MinMaxgd, A Toolbox to Handle Periodic Series in Semiring $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$.

ISTIA - LISA - University of Angers

L. Hardouin, B. Cottenceau, M. Lhommeau

July 1998 - June 2013

Chapitre 1

Introduction

This document presents a software toolbox. It aims to handle increasing pseudo-periodic series in semiring $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ introduced by the $(max, plus)$ team of INRIA Rocquencourt (see [Cohen, 1993]). The algorithms proposed in this software toolbox are initiated in 1992 in the PhD of S. Gaubert and continued in 1994 during the master of Benoit Gruet [Gruet, 1995]. It is still in evolution in order to be improved until today. The ancester of this software toolbox was "MAX" (voir `http:\\maxplus.org`), it was developed with Maple during the PhD of S. Gaubert [Gaubert, 1992]. The present software toolbox is developped in C++ language, it is based on an improvement of the algorithms proposed by S. Gaubert and an extension to some other operations. The C++ library can be interfaced to Scilab and more efficiently with Scicoslab. This toolbox and interfaces are downloadable in the following URL :
`http://istia.univ-angers.fr/~hardouin/outils.html`.
In this note we focus first on what is the objects considered, namely periodic series, then the algorithm issues are addressed. The last part is a part dedicated on how to use the C++ library.

Let us recall that a toolbox for (max,plus) calculus developed by INRIA Rocquencourt is also available in Scicoslab (see `http://www.scilab.org/contrib/`).
Contact :
ISTIA/LISA University of Angers,
62 Avenue Notre Dame du lac,
49000 Angers
`bertrand.cottenceau@univ-angers.fr`
`laurent.hardouin@univ-angers.fr`
`mehdi.lhommeau@univ-angers.fr`

Chapitre 2

Dioid $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$

This chapter is manly based on ([Cohen et al., 1989, Baccelli et al., 1992, Gaubert, 1992, Gruet, 1995, Cottenceau, 1999, Abeka, 2005]). Shortly the main facts about $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ are recalled.

2.1 Dioid $\mathbb{B}[\![\gamma, \delta]\!]$

Set of points in $\mathbb{Z}^2$ and its bi-dimensional representation are considered. The idea is to code each points by two variables γ and δ with exponents in $\mathbb{Z}$. These exponents represent the co-ordinates of the points, and the set of points is represented by a series of two variables.

Definition 1 *(Dioïde $\mathbb{B}[\![\gamma, \delta]\!]$) The dioid of formal power series with boolean coefficients and two variables γ and δ with exponents in $\mathbb{Z}$ is denoted $\mathbb{B}[\![\gamma, \delta]\!]$. A formal series of $\mathbb{B}[\![\gamma, \delta]\!]$ is written in an unique manner as follows :*

$$s = \bigoplus_{n,t \in \mathbb{Z}} s(n,t)\gamma^n \delta^t, \tag{2.1}$$

with $s(n,t) = e$ ou ε where e (respectively ε) is the unit element (respectively the zero element). $\mathbb{B}[\![\gamma, \delta]\!]$ is a complete semiring.

Definition 2 *(Support of a series s) The support of series s is defiend as a part of $\mathbb{Z}^2$ sucht that :*

$$Supp(s) = \{(n,t) \in \mathbb{Z}^2 \mid s(n,t) \neq \varepsilon\}$$

2.1.1 Graphical representation of the elements of $\mathbb{B}[\![\gamma, \delta]\!]$

A series $s \in \mathbb{B}[\![\gamma, \delta]\!]$ is depicted as a collection of points$(n,\ t)$ in $\mathbb{Z}^2$ belonging to the support of the series. Practically series $s = \gamma^2\delta^3 \oplus \gamma^3\delta^4 \oplus \gamma^5\delta^8 \oplus \gamma^6\delta^5\ \in \mathbb{B}[\![\gamma, \delta]\!]$ will be depicted by the points $(2,\ 3)$, $(3,\ 4)$, $(5,\ 8)$ et $(6,\ 5)$ de $\mathbb{Z}^2$. (see Figure 2.1).

2.2 Dioid $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$

In order to take the non decreasing specificity of trajectory into account, are considered only series which are invariant according to the product by γ^* (increasing according to the event) and the product by $(\delta^{-1})^*$ (increasing according to time). Hence, only increasing series of $\mathbb{B}[\![\gamma, \delta]\!]$ are considered, it is a subdioid denoted $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$.

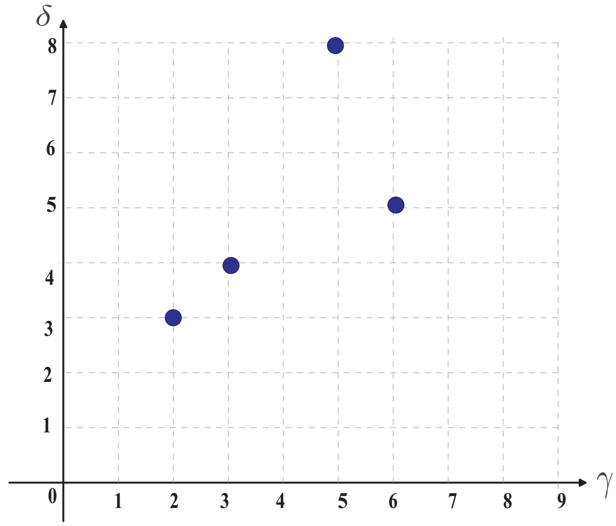

FIGURE 2.1 – Graphical representation of a series s in $\mathbb{B}[\![\gamma,\delta]\!]$.

Theorem 1

1. *$\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ is a dioid corresponding to a quotient dioid, by considering the congruence*

$$\{X_1(\gamma,\delta)\mathcal{R}_{(\gamma,\delta)}X_2(\gamma,\delta)\} \Longleftrightarrow \{\gamma^*(\delta^{-1})^*X_1(\gamma,\delta) = \gamma^*(\delta^{-1})^*X_2(\gamma,\delta)\}$$

2. *Each class of the quotient dioid $\mathbb{B}[\![\gamma,\delta]\!]_{/\mathcal{R}_{(\gamma,\delta)}}$ admits a greatest element in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$.*

Property 1 *Dioid $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ is a complete and distributive dioid with a neutral element for the $\oplus$ operator, namely $\varepsilon = \varepsilon(\gamma,\delta)$ (the null series in $\mathbb{B}[\![\gamma,\delta]\!]$) and a neutral element for the $\otimes$operator, namely $e = (\gamma \oplus \delta^{-1})^*$.*

Example 1 *Let s_1 et s_2 be series in $\mathbb{B}[\![\gamma,\delta]\!]$ defined as follows*

$$s_1 = \gamma^2\delta^3 \oplus \gamma^3\delta^2 \oplus \gamma^5\delta^6,$$

$$s_2 = \gamma^2\delta^3 \oplus \gamma^5\delta^6.$$

The computation of $(\gamma^*(\delta^{-1})^*)s_1$ is detailed below

$$\begin{array}{rcl}
(\gamma^*(\delta^{-1})^*)s_1 & = & (e \oplus \gamma^1 \oplus \gamma^2 \oplus \gamma^3 \oplus \ldots)(e \oplus \delta^{-1} \oplus \delta^{-2} \oplus \delta^{-3} \oplus \ldots)(\gamma^2\delta^3 \oplus \gamma^3\delta^2 \oplus \gamma^5\delta^6) \\
& = & (e \oplus \gamma^1\delta^{-1} \oplus \gamma^1\delta^{-2} \oplus \gamma^1\delta^{-3} \oplus \ldots \oplus \gamma^2\delta^{-1} \oplus \gamma^2\delta^{-2} \oplus \gamma^2\delta^{-3} \oplus \ldots)(\gamma^2\delta^3 \oplus \gamma^3\delta^2 \oplus \gamma^5\delta^6) \\
& = & (\gamma^*(\delta^{-1})^*)(\gamma^2\delta^3 \oplus \gamma^5\delta^6) \\
& = & (\gamma^*(\delta^{-1})^*)s_2.
\end{array}$$

Clearly, this leads to $(\gamma^*(\delta^{-1})^*)s_1 = (\gamma^*(\delta^{-1})^*)s_2 = (\gamma^*(\delta^{-1})^*)(\gamma^2\delta^3 \oplus \gamma^5\delta^6)$, hence series s_1 and s_2 belong to the same equivalence class in $\mathbb{B}[\![\gamma,\delta]\!]_{/\mathcal{R}_{(\gamma,\delta)}}$. Furthermore $(\gamma^*(\delta^{-1})^*)(\gamma^2\delta^3 \oplus \gamma^5\delta^6)$ is the greatest element of this class in $\mathbb{B}[\![\gamma,\delta]\!]$.

