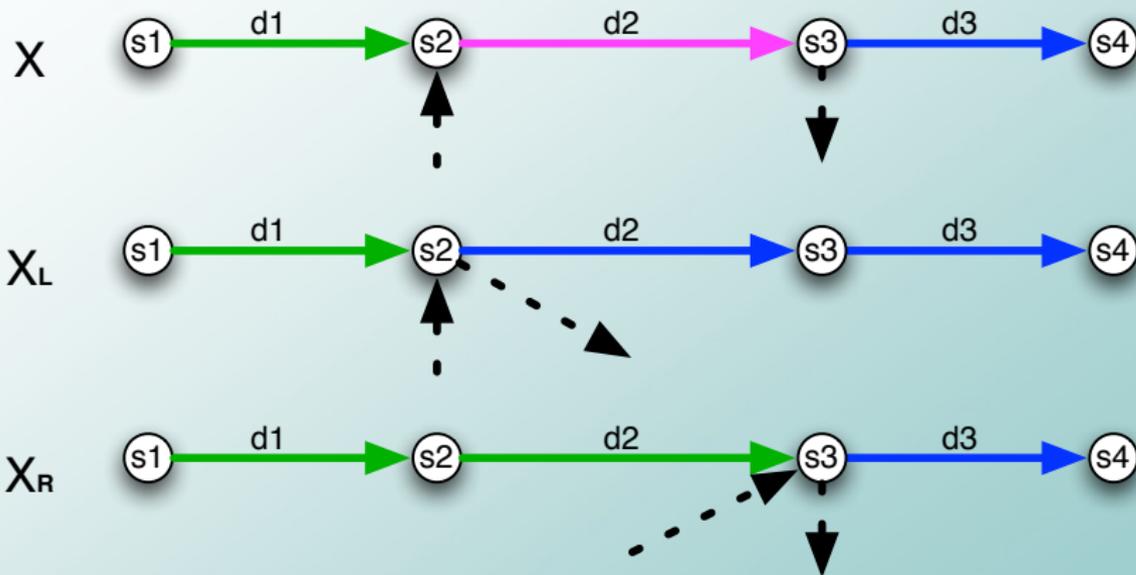
Les motifs contextuels

Lorsqu'entrée et sortie coïncident, on obtient des **contextes**



Contextes droits et gauches d'un élément

Chaque élément est **structurellement** associé à un **contexte droit** et un **contexte gauche**:



Propriétés des contextes

Lemma

Pour tout motif X et Y :

- (0) $(X_L)_R = (X_L)_L = X_L$ et $(Y_R)_R = (Y_R)_L = Y_R$ (*projection*),
- (1) $X_L X_L = X_L$ et $Y_R Y_R = Y_R$ (*idempotence*),
- (2) $X_L Y_L = Y_L X_L$ et $X_R Y_R = Y_R X_R$ (*commutation*),
- (3) $X_R X = X X_L = X$ (*neutralité locale*).

Ordre naturel sur les motifs

Definition

$X \leq Y$ lorsque $\text{sync}(X) = \text{sync}(Y)$ et $\text{real}(X) \supseteq \text{real}(Y)$.

Lemma

Pour tout X et Y on a équivalence de :

- $X \leq Y$,
- $X = X_R Y X_L$,
- $X = E Y F$ pour E et F deux contextes.

i.e. version bi-lateral de l'ordre naturel de Nambooripad [Nam80].

Syntaxe vs sémantique

Les contextes droits et gauches ont aussi une définition **sémantique**;

