Chapitre VIII

MAX: un outil de calcul rationnel dans les dioïdes

Introduction

MAX se présente comme un ensemble de macros écrites en MAPLE, Version V, qui permettent de manipuler des séries rationnelles dans le dioïde $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]$. On a également inclus le traitement de l'algèbre $(\max,+)$. Notre propos n'était pas ici de faire un logiciel "définitif", mais de valider les algorithmes décrits plus haut. Voici la transcription LaTeX commentée d'une session MAX qui calcule le transfert du graphe d'événements temporisé traité dans [23].

```
    |\^/|      MAPLE V
._|\|   |/|_. Copyright (c) 1981-1990 by the University of Waterloo.
 \  MAPLE  /  All rights reserved.  MAPLE is a registered trademark of
 <____ ____>  Waterloo Maple Software.
      |       Type ? for help.
                    MAX Version 0.1
             Wed Feb 19 18:00:26 WET 1992

                        type ?MAX or help(MAX); for help
```

définition des matrices a, b, c du système (fonction de conversion p2l)

```
MAX> a:=p2l(array([[eps,g,eps],[d,eps,eps],[e,d,g*d^2]])):
MAX> b:=p2l(array([[d^3,eps],[eps,g*d],[eps,eps]])):
MAX> c:=p2l(array([[eps,g,d^3]])):
```

visualisation du résultat (prettyprint l2p)

```
MAX> l2p(a),l2p(b),l2p(c);
```

$$\begin{bmatrix} \varepsilon & \gamma & \varepsilon \\ \delta & \varepsilon & \varepsilon \\ e & \delta & \gamma\delta^2 \end{bmatrix}, \begin{bmatrix} \delta^3 & \varepsilon \\ \varepsilon & \gamma\delta \\ \varepsilon & \varepsilon \end{bmatrix}, \begin{bmatrix} \varepsilon & \gamma & \delta^3 \end{bmatrix}$$

```
MAX> time();
```

$$.816$$

calcul du transfert et affichage

```
MAX> h:=evald(c&* star(a) &* b): l2p(h);
```

$$\begin{bmatrix} \delta^8(\gamma\delta^2)^* & \gamma\delta^5(\gamma\delta^2)^* \end{bmatrix}$$

```
MAX> time();
```

$$3.183$$

Les temps CPU (en secondes) sont ceux d'une SUN-Sparcstation 2. En sus de la présentation générale qui suit, nous renvoyons le lecteur à la documentation en ligne de MAX qui décrit plus précisément les différentes fonctions et opérateurs.

1 Syntaxe générale

MAX reconnaît les types élémentaires suivants: type constante (noté const dans la suite) correspondant au dioïde $\overline{\mathbb{R}}_{\max}$, type rat (rationnels dans $\mathcal{M}_{ax}^{in}[[\gamma,\delta]]$), qui inclut les polynômes. MAX reconnaît les opérations algébriques de base suivantes:

opération	domaine
&+ (somme)	const &+ const → const rat &+ rat → rat
&* (produit)	analogue à &+
`star` (étoile)	const → const rat → rat ⟨matrice⟩ → ⟨matrice⟩
& ** (puissance)	const & ** ⟨réel ⟩ → const rat & ** ⟨ entier naturel⟩ → rat ⟨ matrice⟩& ** ⟨ entier naturel⟩ → ⟨ matrice⟩
`evald`	⟨ expression matricielle⟩ → ⟨matrice⟩

Ces opérateurs n'ont pas les ordres de priorité usuels (les opérateurs de type & s'évaluent de la gauche vers la droite en MAPLE). Ils doivent donc être parenthésés si nécessaire. Par exemple a&+b&*c s'évalue en (a&+b)&*c. Pour obtenir $a \oplus (b \otimes c)$, il faudra donc écrire a&+(b&*c). Pour les expressions algébriques matricielles, on dispose de l'analyseur `evald`, analogue à la fonction `evalm` de MAPLE. `evald` reconnaît les expressions algébriques matricielles expmat formées par la règle de grammaire suivante:

expmat → ⟨matrice⟩ |expmat +expmat | expmat &*expmat | expmat ∗∗⟨entier naturel⟩.