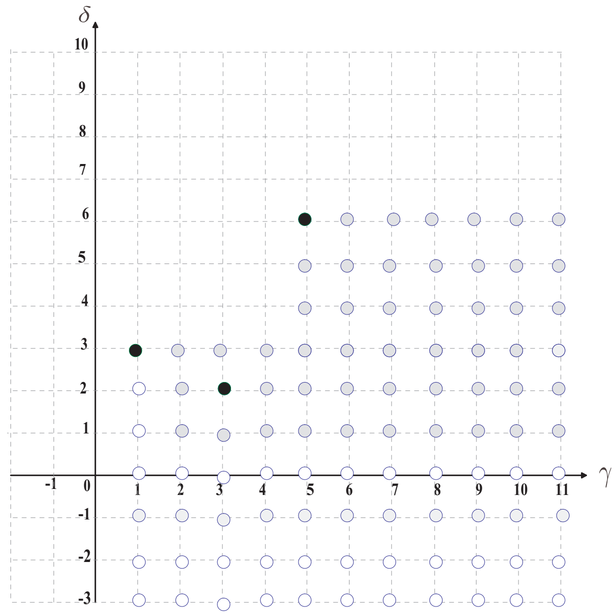

FIGURE 2.2 – Graphical representation of series $\bullet \equiv s_1$ $\circ \equiv (\gamma^*(\delta^{-1})^*)s_1$

2.2.1 Graphical representation of element in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$

The graphical representation of element of $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ is based on the one in $\mathbb{B}[\![\gamma,\delta]\!]$. Graphically, for monomial $\gamma^n\delta^t$, it is not the point (n,t) which is considered but the "south-east" cone having (n,t) as vertex. (see figure 2.2)

Maximal representative Dioids $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ and $\mathbb{B}[\![\gamma,\delta]\!]/\mathcal{R}_{(\gamma)^*(\delta^{-1})^*}$ are isomorphic. In other words, $\forall a \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ the following equality holds $a = (\gamma^*(\delta^{-1})^*)a$ and $(\gamma^*(\delta^{-1})^*)a$ is the maximal representative of the equivalence class of a in $\mathbb{B}[\![\gamma,\delta]\!]$. Graphically in in $\mathbb{Z}^2$, it corresponds to the set of all points belonging to the south-east cone.

Example 2 (Maximal representative) *Let $s = \gamma^1\delta^3 \oplus \gamma^2\delta^2 \oplus \gamma^2\delta^4 \oplus \gamma^4\delta^3 \oplus \gamma^4\delta^6 \oplus \gamma^5\delta^8 \oplus \gamma^6\delta^7 \oplus \gamma^7\delta^8 \oplus \gamma^8\delta^9$ be a series in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$.*
It can be check that it is equal to the following one :

$$\gamma^1\delta^3 \oplus \gamma^2\delta^4 \oplus \gamma^4\delta^6 \oplus \gamma^5\delta^8 \oplus \gamma^8\delta^9$$

$i.e.$ these both series are in the same equivalence class. The maximal representative of s is given by

$$(\gamma^*(\delta^{-1})^*)(\gamma^1\delta^3 \oplus \gamma^2\delta^4 \oplus \gamma^4\delta^6 \oplus \gamma^5\delta^8 \oplus \gamma^8\delta^9).$$

it is given in Figure 2.3.

Minimal Representative As seen previously all element in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ admits a maximal representative in $\mathbb{B}[\![\gamma,\delta]\!]$. Dually a minimal representative can be associated to each element. Especially for polynomial a minimal representative can be obtained by considering only the monomials corresponding to the vertices of the union of cones.

Example 3 (Minimal representative) *Let $s = \gamma^1\delta^4 \oplus \gamma^2\delta^2 \oplus \gamma^5\delta^6 \oplus \gamma^6\delta^3$ a polynomial of $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, the element $\gamma^*(\delta^{-1})^*(\gamma^1\delta^4 \oplus \gamma^5\delta^6)$ is the maximal representative (graphically it corresponds to the*

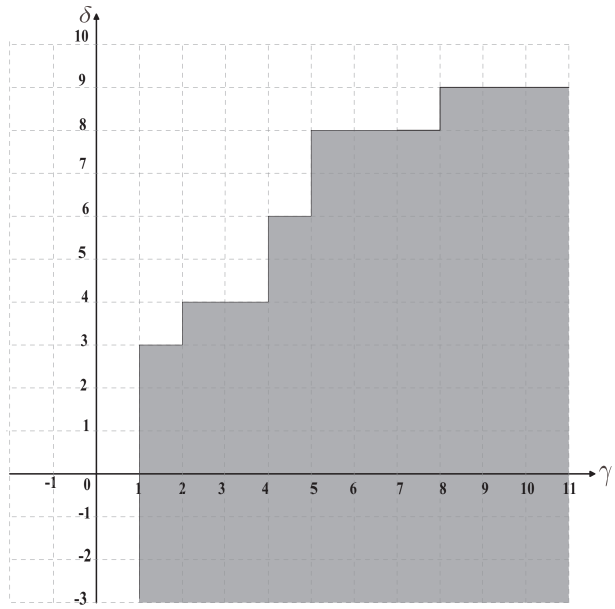

FIGURE 2.3 – Maximal representative of series $s = \gamma^1\delta^3 \oplus \gamma^2\delta^2 \oplus \gamma^2\delta^4 \oplus \gamma^4\delta^3 \oplus \gamma^4\delta^6 \oplus \gamma^5\delta^8 \oplus \gamma^6\delta^7 \oplus \gamma^7\delta^8 \oplus \gamma^8\delta^9$.

union of the two cones having the following vertices $(1,\ 4)$ *and* $(5,\ 6)$*. On the other hand* $(\gamma^1\delta^4 \oplus \gamma^5\delta^6)$ *is the minimal representative (only the two vertices are considered).*
The minimal representative of s *is given in Figure 2.4.*

2.3 Monomials in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$

As said previously a monomial $\gamma^n\delta^t$ represents the south-east cone with vertex (n,t). (see Figure 2.2)

Remark 1 *The following notation will be used for the bottom and the top element of* $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ *:* $\varepsilon = \gamma^{+\infty}\delta^{-\infty}$ *et* $\top = \gamma^{-\infty}\delta^{+\infty}$.

From previous definition the operations of addition, product, infimum, can be given in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$.

1. The sum of two monomials $\gamma^n\delta^t$ and $\gamma^{n'}\delta^{t'}$ corresponds to the union of the "south-east" cones having (n,t) and (n',t') as vertex. Hence, the sum of two monomials is a polynomial with two monomials, except if $n \leq n'$ and $t \geq t'$.
2. The product of two monomials $\gamma^n\delta^t$ and $\gamma^{n'}\delta^{t'}$ corresponds to the cone having $(n+n', t+t')$ as vertex.
3. l'inf of two monomials $\gamma^n\delta^t$ and $\gamma^{n'}\delta^{t'}$ is represented by the intersection of the "south-east" cone the vertices of which is $max(n,n')$ and $min(t,t')$.

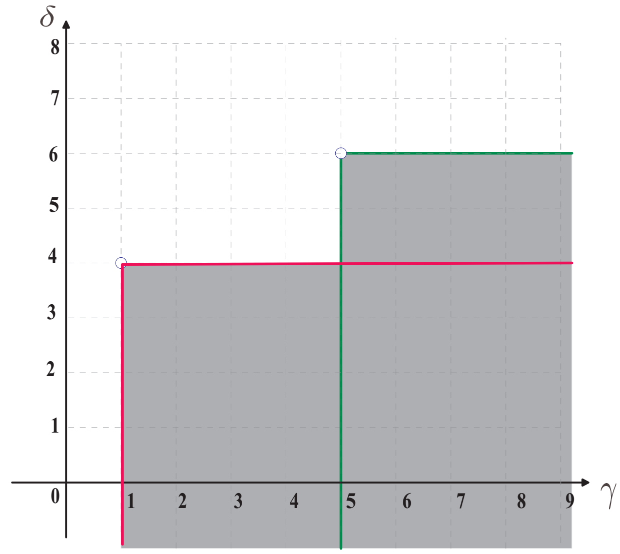

FIGURE 2.4 – The maximal representative (grey zone) and the minimal representative (vertices) $(\gamma^1\delta^4 \oplus \gamma^5\delta^6)$

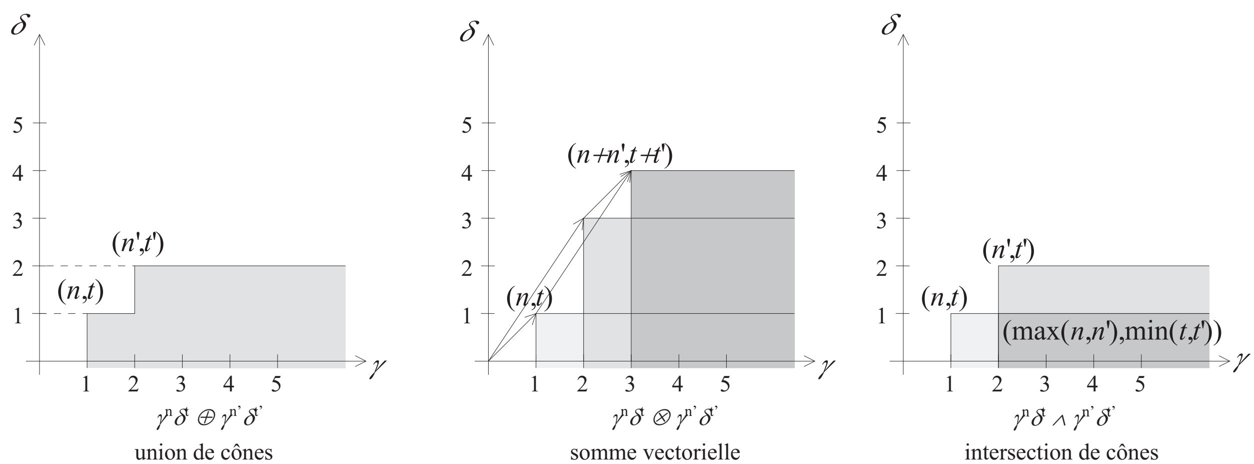

FIGURE 2.5 – Graphical representation of the monomials operation in $\mathcal{M}_{in}^{ax}[[\gamma,\delta]]$