Lemma

Pour tout élément X ,

$$(1) X_L = \min\{E \leq 1 : XE = X\} \text{ (stab. droit canonique),}$$

$$(2) X_R = \min\{E \leq 1 : EX = X\} \text{ (stab. gauche canonique).}$$

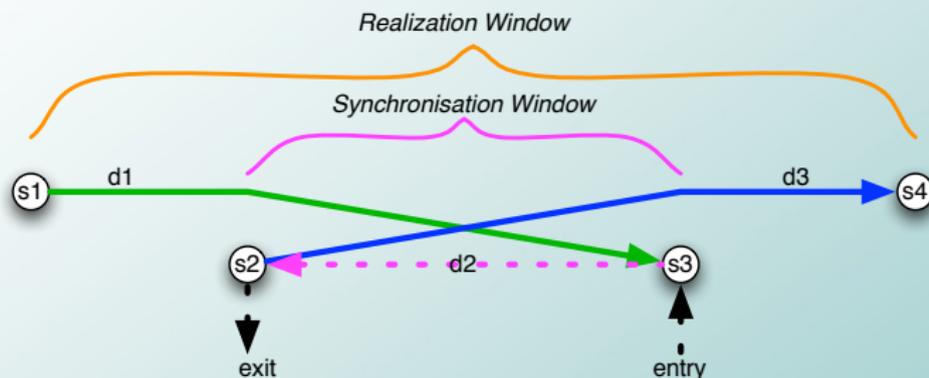
En particulier, l'ensemble des contextes peut-être redéfinit par $U = \{X : X \leq 1\}$ l'ensemble des **sous-unités**.

Remarque

Ces structures (sans neutre) existent déjà dans la théorie des semigroupes inversifs [Law98] : c'est les semigroupes **U -semiadéquat** de Lawson [Law91] appelé ici quasi-inverses.

Motif inverse

On peut aussi définir (sens (musical ?) des motifs avec synchronisation négative (inverse) :



Lemma

Pour tout motif X il existe un **unique** motif X^{-1} tel que

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

avec $X_L = X^{-1}X$ et $X_R = XX^{-1}$, i.e. c'est un **monoïde inversif**.

5. Langages

Où on est amené à revisiter la théorie algébrique des langages

Classes de langages classiques

Definition

Pour tout langage $L \subseteq T_A = 0 + A^* \times A^* \times A^*$ on a:

- L est **REC** lorsque $L = \varphi^{-1}(\varphi(L))$ pour $\varphi : T_A \rightarrow S$ morphisme de monoïde et S fini,
- L est **RAT_R** ou **RAT_C** lorsque L est définissable à l'aide d'ensemble fini, somme, produit, étoile et (R) opérateurs de résidus ou (C) opérateurs de contexte,
- L est **MSO** lorsque L est définissable à l'aide d'une formule de logique monadique du second ordre.

Theorem ([Jan12a])

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

Classes de langages classiques

Definition

Pour tout langage $L \subseteq T_A = 0 + A^* \times A^* \times A^*$ on a :

- L est **REC** lorsque $L = \varphi^{-1}(\varphi(L))$ pour $\varphi : T_A \rightarrow S$ morphisme de monoïde et S fini,
- L est **RAT_R** ou **RAT_C** lorsque L est définissable à l'aide d'ensemble fini, somme, produit, étoile et (R) opérateurs de résidus ou (C) opérateurs de contexte,
- L est **MSO** lorsque L est définissable à l'aide d'une formule de logique monadique du second ordre.

Theorem ([Jan12a])

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

Langages de tuiles vs langages de mots

Theorem ([Jan12a])

Un langage $L \subseteq T_A$ est MSO définissable si et seulement si

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

pour I fini, et pour tout $i \in I$, L_i , C_i et $R_i \subseteq A^*$ rationnels.

Remarque

La classe des langages définissables en MSO est donc simple (théorème ci dessus) et robuste (théorème précédent).

Remédier à l'effondrement de REC

Definition ([MR77])

$\varphi : (M, \leq) \rightarrow (N, \leq)$ est un **pré-morphisme** lorsque φ est **croissante**, et pour tout x et $y \in M$, $\varphi(xy) \leq \varphi(x)\varphi(y)$ (au lieu de $\varphi(xy) = \varphi(x)\varphi(y)$ pour les morphismes).

Definition ([Jan12c])

Un langage $L \subseteq M$ est **quasi-reconnaisable** lorsque $L = \varphi^{-1}(\varphi(L))$ pour un **premorphisme** $\varphi : M \rightarrow N$ avec N fini.

Résultat encourageant

Theorem ([Jan12c])

Dans les cas raisonnables, $QREC = MSO$.