L'analyseur `evald` respecte les ordres de priorité usuels. Par exemple, si l'on a un système représenté par un quadruplet A, B, C, D, `H:=evald(D+C&*star(A)&*B)` calcule le transfert.

1.0.1 Remarque La somme de matrices se note "+" et non "&+". Il s'agit d'une contrainte liée à MAPLE, qui ne permet pas de mettre des ordres de priorité sur les opérateurs de type "&".

2 Opérations sur les constantes

2.1 Le dioïde $\overline{\mathbb{R}}_{\max}$

Les réels , entiers et rationnels sont représentés sous forme MAPLE standard. ε est noté `eps` et $+\infty$ est noté `inf`. e est admis comme synonyme de 0.

```
MAX>2 &+ 2;
                                        2
MAX> 0&* 1;
                                        1
MAX> e&+ 2;
                                        2
MAX> (2/3) &+ (4/5);
                                       4/5
MAX> 4 &* inf;
                                       inf
MAX> 2/3&+ eps;
                                       2/3
MAX> eps&* inf;
                                       eps
MAX> eps&* 4.5;
                                       eps
MAX> 4.5&* 10;
                                       14.5
MAX> 10&+ 7&* 2;
                                        12
MAX> 10&+ (7&* 2);
                                        10
MAX> star(0);
                                        0
MAX> star(-1);
                                        0
MAX> star(e);
                                        e
MAX> star(2);
                                       inf
```

2.2 Opérations matricielles dans $\overline{\mathbb{R}}_{\max}$

La session suivante se passe de commentaires.

```
MAX> a:=array([[1,2],[4,7]]);
```

$$a := \begin{bmatrix} 1 & 2 \\ 4 & 7 \end{bmatrix}$$

```
MAX> b:=array([[0,5],[-6,-1]]);
```

$$b := \begin{bmatrix} 0 & 5 \\ -6 & -1 \end{bmatrix}$$

```
MAX> s:=evald(a+b);
```

$$s := \begin{bmatrix} 1 & 5 \\ 4 & 7 \end{bmatrix}$$

```
MAX> p:=evald(a&*b);
```

$$p := \begin{bmatrix} 1 & 6 \\ 4 & 9 \end{bmatrix}$$

```
MAX> t:=tr(p);
```

$$t := 9$$

```
MAX> a0:=star(a);
```

$$\begin{bmatrix} +\infty & +\infty \\ +\infty & +\infty \end{bmatrix}$$

```
MAX> c1:=star(b);
```

$$c1 := \begin{bmatrix} 0 & 5 \\ -6 & 0 \end{bmatrix}$$

```
MAX> id(2);
```

$$\begin{bmatrix} e & \varepsilon \\ \varepsilon & e \end{bmatrix}$$

```
MAX> c2:=evald(id(2)+ b+ b**2);
```

$$c2 := \begin{bmatrix} 0 & 5 \\ -6 & 0 \end{bmatrix}$$

2.2.1 Fonctions `eigenm`, `eigenv` La fonction `eigenm` (pour eigen-matrix) calcule la somme

$$\texttt{eigenm}(A) = A \oplus (A^{\otimes 2})^{\odot \frac{1}{2}} \oplus \ldots \oplus (A^{\otimes (n)})^{\odot \frac{1}{n}} \ ,$$

où les puissances fractionnaires s'entendent au sens du produit de Hadamard du dioïde, ce qui se réécrit comme suit dans l'algèbre habituelle:

$$\texttt{eigenm}(A) = \max_{1 \le i \le n} (\frac{1}{i} \times A^{\otimes i}) \ .$$

La fonction `eigenv` retourne la trace de `eigenm`(A), c'est à dire la plus grande valeur propre de A, $\rho(A)$. Rappelons en effet que $\rho(A)$ est égal au poids moyen maximal des circuits de A. Comme $\mathrm{tr}((A^{\otimes i})^{\odot \frac{1}{i}})$ est égal au poids moyen maximal des circuits de longueur i, on a clairement $\rho(A) = \mathtt{tr}(\mathtt{eigenm}(A))$.