By recalling that a semiring is a lattice with an order relation defined as follows :

$$a \oplus b = a \Leftrightarrow a \succeq b \Leftrightarrow a \wedge b = b,$$

the following rules are easy to establish for monomials in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$:

$$orderrelationmonomials \gamma^n\delta^t \preceq \gamma^{n'}\delta^{t'} \quad \Leftrightarrow n \geq n' \text{ and } t \leq t' \tag{2.2}$$

$$\begin{aligned}
\gamma^n\delta^t \oplus \gamma^{n'}\delta^{t'} &= \gamma^{min(n,n')}\delta^t && (2.3)\\
\gamma^n\delta^t \oplus \gamma^{n}\delta t' &= \gamma^{n}\delta^{max(t,t')} && (2.4)\\
\gamma^n\delta^t \wedge \gamma^{n'}\delta^{t'} &= \gamma^{max(n,n')}\delta^{min(t,t')} && (2.5)\\
\gamma^n\delta^t \otimes \gamma^{n'}\delta^{t'} &= \gamma^{n+n'}\delta^{t+t'}. && (2.6)
\end{aligned}$$

2.4 Operations over polynomials in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$

Definition 3 *A polynomial in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ is defined as the sum of m monomials.*

$$p = \bigoplus_{i=1}^{m} \gamma^{n_i}\delta^{t_i}$$

Polynomial can be given in a canonical form corresponding to its minimal representative.

$$p = \gamma^{n_1}\delta^{t_1} \oplus \gamma^{n_2}\delta^{t_2} \oplus ... \oplus \gamma^{n_m}\delta^{t_m}$$

with $n_1 < n_2 < ... < n_m$ and $t_1 < t_2 < ... < t_m$, $i.e.$ only non comparable monomials are in the polynomial.

2.4.1 Sum of two polynomials in canonical form

$$p \oplus p' = \bigoplus_{i=1}^{m} \gamma^{n_i}\delta^{t_i} \oplus \bigoplus_{j=1}^{m'} \gamma^{n'_j}\delta^{t'_j}$$

The polynomials are assumed to be in canonical form, $i.e$, the monomials are sorted in each polynomial. The result is given under the canonical form. It is obtained by merging the two polynomials according to the order relation **??**. The complexity is given by $\mathcal{O}(m+m')$.

2.4.2 Product of two polynomials in canonical form

$$p \otimes p' = \bigoplus_{i=1}^{m}\bigoplus_{j=1}^{m'} \gamma^{n_i+n'_j}\delta^{t_i+t'_j}$$

The polynomials are assumed to be in canonical form, $i.e$, the monomials are sorted in each polynomial. The result is given under the canonical form. This product needs (mm') products of monomials. Then, it is necessary to merge m polynomials composed with m' monomials (sum of m polynomials, each being composed with m' monomials), this can be done with the following complexity $\mathcal{O}(m'm/2\log(m))$. It could be judicious to permute p and p' before to proceed to this computation.

2.4.3 Operation $\wedge$ between two polynomials in canonical form

$$p \wedge p' = \bigoplus_{i=1}^{m}(\gamma^{n_i}\delta^{t_i} \wedge p') = \bigoplus_{i=1}^{m}\bigoplus_{j=1}^{m'} \gamma^{max(n_i,n'_j)}\delta^{min(t_i,t'_j)}$$

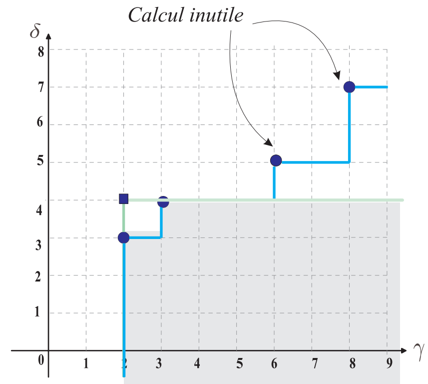

FIGURE 2.6 – Illustration of a practical trick to improve computation efficiency of the operation $\wedge$ between a monomial and a polynomial $\gamma^2\delta^4 \wedge (\gamma^2\delta^3 \oplus \gamma^3\delta^4 \oplus \gamma^6\delta^5 \oplus \gamma^8\delta^7) = \gamma^2\delta^3 \oplus \gamma^3\delta^3$

The polynomials are assumed to be in canonical form, $i.e$, the monomials are sorted in each polynomial. The result is given in canonical form. This operation needs (mm') operation $\wedge$ between monomials. Then, it is necessary to merge m polynomials composed with m' monomials (sum of m polynomials, each being composed with m' monomials), this can be done with the following complexity : $\mathcal{O}(m'm/2\log(m))$. It could be judicious to permute p and p' before to proceed to this computation.

Remark 2 *Generally, by computing $\gamma^{n_i}\delta^{t_i} \wedge p' = \bigoplus_{j=1}^{m'} \gamma^{max(n_i,n_{j'})}\delta^{min(t_i,t_{j'})}$ it is not necessary to compute until $j = m'$. Indeed if $min(t_i, t'_j) = t_i$ it is useless to continue the computations. This trick doesn't modify the complexity (the worst case has to be considered) but, practically, it increases greatly the algorithm efficiency. Figure 2.6 illustrates this trick.*

2.5 Periodical series in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$

Definition 4 (Periodical series) *A periodical series in $\mathcal{M}_{in}^{ax}[[\gamma,\delta]]$ can be written in the following form*

$$s = p \oplus qr^*$$

where p and q are polynomials and r is a monomial. In the next the following notations are considered :

$$p = \bigoplus_{i=1}^{m} \gamma^{n_i}\delta^{t_i}, q = \bigoplus_{j=1}^{l} \gamma^{N_j}\delta^{T_j} \quad \text{and} \quad r = \gamma^{\nu}\delta^{\tau}.$$

In this work r is assumed to be causal, that is $\nu \geq 0$ and $\tau \geq 0$, or $r = \varepsilon$.

Definition 5 (Asymptotic slope) *The asymptotic slope of series s is defined as*

$$\sigma_\infty(s) = \nu/\tau.$$

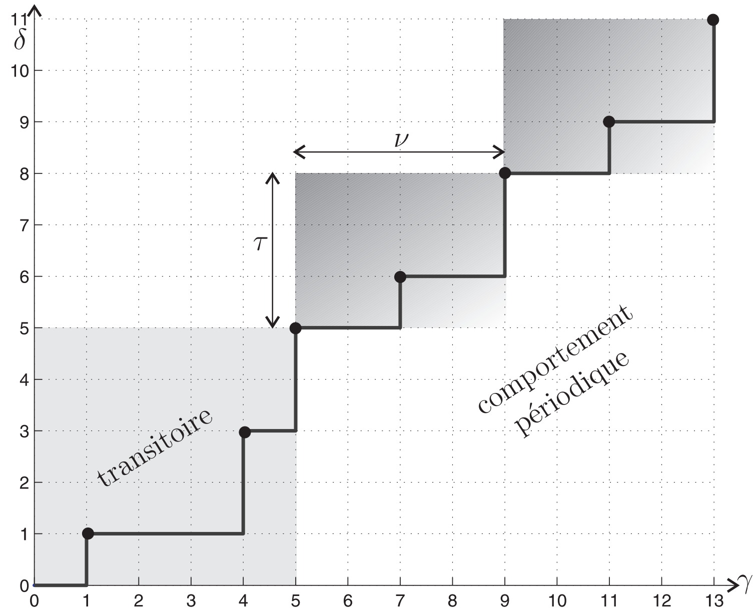

FIGURE 2.7 – Graphical representation of series $s = p \oplus qr^* = e \oplus \gamma\delta \oplus \gamma^4\delta^3 \oplus (\gamma^5\delta^5 \oplus \gamma^7\delta^6)(\gamma^4\delta^3)^*$.