Quasi-reconnaissabilité vs reconnaissabilité

Lemma ([Jan12b])

Il existe une transformation naturelle Q telle que, pour tout morphisme $\varphi : A^* \rightarrow S$, il existe des premorphismes surjectifs $\sigma_A : Q(A^*) \rightarrow A^*$ et $\sigma_S : Q(S) \rightarrow S$ tels que le diagramme suivant commute:

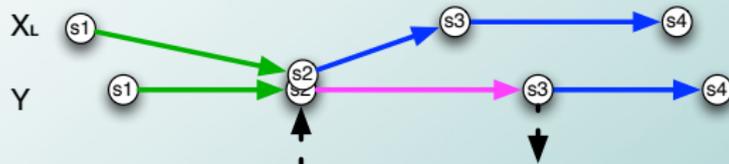
$$\begin{array}{ccc} T_A \subseteq Q(A^*) & \xrightarrow{Q(\varphi)} & Q(S) \\ \sigma_A \downarrow & & \downarrow \sigma_S \\ A^* & \xrightarrow{\varphi} & S \end{array}$$

6. Applications

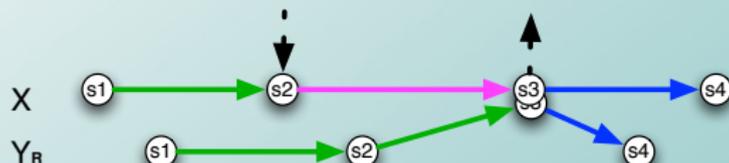
Amadeus II : the return...

Un parallélisme partiel : fork and join

- Démarrer deux motifs X et Y en même temps : $X_L Y$,

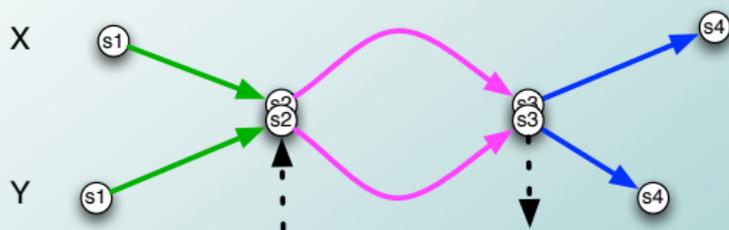


- Finir deux motifs X et Y en même temps : XY_R .



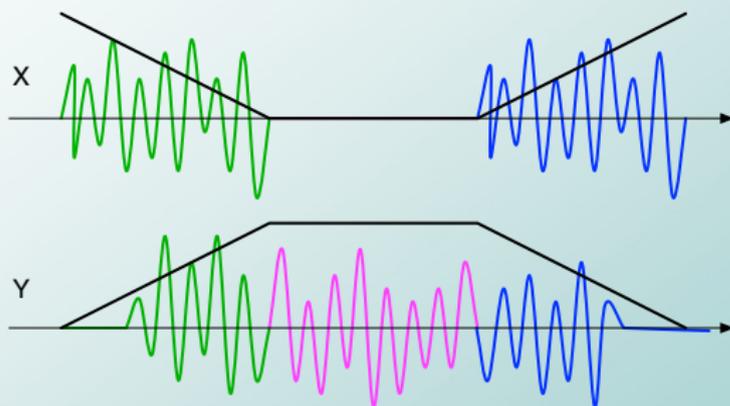
Composition parallèle

Nouvel opérateur



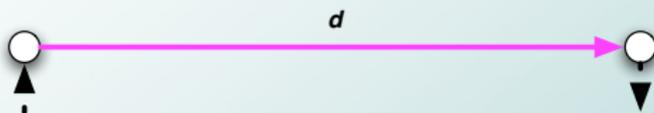
avec time-stretching pour ajuster les fenêtres de synchronisation.

Application à la reconstruction audio

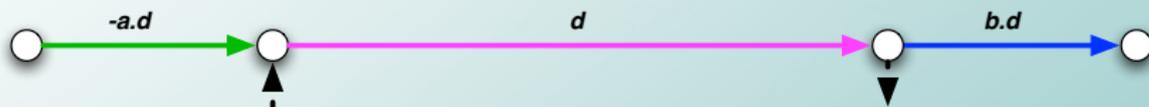


Resynchronization

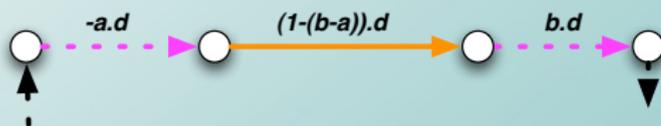
A partir d'un motif simple, déplacement des points de synchro:



Expansion : par offset à gauche $a < 0$ et à droite $b > 0$:

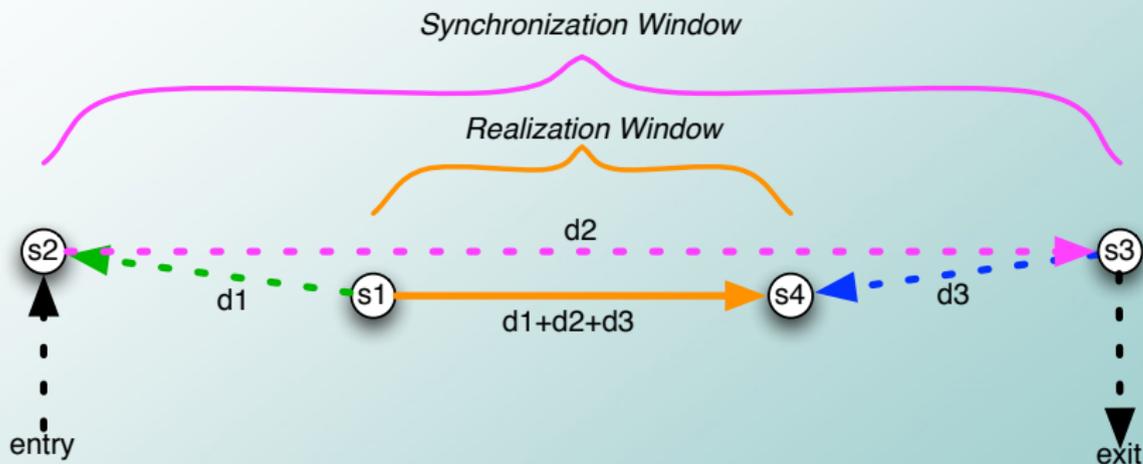


Contraction : par offset à droite $a > 0$ et à droite $b < 0$:

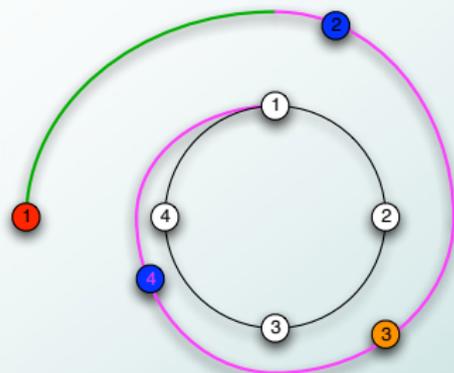


Extension du modèle de motifs

Après contraction, la synchronisation englobe la réalisation :

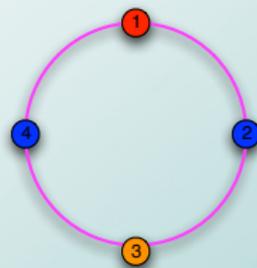


Application au live looping

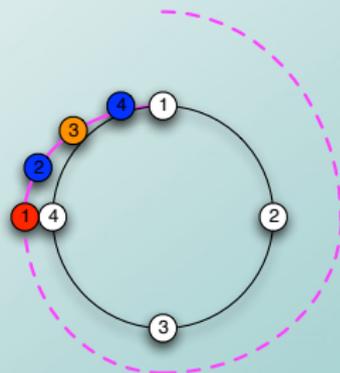


Expansion gauche (anticipation)

Play



Play



Contraction gauche (retard)

Play

Exemples rythmiques d'expansions et de contractions gauches...

7. Conclusion

- Des opérateurs de compositions prometteurs en musique, en algèbre et en informatique,
- Une extension au cas temporisée qui mériterait d'être étudié (avec O. Menini ?)
- Une extension au cas des arbres qui mériterait de même (avec P. Silva ?)

Un sujet de thèse **INEDIT** à pourvoir à la rentrée 2012 !

En vous remerciant de votre attention !



P. Desain and H. Honing.

Loco: a composition microworld in logo.

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



David Janin.

Modélisation compositionnelle des structures rythmiques : une exploration didactique.

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



David Janin.

On languages of one-dimensional overlapping tiles.

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



David Janin.

Quasi-inverse monoids.

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



David 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.



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

E-unitary converters 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.