Pour la matrice `a` définie plus haut, on a ainsi:

```
MAX> a1:=eigenm(a);
```

$$a1 := \begin{bmatrix} 3 & 9/2 \\ 11/2 & 7 \end{bmatrix}$$

```
MAX> k:=eigenv(a);
```

$$k := 7$$

```
MAX> d0:=array([[0,4,inf,eps],[-70,0,inf ,35],[eps,eps,0,inf],[eps,eps,12,0]]);
```

$$d0 := \begin{bmatrix} 0 & 4 & +\infty & \varepsilon \\ -70 & 0 & +\infty & 35 \\ \varepsilon & \varepsilon & 0 & +\infty \\ \varepsilon & \varepsilon & 12 & 0 \end{bmatrix}$$

```
MAX> d1:=eigenm(d0);
```

$$d1 := \begin{bmatrix} 0 & 4 & +\infty & +\infty \\ -\frac{35}{2} & 0 & +\infty & +\infty \\ \varepsilon & \varepsilon & +\infty & +\infty \\ \varepsilon & \varepsilon & +\infty & +\infty \end{bmatrix}$$

2.2.2 La fonction `karp` La fonction `karp` calcule la valeur propre $\rho(A)$ dans $\mathbb{R}_{\max}$ d'une matrice A irréductible en utilisant l'algorithme de Karp [54]. $\rho(A)$ est donné par la formule suivante

$$\rho(A) = \bigoplus_{1\leq i\leq n} \bigwedge_{0\leq k\leq n-1} \left(\frac{A_{1i}^n}{A_{1i}^k}\right)^{\frac{1}{n-k}} ,$$

ou avec les notations usuelles:

$$\rho(A) = \max_{1\leq i\leq n} \min_{0\leq k\leq n-1} \frac{A_{1i}^{\otimes n} - A_{1i}^{\otimes k}}{n-k} .$$

Si A n'est pas irréductible, `karp` retourne la valeur propre de la composante irréductible associée à l'indice 1.

Pour les matrices traitées plus haut, on a par exemple:

```
MAX> k:=karp(a);
```

$$k := 7$$

```
MAX> l:=karp(d0);
```

$$l := 0$$

3 Calcul rationnel dans $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[[\gamma,\delta]]$

3.1 Polynômes

Nous avons adopté une représentation des polynômes de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$ sous forme de liste, distincte des polynômes de MAPLE.

Le type poly, représentant les polynômes de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$, est défini comme suit:
poly= $\langle$liste de monômes$\rangle$ | `eps` | e.

Un monôme est une liste [t,n] formée de deux entiers positifs ou nuls t et n, correspondant à $\gamma^n\delta^t$. On notera que la liste vide [] (formée de 0 monômes) est synonyme de eps. La liste [[0,0]] correspondant au monôme $\gamma^0\delta^0$ est un synonyme de l'unité e.

3.1.1 Fonctions de conversion `p2l` et `l2p` La fonction de conversion `p2l`, (pour "polynomials to lists") convertit un polynôme conventionnel en g et d en une liste de type poly mise sous forme minimale. Les indéterminées g et d correspondent respectivement aux indéterminées γ et δ de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$. La fonction inverse `l2p` convertit une liste représentant un polynôme de $\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta]$ en polynôme classique en g et d. Les fonctions `p2l` et `l2p` ne sont pas exactement inverses l'une de l'autre. On a `p2l`$\circ$ `l2p`$\preceq$ Id et `l2p`$\circ$ `p2l`$\preceq$ Id. En effet, `p2l` normalise les polynômes: étant donné un polynôme $q \in \mathbb{B}[\gamma,\delta]$, `l2p`$\circ$`p2l`$(q)$ est égal au représentant minimal de q. De même, l étant une liste de type poly `p2l`$\circ$`l2p`(l) est égal à la liste minimale représentant l.