Definition 6 (Proper Representation) *A series s is under a proper form if*

$$(n_m, t_m) < (N_1, T_1) \quad \textit{and} \quad (N_l, T_l) - (N_1, T_1) < (\nu, \tau).$$

Definition 7 *A proper form $s = p \oplus qr^*$ is said simpler than an other proper form $s = p' \oplus q'r'^*$ if*

$$(n_m, t_m) \leq (n'_m, t'_m) \;\; \textit{and} \;\; (\nu, \tau) \leq (\nu', \tau').$$

Theorem 2 *A periodical series s admits a simpler representation. This simpler representation is the canonical form of the series s.*

2.5.1 Sum of periodic series

$$s \oplus s' = (p \oplus qr^*) \oplus (p' \oplus q'r'^*)$$

The algorithms used to develop the operations over periodic series are mainly based on the handling of simple elements which are specific series with the following form :

$$s = \gamma^n\delta^t(\gamma^\nu\delta^\tau)^*.$$

Lemma 1 (Domination) *This lemma is rewritten from the Lemma 4.1.4 in [Gaubert, 1992].*
Let $s = \gamma^n\delta^t(\gamma^\nu\delta^\tau)^*$ and $s' = \gamma^{n'}\delta^{t'}(\gamma^{\nu'}\delta^{\tau'})^*$ be two simple elements, the asymptotic slopes of which are assumed not equal, i.e., $\nu/\tau \neq \nu'/\tau'$. If $\sigma_\infty(s) = \nu/\tau < \sigma_\infty(s') = \nu'/\tau'$ then it exists an integer $K \in \mathbb{N}$ such that

$$\gamma^{n'}\delta^{t'}\gamma^{K\nu'}\delta^{K\tau'}(\gamma^{\nu'}\delta^{\tau'})^* \preceq \gamma^n\delta^t(\gamma^\nu\delta^\tau)^*. \tag{2.7}$$

Proof : *We can write*

$$\gamma^n\delta^t(\gamma^\nu\delta^\tau)^* = \bigoplus_{i\geq 0} \gamma^{n+i\nu}\delta^{t+i\tau}$$

and

$$\gamma^{n'}\delta^{t'}\gamma^{K\nu'}\delta^{K\tau'}(\gamma^{\nu'}\delta^{\tau'})^* = \bigoplus_{j\geq K}\gamma^{n'+j\nu'}\delta^{t'+j\tau'}.$$

Hence, it exists a positive integer K such that inequality (2.7) holds if and only if

$$x\in\mathbb{N}, \forall x\geq K, \exists y\in\mathbb{N} \text{ such that } \begin{cases} n'+x\nu'\geq n+y\nu \\ t'+x\tau'\leq t+y\tau \end{cases}$$

or in other way

$$\forall x\geq K, \exists y\in\mathbb{N} \text{ such that } \frac{n'+x\nu'-n}{\nu}\geq y\geq\frac{t'+x\tau'-t}{\tau}. \tag{2.8}$$

Such integer $y\in\mathbb{Z}$ exists if

$$\left(\frac{n'+x\nu'-n}{\nu}\right)-\left(\frac{t'+x\tau'-t}{\tau}\right)\geq 1$$

which holds for a sufficiently large x, for example,

$$x\geq\left\lceil\frac{\nu(t'-t)+\tau(n-n')+\nu\tau}{\tau\nu'-\nu\tau'}\right\rceil$$

où $\lceil a\rceil\in\mathbb{Z}$ represents the smallest integer greater than $a\in\mathbb{Q}$. Furthermore, y must be positive which is ensured, according to (2.8) if

$$n'+x\nu'\geq n,$$

then in particular if

$$x\geq\left\lceil\frac{n-n'}{\nu'}\right\rceil.$$

To conclude, since K must be positive, it is sufficient to consider

$$K=\max\left(\left\lceil\frac{\nu(t'-t)+\tau(n-n')+\nu\tau}{\tau\nu'-\nu\tau'}\right\rceil,\left\lceil\frac{n-n'}{\nu'}\right\rceil,0\right) \tag{2.9}$$

to ensure that the domination given in Lemma 1 holds. □

This Lemma is illustrated in Figure 2.8. The computaiton of this integer K is then worth of interest since it represents from when series s is definitively above series s'. In Figure 2.8, the smallest K respecting Lemma 1 is $K=3$.

Remark 3 *It must be noticed that K given by expression (2.9)is not necessarily the smallest positive integer achieving the domination condition given in equation (2.7).*

Theorem 3 *The sum of two simple elements s and s' is a periodical series with an asymptotic slope such that*

$$\sigma_\infty(s\oplus s')=\min(\sigma_\infty(s),\sigma_\infty(s')).$$

Proof : Two cases have to be considered.

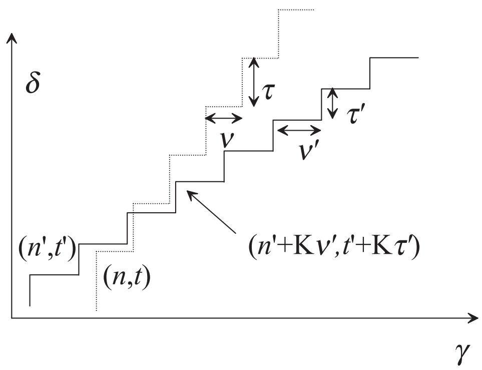

FIGURE 2.8 – Asymptotical domination of simple series with different slope

- **if** $\sigma_\infty(s) = \nu/\tau < \sigma_\infty(s') = \nu'/\tau'$. The sum of two simple elements can then be written as

$$\begin{aligned} s \oplus s' &= \gamma^n\delta^t(\gamma^\nu\delta^\tau)^* \oplus \gamma^{n'}\delta^{t'}(\gamma^{\nu'}\delta^{\tau'})^* \\ &= \gamma^n\delta^t(\gamma^\nu\delta^\tau)^* \oplus \Big[\gamma^{n'}\delta^{t'} \oplus \gamma^{n'+\nu'}\delta^{t'+\tau'} \oplus \cdots \\ &\quad \cdots \oplus \gamma^{n'+(K-1)\nu'}\delta^{t'+(K-1)\tau'}\Big] \oplus \gamma^{n'}\delta^{t'}\gamma^{K\nu'}\delta^{K\tau'}(\gamma^{\nu'}\delta^{\tau'})^* \end{aligned}$$

where K is given by equation (2.9). According to Lemma 1, the last term of the right hand expression is dominated by the other element, hence it can be removed. The sum of the simple elements can the be written

$$s \oplus s' = \bigoplus_{j=0}^{K-1} \gamma^{n'+j\nu'}\delta^{t'+j\tau'} \oplus \gamma^n\delta^t(\gamma^\nu\delta^\tau)^* = p'' \oplus q''r''^*$$

which is a series with an asymptotic slope $\sigma_\infty(s \oplus s') = \min(\sigma_\infty(s), \sigma_\infty(s'))$. The complexity is linked to the developing of the transient part, and is linear in K.

- **if** $\sigma_\infty(s) = \nu/\tau = \sigma_\infty(s') = \nu'/\tau'$. then $\exists k, k'$ such that

$$lcm(\nu,\nu') = k\nu = k'\nu' = \nu'' \text{ and } lcm(\tau,\tau') = k\tau = k'\tau' = \tau''.$$

The term r^* (resp.r'^*) can be written $r^* = \left(e \oplus r \oplus \ldots \oplus r^{k-1}\right)(r^k)^*$ (resp.$r'^* = \left(e \oplus r' \oplus \ldots \oplus r'^{k-1}\right)(r'^{k'})^*$) $i.e.$

$$\begin{aligned} r^* &= (\gamma^\nu\delta^\tau)^* = \left(e \oplus \gamma^\nu\delta^\tau \oplus \cdots \oplus \gamma^{(k-1)\nu}\delta^{(k-1)\tau}\right)(\gamma^{\nu''}\delta^{\tau''})^* \\ (r')^* &= (\gamma^{\nu'}\delta^{\tau'})^* = \left(e \oplus \gamma^{\nu'}\delta^{\tau'} \oplus \cdots \oplus \gamma^{(k'-1)\nu'}\delta^{(k'-1)\tau'}\right)(\gamma^{\nu''}\delta^{\tau''})^*. \end{aligned}$$

The sum of two simple elements with the same asymptotic slope can the be written

$$\begin{aligned} s \oplus s' &= \gamma^n\delta^t(\gamma^\nu\delta^\tau)^* \oplus \gamma^{n'}\delta^{t'}(\gamma^{\nu'}\delta^{\tau'})^* \\ &= \Big[\gamma^n\delta^t(e \oplus \gamma^\nu\delta^\tau \oplus \cdots \oplus \gamma^{(k-1)\nu}\delta^{(k-1)\tau}) \\ &\quad \oplus \gamma^{n'}\delta^{t'}(e \oplus \gamma^{\nu'}\delta^{\tau'} \oplus \cdots \oplus \gamma^{(k'-1)\nu'}\delta^{(k'-1)\tau'})\Big](\gamma^{\nu''}\delta^{\tau''})^* = q''r''^* \end{aligned}$$

which is a periodical series with the asymptotic slope $\sigma_\infty(s \oplus s') = \min(\sigma_\infty(s), \sigma_\infty(s')) = \frac{\nu}{\tau} = \frac{\nu'}{\tau'}$. The complexity is based on the sum of two polynomials with k and k' monomials then it is $(\mathcal{O}(k + k'))$. It can also expressed according to ν and ν' by considering a upper approximation of the lcm, which leas to $k = \nu'$ and $k' = \nu$.

□

Remark 4 *The results obtained in the previous proofs are under the form of periodical series but the resulting series are not necessarily under their canonical form. It is necessary to apply the algorithm leading to the canonical form in the end. The same remark could be done for all the algorithms introduced in the next.*

Theorem 4 *The sum of two periodical series s and s' is a periodical series with an asymptotic slope*

$$\sigma_\infty(s \oplus s') = \min(\sigma_\infty(s), \sigma_\infty(s')).$$

Proof :

• **if** $\sigma_\infty(s) = \sigma_\infty(s')$ *then by considering*

$$\nu'' = ppcm(\nu, \nu'),\ \tau'' = ppcm(\tau, \tau'),\ \ k = ppcm(\nu, \nu')/\nu,\ \textit{and}\ k' = ppcm(\nu, \nu')/\nu'$$

the sum can be written

$$\begin{aligned} s \oplus s' &= p \oplus qr^* \oplus p' \oplus q'r'^* \\ &= [p \oplus p'] \oplus \left[q(e \oplus \ldots \oplus r^{(k-1)}) \oplus q'(e \oplus \ldots \oplus r'^{(k'-1)}) \right] (\gamma^{\nu''}\delta^{\tau''})^* \\ &= p'' \oplus q''(r'')^* \end{aligned}$$

which is a periodical series with an asymptotic slope $\sigma_\infty(s \oplus s') = \min(\sigma_\infty(s), \sigma_\infty(s')) = \frac{\nu''}{\tau''} = \frac{\nu}{\tau} = \frac{\nu'}{\tau'}$. The complexity is based on the sum and the product of polynomials. Let us note that $q(e \oplus \ldots \oplus r^{(k-1)})$ is compute with a complexity $\mathcal{O}(kl/2\log(l))$, where $k = ppcm(\nu, \nu')/\nu$ which can be approximated by ν'. In the same way $q'(e \oplus \ldots \oplus r'^{(k'-1)})$ is compute with a complexity $\mathcal{O}(k'l'/2log(l'))$, where $k' = ppcm(\nu, \nu')/\nu'$ which can be approximated by ν.

• **if** $\sigma_\infty(s) < \sigma_\infty(s')$ *and if s and s' are under their canonical form, it is possible to approximate qr^* by a smaller simple element mr^* and to approximate $q'r'^*$ by a greater simple element $m'r'^*$. Indeed the following inequality holds*

$$qr^* = \left(\bigoplus_{j=1}^{l} \gamma^{N_j}\delta^{T_j} \right) (\gamma^\nu\delta^\tau)^* \succeq \gamma^{N_1}\delta^{T_1}(\gamma^\nu\delta^\tau)^*$$

and also

$$q'r'^* = \left(\bigoplus_{j=1}^{l'} \gamma^{N'_j}\delta^{T'_j} \right) (\gamma^{\nu'}\delta^{\tau'})^* \preceq \gamma^{N'_1}\delta^{T'_1+\tau'}(\gamma^{\nu'}\delta^{\tau'})^*$$

(Figure 2.9 gives a graphical interpretation of these relations).
Lemma 1showed that it exists an integer K such that

$$\gamma^{N'_1}\delta^{T'_1+\tau'}\gamma^{K\nu'}\delta^{K\tau'}(\gamma^{\nu'}\delta^{\tau'})^* \preceq \gamma^{N_1}\delta^{T_1}(\gamma^\nu\delta^\tau)^*,$$

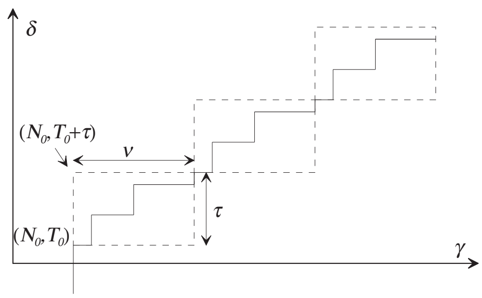

FIGURE 2.9 – Upper and lower approximation of qr^* by two simple elements : $\gamma^{N_1}\delta^{T_1}(\gamma^\nu\delta^\tau)^* \preceq qr^* \preceq \gamma^{N_1}\delta^{T_1+\tau}(\gamma^\nu\delta^\tau)^*$

then by considering isotony of the product law the following relations hold

$$\gamma^{K\nu'}\delta^{K\tau'}q'(\gamma^{\nu'}\delta^{\tau'})^* \preceq \gamma^{N_1'}\delta^{T_1'+\tau'}\gamma^{K\nu'}\delta^{K\tau'}(\gamma^{\nu'}\delta^{\tau'})^* \preceq \gamma^{N_1}\delta^{T_1}(\gamma^\nu\delta^\tau)^* \preceq q(\gamma^\nu\delta^\tau)^*.$$

The sum of two periodical series with different asymptotic slope can be written as

$$\begin{aligned} s \oplus s' &= \left[p \oplus p' \oplus q'\left[\bigoplus_{i=0}^{K-1}\gamma^{i\nu'}\delta^{i\tau'}\right]\right] \oplus qr^* \\ &= p'' \oplus q''r''^* \end{aligned}$$

where

$$K = \max\left(\left\lceil\frac{\nu(T_1'+\tau'-T_1)+\tau(N_1-N_1')+\nu\tau}{\tau\nu'-\nu\tau'}\right\rceil, \left\lceil\frac{N_1-N_1'}{\nu'}\right\rceil, 0\right).$$

The computation is then based on sums and products of polynomials. Let us note that the product $q'\left[\bigoplus_{i=0}^{K-1}\gamma^{i\nu'}\delta^{i\tau'}\right]$ *can be compute with the following complexity* $\mathcal{O}(Kl'\log(l'))$. □

2.6 Product of two periodical series

Theorem 5 *Let* $s = \gamma^n\delta^t(\gamma^\nu\delta^\tau)^*$ *and* $s = \gamma^{n'}\delta^{t'}(\gamma^{\nu'}\delta^{\tau'})^*$ *be two simple elements. The product* $s \otimes s'$ *is a periodical series with an asymptotic slope*

$$\sigma_\infty(s \otimes s') = \min(\sigma_\infty(s), \sigma_\infty(s')).$$

Proof :

• **if** $\sigma_\infty(s) < \sigma_\infty(s')$ then according to Lemma 1, it exists an integer K such that

$$r'^K r'^* \preceq r^*. \tag{2.10}$$

Then,by considerint the isotony of the product law, $r'^K r'^* r^* \preceq r^* r^* = r^*$. By recalling that $r^* r'^* = r^*(e \oplus r' \oplus \cdots \oplus r'^{(K-1)} \oplus r'^K r'^*)$ which, according to equtation (2.10), can be simplified as

$$r^* r'^* = r^*(e \oplus r' \oplus \cdots \oplus r'^{(K-1)}).$$

The product of two simle elements with different asymptotic slope can then be written easily

$$\gamma^n\delta^t(\gamma^\nu\delta^\tau)^* \otimes \gamma^{n'}\delta^{t'}(\gamma^{\nu'}\delta^{\tau'})^* = \gamma^{n+n'}\delta^{t+t'}\left(e \oplus \gamma^{\nu'}\delta^{\tau'} \oplus \cdots \oplus \gamma^{(K-1)\nu'}\delta^{(K-1)\tau'}\right)(\gamma^\nu\delta^\tau)^*$$

with

$$K = \max\left(\left\lceil \frac{\nu\tau}{\tau\nu' - \nu\tau'} \right\rceil, 0\right).$$

The complexity of polynomial expansion is linear ($\mathcal{O}(K)$).

- **If** $\sigma_\infty(s) = \sigma_\infty(s')$**.** By considering $\nu'' = ppcm(\nu, \nu') = k\nu = k'\nu'$ and $\tau'' = ppcm(\tau, \tau') = k\tau = k'\tau'$, the following equalities hold

$$\begin{aligned} r^* &= [e \oplus \cdots \oplus r^{(k-1)}](r'')^* \\ r'^* &= [e \oplus \cdots \oplus r'^{(k'-1)}](r'')^* \end{aligned}$$

The product of two simple elements with the same slope can be written

$$\begin{aligned} s \otimes s' &= \gamma^{n+n'}\delta^{t+t'}[e \oplus \cdots \oplus r^{(k-1)}] \otimes [e \oplus \cdots \oplus r'^{(k'-1)}](r'')^* \\ &= q''r''^*. \end{aligned}$$

The algorithm complexity is then based on the product of two polynomials, $i.e.$ $\mathcal{O}(k'k\log(k))$, with $k = ppcm(\nu, \nu')/\nu$ and $k' = ppcm(\nu, \nu')/\nu'$.