On a par exemple:

```
MAX> p2l(e);
```

$$[[0,0]]$$

```
MAX> p2l(1);
```

$$[[0,0]]$$

```
MAX> a:=[[0,1],[2,3],[10,9]];
```

$$a := [[0,1],[2,3],[10,9]]$$

```
MAX> v:=l2p(a);
                          3  2     9  10
                v := g + g  d  + g  d
```

Dans la suite, pour des raisons de lisibilité, on donnera systématiquement les transcriptions LaTeX de la session. Au lieu de la copie d'écran MAPLE ci-dessus, on aura donc:

$$v := \gamma \oplus \gamma^3\delta^2 \oplus \gamma^9\delta^{10}$$

```
MAX> b:=p2l(v);
```

$$b := [[0,1],[2,3],[10,9]]$$

```
MAX> w:=g+g^3*d^2+g^4*d+g^9*d^10;
```

$$\gamma \oplus \gamma^3\delta^2 \oplus \gamma^4\delta \oplus \gamma^9\delta^{10}$$

```
MAX> c:=p2l(w);
```

$$c := [[0,1],[2,3],[10,9]]$$

```
MAX> l2p(c);
```

$$\gamma \oplus \gamma^3\delta^2 \oplus \gamma^9\delta^{10}$$

Le monôme redondant $\gamma^4\delta$ a été éliminé lors du calcul de `p2l(w)`.

```
MAX> f:=p2l(g+g^3*d^2+g^4*d+g^9*d^10)&*p2l(g*d^5);
```

$$f := [[5,2],[7,4],[15,10]]$$

```
MAX> l2p(f);
```

$$\gamma^2\delta^5 \oplus \gamma^4\delta^7 \oplus \gamma^{10}\delta^{15}$$

```
MAX> f&+ p2l(d^9);
```

$$[[9,0],[15,10]]$$

```
MAX> l2p(");
```

$$\delta^9 \oplus \gamma^{10}\delta^{15}$$

```
MAX> h1:=f&* p2l(g*d^5):l2p(h1);
```

$$\gamma^3\delta^{10} \oplus \gamma^5\delta^{12} \oplus \gamma^{11}\delta^{20}$$

```
MAX> h2:=f&* p2l(g^2*d^10):l2p(h2);
```

$$\gamma^4\delta^{15} \oplus \gamma^6\delta^{17} \oplus \gamma^{12}\delta^{25}$$

```
MAX> l2p(h1&+ h2);
```

$$\gamma^3\delta^{10} \oplus \gamma^4\delta^{15} \oplus \gamma^6\delta^{17} \oplus \gamma^{11}\delta^{20} \oplus \gamma^{12}\delta^{25}$$

```
MAX> l2p(f&* p2l(g*d^5+g^2*d^10));
```

$$\gamma^3\delta^{10} \oplus \gamma^4\delta^{15} \oplus \gamma^6\delta^{17} \oplus \gamma^{11}\delta^{20} \oplus \gamma^{12}\delta^{25}$$

```
MAX> l2p(f &+ []);
```

$$\gamma^2\delta^5 \oplus \gamma^4\delta^7 \oplus \gamma^{10}\delta^{15}$$

```
MAX> f&* [];
```

$$\varepsilon$$

```
MAX> f&* eps;
```

$$\varepsilon$$

```
MAX> l2p(f&* e);
```

$$\gamma^2\delta^5 \oplus \gamma^4\delta^7 \oplus \gamma^{10}\delta^{15}$$

```
MAX> l2p(f&* [[0,0]]);
```

$$\gamma^2\delta^5 \oplus \gamma^4\delta^7 \oplus \gamma^{10}\delta^{15}$$

3.1.2 Fonction `canonic` La fonction `canonic` met une liste de type poly ou rat sous forme minimale. On a l'identité `canonic` = `p2l`∘`l2p`.

```
MAX> u:=[[0,1],[1,0]];
```