□

Theorem 6 *The product of two periodical series is a periodical series with the asymptotic slope*

$$\sigma_\infty(s \otimes s') = \min(\sigma_\infty(s), \sigma_\infty(s')).$$

Proof : The product of two periodical series can be written

$$\begin{aligned} s \otimes s' &= [p \oplus qr^*] \otimes [p' \oplus q'r'^*] \\ &= pp' \oplus pq'r'^* \oplus p'qr^* \oplus qq'r^*r'^*. \end{aligned}$$

The computation of this product is based on the product and the sum of polynomials. The last term ($r^*r'^*$) is the product of simple elements. □

2.6.1 Star of a polynomial

$$p = \bigoplus_{i=1}^{m} \gamma^{n_i}\delta^{t_i}$$

$$s = p^* = (\bigoplus_{i=1}^{m} \gamma^{n_i}\delta^{t_i})^*$$

Theorem 7 *The star of a polynomial p is a periodical series s with an asymtotic slope*

$$\sigma_\infty(s) = \min(n_i/t_i) \text{ with } i \in [1, m]$$

Proof : Sum of monomials is commutative, hence

$$s = (\bigoplus_{i=1}^{m} \gamma^{n_i}\delta^{t_i})^* = \bigotimes_{i=1}^{m} (\gamma^{n_i}\delta^{t_i})^*$$

The computation of the star of a polynomial with m monomials can then be computed as the product of m simple elements. The asymptotic slope is then given by $\sigma_\infty(s) = \min(n_i/t_i)$ with $i \in [1, m]$. The computation complexity can then be bounded by considering the one of the previous theorem. □

2.6.2 Star of periodical series

Let s be a periodical series :

$$s = p \oplus qr^*$$

s^* is given by :

$$s^* = (p \oplus qr^*)^* = p^*(e \oplus q(q \oplus r)^*).$$

Hence, this calculation can be decomposed as the computation of star of the polynomials (p^* and $(q \oplus r)^*$), it is then sufficient to sum the two resulting series.

2.7 Software toolbox MinMaxgd

MinMaxgd is as set of C++ classes, they make it possible to handle periodical series in dioid $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$. Below is given the howto of this tools.

2.7.1 OPerations with monomials

The elemntary object is a monomial, they are represented by a class called gd. We recall that the sum of monomial is not a monomial. Nevertheless the following internal operator are defined.

gd r(2,3) ;	declaration and initialization of a monomial $r = \gamma^2\delta^3$
gd r ; r.init(2,3)	declaration of a monomial, default value ε method affecting $r = \gamma^2\delta^3$
r3= inf(r1,r2)	$\wedge$ of two monomials, $r3 = r1 \wedge r2$
r3= otimes(r1,r2) ;	product of two monomials $r3 = \otimes(r1, r2)$
r3=frac(r1,r2) ;	Residuation of two monomials $r3 = r2 \backslash\!\!\circ\, r1 = r1 \not{\circ}\, r2$
==,! =,<=,>=	Comparing expression, it return 0 if the condition is false , 1 either.

The neutral element e is a monomial coded with the value $e(0,0)$, the absorbing element ε is coded as a monomial defined as $\varepsilon(2147483647, -2147483648)$.

Below a small example illustrates the use of these methods.

```
void main(void) {
    gd a(2,3);
    gd b,res_otimes,res_inf,res_frac;

    b.init(3,6);

    res_otimes = otimes(a,b);

    res_inf = inf(a,b);

    res_frac = frac(a,b);
    cout<<"res_frac is equal to"<<res_frac<<endl; // the monomial is printed
}
```

In this example $a = \gamma^2\delta^3$ and $b = \gamma^3\delta^6$.

the results are

- res_otimes = $a \otimes b = \gamma^5\delta^9$
- res_inf = $a \wedge b = \gamma^3\delta^3$
- res_frac = $a \phi b = b \diamond a = \gamma^{-1}\delta^{-3}$

2.7.2 Operations with polynomials

A polynomial is defined in class called $poly$. It is composed of private member : an array of monomials, n defining the number of monomials in the array, $simple$ an integer equal to 1 if the polynomial is under its proper form, and $nblock$ which corresponds to the memory size of the polynomial, a block is composed of 64 monomials.

Below the different method available are described.

poly p ;	declaration of a polynomial, it is initiated with monomial ε, $i.e.$, $n = 1$, $nblock = 1$, $simple = 1$
p.init(2,3)(4,5) ;	initialization of a polynomial, it is automatically put in proper form $p = \gamma^2\delta^3 \oplus \gamma^4\delta^5$
poly oplus(poly,poly) ;	sum of two polynomials, the result is given in proper form
poly oplus(poly,gd) ;	sum of a polynomial with a monomial
poly oplus(gd,poly) ;	sum of a monomial with a polynomial
poly otimes(poly,poly ;)	product of two polynomials
poly otimes(poly,gd) ;	product of a polynomial with a monomial
poly otimes(gd,poly) ;	product of a monomial with a polynomial
poly frac(poly,poly)	residuationof two polynomials
poly frac(poly,gd)	residuation of a polynomial with a monomial
poly frac(gd,poly) ;	residuation of a monomial with a polynomial
poly inf(poly,poly) ;	inf of two polynomials
poly inf(poly,gd) ;	inf of a polynomial with a monomial
poly inf(gd,poly) ;	inf of a monomail with a polynomial
poly prcaus(poly) ;	projection of a polynomial in the causal set

Some other initialization exist, the developer can check in the header files associated to this class. Below an example to illustrate the use of the class $poly$.

```
void main(void) {
    poly p1,p2;
    poly res_oplus,res_frac,res_inf,res_otimes;

    p1.init(1,1)(2,3)(4,5);
    p2.init(1,3)(3,3)(8,4);

    res_otimes = otimes(p1,p2);
    res_frac = frac(p1,p2);
    res_inf = inf(p1,p2);
    res_oplus = oplus(p1,p2);
    cout<<"res_oplus="<<res_oplus<<endl; //result is printed
}
```

In this example $p1 = \gamma^1\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5$ and $p2 = \gamma^1\delta^3 \oplus \gamma^3\delta^3 \oplus \gamma^8\delta^4$

the following results is obtained

- res_otimes $= p1 \otimes p2 = \gamma^2\delta^4 \oplus \gamma^3\delta^6 \oplus \gamma^5\delta^8 \oplus \gamma^{12}\delta^9$
- res_frac $= p2 \circ\!\!\!\!\setminus p1 = \gamma^0\delta^{-2} \oplus \gamma^1\delta^0 \oplus \gamma^3\delta^1$
- res_inf $= p1 \wedge p2 = \gamma^1\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^8\delta^4$
- res_oplus $= p1 \oplus p2 = \gamma^1\delta^3 \oplus \gamma^4\delta^5$

2.7.3 OPeration between periodical series

The periodical series (see definition 4) is defined in the class $serie$ The member are two polynomials, p (the polynomial depicting the transient part of the series) and q (the polynomial depcting the pattern repeated periodically), the slope si represented by a monomial r.

serie s ;	declaration of a series s, it is equal to ε
s.init(p,q,r) ;	initialization of a series s thanks to polynomials p,q, and a monomial r
serie oplus(serie,serie) ;	sum of two series
serie oplus(serie,poly) ;	sum of series with a polynomial
serie oplus(poly,serie) ;	sum of a polynomial with a series
serie oplus(serie,gd) ;	sum of a series with a monomial
serie oplus(gd,serie) ;	sum of a monomialwith a series
serie otimes(serie,serie ;	prouct of two series
serie otimes(serie,poly) ;	product of a series with a polynomial
serie otimes(poly,serie) ;	product of a polynomial with a series
serie otimes(serie,gd) ;	product of a series with a monomial
serie otimes(gd,serie) ;	product of a monomial with a series
serie frac(serie,serie) ;	residutation of two series
serie frac(serie,poly) ;	residuation of a series with a polynomial
serie frac(poly,serie) ;	residuation of a polynomial with a series
serie frac(serie,gd) ;	residuation of a series with a monomial
serie frac(gd,serie) ;	residuation of a monomialwith a series
serie inf(serie,serie),	inf of two series
serie inf(serie,poly) ;	inf of a series with a polynomial
serie int(poly,serie) ;	inf of a polynomial with a series
serie int(serie,gd ;)	inf of a series with a monomial
serie int(gd,serie) ;	inf of a monomial with a series
serie star(serie) ;	star of a series
serie star(poly) ;	star of a polynomial
serie star(gd) ;	star of a monomial
serie prcaus(serie) ;	projection in causl set of a series
void canon() ;	put the series in its canonical form

Illustration for operations between series operations

```
void main(void) {
```

```
    serie s1,s2,s_otimes,s_frac,s_oplus,s_inf,s_star;
    poly p1,q1,p2,q2;
    gd r1,r2;

    p1.init(1,1)(2,3)(4,5);
    q1.init(10,11)(12,15);
    r1.init(2,3);

    p2.init(1,3)(3,3)(8,4);
    q2.init(10,5)(12,7)(13,9);
    r2.init(4,4);

    s1.init(p1,q1,r1);
    s2.init(p2,q2,r2);

    s_otimes = s_otimes(s1,s2);
    s_frac = frac(s1,s2);
    s_oplus = oplus(s1,s2);
    s_inf = inf(s1,s2);
    s_star = star(s1);
    cout<<"s_star ="<<s_star<<endl;
}
```

In this example $s1 = \gamma^1\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5 \oplus (\gamma^{10}\delta^{11} \oplus \gamma^{12}\delta^{15})(\gamma^2\delta^3)^*$ and $s2 = \gamma^1\delta^3 \oplus \gamma^3\delta^3 \oplus \gamma^8\delta^4 \oplus (\gamma^{10}\delta^5 \oplus \gamma^{12}\delta^7 \oplus \gamma^{13}\delta^9)(\gamma^4\delta^4)^*$

The following results are obtained

s_otimes = $s1 \otimes s2 = \gamma^2\delta^4 \oplus \gamma^3\delta^6 \oplus \gamma^5\delta^8 \oplus \gamma^{11}\delta^{14} \oplus (\gamma^{13}\delta^{18})(\gamma^2\delta^3)^*$
s_frac = $s1 \backslash\!\!\!\circ\, s2 = \gamma^0\delta^{-2} \oplus \gamma^1\delta^0 \oplus \gamma^3\delta^2 \oplus \gamma^9\delta^8 \oplus (\gamma^{11}\delta^{12})(\gamma^2\delta^3)^*$
s_oplus = $s1 \oplus s2 = \gamma^1\delta^3 \oplus \gamma^4\delta^5 \oplus \gamma^{10}\delta^{11} \oplus (\gamma^{12}\delta^{15})(\gamma^2\delta^3)^*$
s_inf = $s1 \wedge s2 = \gamma^1\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^8\delta^4 \oplus \gamma^{10}\delta^5 \oplus (\gamma^{12}\delta^7 \oplus \gamma^{13}\delta^9)(\gamma^4\delta^4)^*$
s_star = $s1^* = (\gamma^0\delta^0 \oplus \gamma^1\delta^1)(\gamma^2\delta^3)^*$

2.7.4 Operations between matrices of periodical series

The last class proposed in this software tools is the class $matrix$ which make it possible to handle matrices of periodic series.

smatrix sm(4,4) ;	declarationof a matrix of size (4×4), each entry is initalized with ε
smatrix oplus(smatrix,smatrix) ;	sum of two matrices
smatrix otimes(smatrix,smatrix)	product of two matrices
smatrix lfrac(smatrix,smatrix) ;	left residaution of two matrices
smatrix rfrac(smatrix,smatrix) ;	right residuation of two matrices
smatrix inf(smatrix,smatrix) ;	inf of two matrices

Example illustrating the use of the Software toolbox

To illustrate the use of the class $matrix$ we propose a methodolgy to model a timed event graph. The one of Figure 2.10 is considered. Below the correposnding $C++$ file making the correposnding computation.

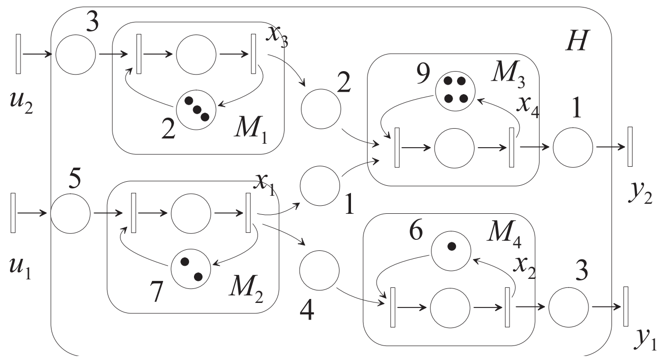

FIGURE 2.10 – A Timed event graph , with 2 inputs and two outputs .

```
void main(void) {
// Declaration of matrices
    gd m;
    smatrix A(4,4),B(4,2),C(2,4);

// INitialization of the entries
    A(0,0).init(epsilon,m.init(2,7),e);
    A(0,1).init(epsilon,epsilon,e);
    A(0,2).init(epsilon,epsilon,e);
    A(0,3).init(epsilon,epsilon,e);

    A(1,0).init(epsilon,m.init(0,4),e);
    A(1,1).init(epsilon,m.init(1,6),e);
    A(1,2).init(epsilon,epsilon,e);
    A(1,3).init(epsilon,epsilon,e);

    A(2,0).init(epsilon,epsilon,e);
    A(2,1).init(epsilon,epsilon,e);
    A(2,2).init(epsilon,m.init(3,2),e);
    A(2,3).init(epsilon,epsilon,e);

    A(3,0).init(epsilon,m.init(0,1),e);
    A(3,1).init(epsilon,epsilon,e);
    A(3,2).init(epsilon,m.init(0,2),e);
    A(3,3).init(epsilon,m.init(4,9),e);
```

The following matrix is obtained $A = \begin{pmatrix} \gamma^2\delta^7 & \varepsilon & \varepsilon & \varepsilon \\ \delta^4 & \gamma\delta^6 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \gamma^3\delta^2 & \varepsilon \\ \delta & \varepsilon & \delta^2 & \gamma^4\delta^9 \end{pmatrix}$

The same for matrix B

```
    B(0,0).init(epsilon,m.init(0,5),e);
    B(0,1).init(epsilon,epsilon,e);
```

```
B(1,0).init(epsilon,epsilon,e);
B(1,1).init(epsilon,epsilon,e);

B(2,0).init(epsilon,epsilon,e);
B(2,1).init(epsilon,m.init(0,3),e);

B(3,0).init(epsilon,epsilon,e);
B(3,1).init(epsilon,epsilon,e);
```

the following matrix is obtained $B = \begin{pmatrix} \delta^5 & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \delta^3 \\ \varepsilon & \varepsilon \end{pmatrix}$

the same for C

```
C(0,0).init(epsilon,epsilon,e);
C(0,1).init(epsilon,m.init(0,3),e);
C(0,2).init(epsilon,epsilon,e);
C(0,3).init(epsilon,epsilon,e);

C(1,0).init(epsilon,epsilon,e);
C(1,2).init(epsilon,epsilon,e);
C(1,0).init(epsilon,epsilon,e);
C(1,3).init(epsilon,m.init(0,1),e);
```

this yields the following matrix $C = \begin{pmatrix} \varepsilon & \delta^3 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \delta \end{pmatrix}$

The three matrices being initialized, it is possible to compute the transfer matrix of the timed event graph, it is obtained by computing $H = CA^*B$.

First the computation of A^* is done

```
smatrix H(4,4);

H=star(A);
```

the the computation of CA^*

```
H=otimes(C,H);
```

to conclude the transfer function is computed CA^*B

```
H=otimes(H,B);
cout<<"transfer function H "<<H<<endl;
```

The computation of H yiels $H = \begin{pmatrix} \delta^{12}(\gamma\delta^6)^* & \varepsilon \\ \delta^7(\gamma^2\delta^7)^* & (\delta^6 \oplus \gamma^3\delta^8)(\gamma^4\delta^9)^* \end{pmatrix}$

Classicaly, a feedback controller is searched, il allows to the system to behave as a reference model (see [Cottenceau et al., 1999, Lhommeau et al., 2004, Cottenceau et al., 2001] for the theoretical details).

Below the following reference model is considered $G_{ref} = (HF_0)^*H$ with $F_0 = \begin{pmatrix} \gamma^3 & \gamma^3 \\ \gamma^3 & \gamma^3 \end{pmatrix}$.

```
    smatrix F_0(2,2),G_ref(2,2);
    F_0(0,0).init(epsilon,m.init(3,0),e);
    F_0(0,1).init(epsilon,m.init(3,0),e);
    F_0(1,0).init(epsilon,m.init(3,0),e);
    F_0(1,1).init(epsilon,m.init(3,0),e);
/* modèle de référence */
    G_ref=otimes(H,F_0);
    G_ref=star(G_ref);
    G_ref=otimes(G_ref,H);
```

Hence it is possible to compute an optimal feedback controller. This computaion needs the operation of residuations ans of the projection in the causal set. This latest projection ensure the realization of the control law, formally the controller is given by : $F = Pr_+(H \backslash G_{ref} / H)$. In the program the optimal feedbakc is denoted F_opt.

First we compute $H \backslash G_{ref}$

```
    smatrix F_opt(2,2);

    F_opt=lfrac(G_ref,H);
```

then $H \backslash G_{ref} / H$

```
    F_opt=rfrac(F_opt,H);
```

The last step is the causal projection of F_opt in the set of causal matrices.

```
    F_opt=prcaus(F_opt);
}
```

Then the following feedback is obtained $F = \begin{pmatrix} \gamma^3(\gamma^1\delta^6)^* & \gamma^3 \oplus \gamma^4\delta^2 \oplus \gamma^5\delta^7 \oplus (\gamma^6\delta^{12})(\gamma^1\delta^6)^* \\ \gamma^3\delta(\gamma^1\delta^6)^* & \gamma^3\delta^1 \oplus \gamma^4\delta^3 \oplus \gamma^5\delta^8 \oplus (\gamma^6\delta^{13})(\gamma^1\delta^6)^* \end{pmatrix}$

A realization of this controller is given in dotted lined in Figure 2.11.

2.8 Conclusion

This document is not exhaustive. Some other examples can be found in the joint works with my colleague Ying Shang from Edwardsville [Hardouin et al., 2011, Shang et al., 2013] university. It was useful to solve disturbance decoupling problems for max-plus linear systems.

This library was also used to synthesis observer for (max,plus) linear systems (see `http://perso-laris.univ-angers.fr/~hardouin/Observer.html`, [Hardouin et al., 2010b, Hardouin et al., 2010a]).

More recently, it was useful in the PhD of Thomas Brunsch from the group of Jörg Raisch in TU Berlin. It was used to simulate real system from an industrial partner building High-Throughput screening systems (see the related papers [Brunsch et al., 2010, Brunsch et al., 2012] and the book chapters [Hardouin et al., 2013, Brunsch et al., 2013]).