$$u := [[0,1],[1,0]]$$

```
MAX> l2p(u);
```

$$\gamma \oplus \delta$$

```
MAX> v:=canonic(u);
```

$$v := [[1,0]]$$

```
MAX> l2p(v);
```

$$\delta$$

```
MAX>p2l(l2p(u));
```

$$[[1,0]]$$

```
MAX>s:=[[[0,0],[1,1]],[[1,1]]];
```

$$s := [[[0,0],[1,1]],[[1,1]]]$$

```
MAX> l2p(s);
```

$$e \oplus \gamma\delta\,(\gamma\delta)^*$$

```
MAX> t:=canonic(s);
```

$$t := [[[0,0]],[[1,1]]]$$

```
MAX> l2p(t);
```

$$(\gamma\delta)^*$$

3.1.3 Remarque La représentation sous forme de listes des polynômes est du d'une part à l'accès peu aisé à la représentation interne des polynômes MAPLE, et d'autre part à la nécessité de normaliser tous les objets avant calcul. La normalisation est faite une fois pour toutes par `p2l` lors de la première définition d'un objet.

3.2 Séries périodiques

3.2.1 Représentation des séries périodiques Les séries périodiques sont, tout comme les polynômes, représentées par des listes. Le type série rationnelle rat est défini comme suit:

rat= poly | [⟨liste de monômes⟩,⟨liste de monômes⟩].

Une série périodique non polynômiale est donc une liste $[p,q]$ formée de deux polynômes non nuls. Soient

$$p = \gamma^{n_1}\delta^{t_1} \oplus \ldots \oplus \gamma^{n_k}\delta^{t_k}$$

le premier polynôme de la liste, et $q = \gamma^{n'_1}\delta^{t'_1} \oplus \ldots \oplus \gamma^{n'_r}\delta^{t'_r}$. La série périodique s associée à $[p, q]$ est donnée par:

$$s = \left(\bigoplus_{i=1}^{k-1} \gamma^{n_i}\delta^{t_i}\right) \oplus \gamma^{n_k}\delta^{t_k} \left(e \oplus \bigoplus_{j=1}^{r-1} \gamma^{n'_j}\delta^{t'_j}\right) (\gamma^{n'_r}\delta^{t'_r})^* \ .$$

Le dernier monôme de p est donc égal au premier monôme du périodique. Le dernier monôme de q est égal à la période.

3.2.2 Définition des séries rationnelles On peut définir une série rationnelle à l'aide de la fonction `p2l` qui reconnaît une expression rationnelle "exprat" en g et d, définie par la règle de grammaire suivante:

exprat $\rightarrow$ e | g | d | exprat + exprat | exprat $*$ exprat | exprat^'*'.

On notera que l'étoile de a se note a^'*' (* entre guillemets)[1]. On a par exemple: `MAX> s1:=e+g^2*d^2+g^3*d^4*(e+g^2*d+g^3*d^3+g^5*d^4)*(g^6*d^6)^'*';`

$$e \oplus \gamma^2\delta^2 \oplus \gamma^3\delta^4 \left(e \oplus \gamma^2\delta \oplus \gamma^3\delta^3 \oplus \gamma^5\delta^4\right) \left(\gamma^6\delta^6\right)^*$$

```
MAX> s:=p2l(s1);
```

$$s := [[[0,0],[2,2]],[[2,1],[3,3]]]$$

```
MAX> l2p(s);
```

$$e \oplus \gamma^2\delta^2 \left(e \oplus \gamma\delta^2\right) \left(\gamma^3\delta^3\right)^*$$

```
MAX> l2p(s[1]);
```

$$e \oplus \gamma^2\delta^2$$

```
MAX> l2p(s[2]);
```

$$\gamma\delta^2 \oplus \gamma^3\delta^3$$

On notera que la série est automatiquement mise sous forme minimale. Voici un exemple où `p2l` calcule une expression plus complexe:

```
MAX> s:=d^8+(g^3+(g^3*d^3+g^2*d^2)^'*')^'*';
```

$$\delta^8 \oplus \left(\gamma^3 \oplus \left(\gamma^3\delta^3 \oplus \gamma^2\delta^2\right)^*\right)^*$$

```
MAX> s1:=p2l(s);
```