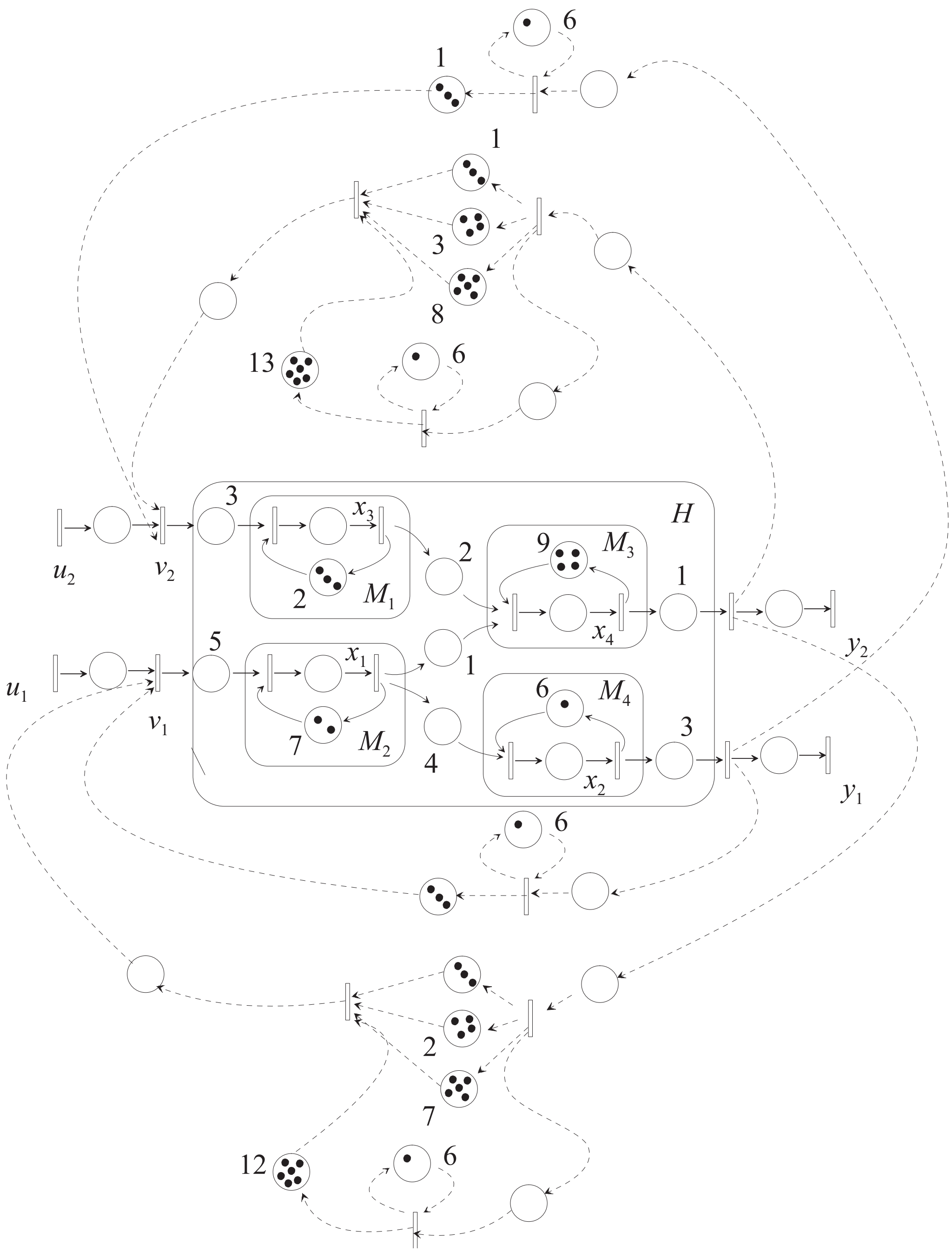

FIGURE 2.11 – Example of TEG with an output feedback.

Bibliographie

[Abeka, 2005] Abeka, A. (2005). *Utilisation de la transformé de Fenchel pour représenter la dynamique des systèmes (max,+) linéaires*. Dea automatique et informatique appliqué, LISA - Université d'Angers.

[Baccelli et al., 1992] Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J. (1992). *Synchronization and Linearity : An Algebra for Discrete Event Systems*. Wiley and Sons.

[Brunsch et al., 2010] Brunsch, T., Hardouin, L., and Raisch, J. (2010). Control of Cyclically Operated High-Throughput Screening Systems. In *10th International Workshop on Discrete Event Systems, 2010, Berlin, WODES'10.*, pages 177–183.

[Brunsch et al., 2012] Brunsch, T., Raisch, J., and Hardouin, L. (2012). Modeling and control of high-throughput screening systems. *Control Engineering Practice*, 20 :1 :14–23. doi :10.1016/j.conengprac.2010.12.006.

[Brunsch et al., 2013] Brunsch, T., Raisch, J., Hardouin, L., and Boutin, O. (2013). Discrete-event systems in a dioid framework : Modeling and analysis. In Seatzu, C., Silva, M., and van Schuppen, J. H., editors, *Control of Discrete-Event Systems*, volume 433 of *Lecture Notes in Control and Information Sciences*, pages 431–450. Springer London.

[Cohen, 1993] Cohen, G. (1993). Two-dimensional domain representation of timed event graphs. In *Summer School on Discrete Event Systems*. Spa, Belgium.

[Cohen et al., 1989] Cohen, G., Moller, P., Quadrat, J.-P., and Viot, M. (1989). Algebraic Tools for the Performance Evaluation of Discrete Event Systems. *IEEE Proceedings : Special issue on Discrete Event Systems*, 77(1) :39–58.

[Cottenceau, 1999] Cottenceau, B. (1999). *Contribution à la commande de systÃ¨mes à événements discrets : synthèse de correcteurs pour les graphes d'événements temporisés dans les dioïdes*. Thèse, LISA - Université d'Angers.

[Cottenceau et al., 1999] Cottenceau, B., Hardouin, L., Boimond, J.-L., and Ferrier, J.-L. (1999). Synthesis of greatest linear feedback for timed event graphs in dioid. *IEEE Transactions on Automatic Control*, 44(6) :1258–1262. Disponible à l'adresse `doi:10.1109/9.769386`.

[Cottenceau et al., 2001] Cottenceau, B., Hardouin, L., Boimond, J.-L., and Ferrier, J.-L. (2001). Model Reference Control for Timed Event Graphs in Dioids. *Automatica*, 37(9) :1451–1458. Disponible à l'adresse `doi:10.1016/S0005-1098(01)00073-5`.

[Gaubert, 1992] Gaubert, S. (1992). *Théorie des Systèmes Linéaires dans les Dioïdes*. Thèse, École des Mines de Paris.

[Gruet, 1995] Gruet, B. (1995). *Structure de commande en boucle fermée des systèmes Ã évènements discrets.* DEA, LISA - Université d'Angers - France.

[Hardouin et al., 2013] Hardouin, L., Boutin, O., Cottenceau, B., Brunsch, T., and Raisch, J. (2013). Discrete-event systems in a dioid framework : Control theory. In Seatzu, C., Silva, M., and van

Schuppen, J. H., editors, *Control of Discrete-Event Systems*, volume 433 of *Lecture Notes in Control and Information Sciences*, pages 451–469. Springer London.

[Hardouin et al., 2011] Hardouin, L., Lhommeau, M., and Shang, Y. (2011). Towards Geometric Control of Max-Plus Linear Systems with Applications to Manufacturing Systems. In *Proceedings of the 50þ IEEE Conference on Decision and Control and European Control Conference*, pages 171–176, Orlando, Floride, États Unis d'Amérique. Disponible à l'adresse istia.univ-angers.fr/~hardouin/YShangDDPCDC2011.pdf.

[Hardouin et al., 2010a] Hardouin, L., Maia, C. A., Cottenceau, B., and Lhommeau, M. (2010a). Max-plus Linear Observer : Application to manufacturing Systems. In *Proceedings of the 10þ International Workshop on Discrete Event Systems, WODES'10*, pages 171–176, Berlin. Disponible à l'adresse istia.univ-angers.fr/~hardouin/Observer.html.

[Hardouin et al., 2010b] Hardouin, L., Maia, C. A., Cottenceau, B., and Lhommeau, M. (2010b). Observer Design for (max,plus) Linear Systems. *IEEE Transactions on Automatic Control*, 55(2). doi:10.1109/TAC.2009.2037477.

[Lhommeau et al., 2004] Lhommeau, M., Hardouin, L., Cottenceau, B., and Jaulin, L. (2004). Interval Analysis and Dioid : Application to Robust Controller Design for Timed Event Graphs. *Automatica*, 40(11) :1923–1930. Disponible à l'adresse doi:10.1016/j.automatica.2004.05.013.

[Shang et al., 2013] Shang, Y., Hardouin, L., Lhommeau, M., and Maia, C. (2013). Open loop controllers to solve the disturbance decoupling problem for max-plus linear systems. In *European Control Conference, ECC 2013*, Zurich.