$$s1 := [[[8,0],[9,9]],[[1,1]]]$$

```
MAX> l2p(s1);
```

$$\delta^8 \oplus \gamma^9\delta^9 (\gamma\delta)^*$$

3.2.3 Opérations &+,&* et `star` Les expressions rationnelles se calculent soit à partir de `p2l`, soit à l'aide des opérateurs &+,&*,et `star` qui s'utilisent comme pour les constantes.

```
MAX>u:=p2l(g)&* star(p2l(g^3*d^3));
```

$$u := [[[0,1]],[[3,3]]]$$

```
MAX> l2p(u);
```

$$\gamma \left(\gamma^3\delta^3\right)^*$$

```
MAX> v:=star(u);
```

$$v := [[[0,0],[3,4]],[[3,3]]]$$

```
MAX> l2p(v);
```

$$e \oplus \gamma^4\delta^3 \left(\gamma^3\delta^3\right)^*$$

```
MAX> w:=p2l(d^9)&+ v: l2p(w);
```

$$\delta^9 \oplus \gamma^{13}\delta^{12} \left(\gamma^3\delta^3\right)^*$$

[1] pour des raisons de syntaxe MAPLE

3.3 Validation de MAX

Nous avons, afin de vérifier systématiquement les algorithmes, programmé des fonctions générant des séries périodiques aléatoires.

3.3.1 Fonctions `randpol`, `randser`, `randlist`

- La fonction `randpol`, prenant pour argument un entier n, produit un polynôme de $\mathcal{M}_{\rm ax}^{\rm in}[\gamma,\delta]$ aléatoire dont le représentant minimal a entre 1 et n termes. La syntaxe sans arguments `randpol()` prend la valeur par défaut $n = 10$.
- La fonction `randser` produit une série périodique aléatoire dont le transitoire a entre 0 et n-1 termes, et le motif entre 1 et n termes. Appelé sans argument, `randser()` prend la valeur par défaut $n = 10$.
- La fonction `randlist` produit une liste non minimale de type poly formée de n termes. Par défaut, $n = 10$.

On a par exemple la session suivante:

`MAX> p:=randpol();`

$$[[2,5],[9,9],[12,11],[16,12]]$$

`MAX> l2p(p);`

$$\gamma^5\delta^2 \oplus \gamma^9\delta^9 \oplus \gamma^{11}\delta^{12} \oplus \gamma^{12}\delta^{16}$$

`MAX> q:=randpol(3);`

$$[[1,2],[2,3]]$$

`MAX> l2p(q);`

$$\gamma^2\delta \oplus \gamma^3\delta^2$$

`MAX> s:=randser();`

$[[[7,7],[10,15],[12,20],[18,27],[26,33],[33,34],[36,35]],[[6,5],[16,6],[26,11],[33,19],$
$[40,20],[49,21],[53,28],[55,38],[65,46]]]$

`MAX> l2p(s);`

$\gamma^7\delta^7 \oplus \gamma^{15}\delta^{10} \oplus \gamma^{20}\delta^{12} \oplus \gamma^{27}\delta^{18} \oplus \gamma^{33}\delta^{26} \oplus \gamma^{34}\delta^{33} \oplus$
$\gamma^{35}\delta^{36}\left(e \oplus \gamma^5\delta^6 \oplus \gamma^6\delta^{16} \oplus \gamma^{11}\delta^{26} \oplus \gamma^{19}\delta^{33} \oplus \gamma^{20}\delta^{40} \oplus \gamma^{21}\delta^{49} \oplus \gamma^{28}\delta^{53} \oplus \gamma^{38}\delta^{55}\right)\left(\gamma^{46}\delta^{65}\right)^*$

`MAX> r:=randser(3);`

$$[[[3,2],[4,4],[5,7]],[[1,3]]]$$

`MAX> l2p(r);`

$$\gamma^2\delta^3 \oplus \gamma^4\delta^4 \oplus \gamma^7\delta^5\left(\gamma^3\delta\right)^*$$

`MAX> randlist(3);`

$$[[3,1],[3,3],[1,3]]$$

`MAX> l2p(");`

$$\gamma\delta^3 \oplus \gamma^3\delta^3 \oplus \gamma^3\delta$$

`MAX> randlist();`

$$[[4,1],[10,2],[10,8],[4,6],[5,2]]$$

```
MAX> l2p(");
```

$$\gamma\delta^4 \oplus \gamma^2\delta^{10} \oplus \gamma^8\delta^{10} \oplus \gamma^6\delta^4 \oplus \gamma^2\delta^5$$

3.3.2 Vérification des algorithmes de somme et produit de polynômes

Nous avons effectué le test de cohérence suivant:

- générer deux polynômes non minimaux p et p' par `randlist`.
- calculer `canonic`(p)&+`canonic`(p'), `canonic`(p)&*`canonic`(p')
- comparer respectivement avec `p2l(l2p(`p`) + l2p(`p'`))` et `p2l(l2p(`p`) * l2p(`p'`))`, où + et $*$ désignent la somme et le produit des polynômes MAPLE standards.

Ce test revient à vérifier la commutativité du diagramme suivant:

$$\begin{tikzcd}
\mathbb{B}[\gamma,\delta] \arrow[r] \arrow[d, "{p,\, p'}"] & \mathbb{B}[\gamma,\delta] \arrow[d, "{\begin{array}{l} p \oplus p' \\ p \otimes p' \end{array}}"] \\
\mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta] \arrow[r] & \mathcal{M}_{\mathrm{ax}}^{\mathrm{in}}[\gamma,\delta] \\
\overline{p}, \overline{p'} & \begin{array}{l} \overline{p} \oplus \overline{p'} = \overline{p \oplus p'} \\ \overline{p \otimes p'} = \overline{p} \otimes \overline{p'} \end{array}
\end{tikzcd}$$

3.3.3 Vérification des algorithmes de somme et produit de séries rationnelles

Nous avons effectué le test de cohérence suivant:

(i) générer deux séries aléatoires s et s',

(ii) calculer $s \oplus s'$, ss',

(iii) calculer p, le développement de s à un ordre élevé et p', développement de s' au même ordre,

(iv) s'assurer que les développements de $s \oplus s'$ et ss' coïncident respectivement $p \oplus p'$ et pp', à des ordres convenables.

Si l'on suppose correctes les opérations sur les polynômes, le succès de ce test "garantit" la validité des algorithmes sur les séries rationnelles. Ce test a été passé pour quelques centaines de séries aléatoires produites par `randser`(15). En ce qui concerne les étoiles, on s'est borné à vérifier l'identité $(a^*)^* = a^*$ pour des séries rationnelles aléatoires.

4 Exemple: calcul du transfert d'un atelier flexible de type flowshop

On considère l'atelier flexible de type flowshop avec trois pièces et trois machines, représenté par le réseau de Petri sur la Figure VIII.1. On donne les temps suivants de passage des pièces sur les machines:

	P1	P2	P3
M1	9	2	1
M2	2	7	2
M3	1	10	1

Une pièce de type 1, convoyée par une palette, passe donc pendant 9 unités de temps sur la machine 1, puis 2 unités de temps sur la machine 2, et 1 unité de temps sur la machine 3. Elle sort alors de l'atelier, et la palette revient charger une nouvelle pièce en amont de la machine 1. De même pour les pièces 2 et 3. Pendant ce temps, la machine 1 traite les 3 pièces suivant la séquence (1 pièce de type 1, 1 pièce de type 2, 1 pièce de type 3) et recommence. On notera que les circuits horizontaux sur le graphe correspondent à des séquences de tâches des machines, et que le circuits verticaux, dits "circuits pièce", correspondent au trajet des palettes. On peut écrire la représentation matricielle suivante.

$$A = \begin{bmatrix} \varepsilon & \varepsilon & \gamma\delta & \varepsilon & \varepsilon & \varepsilon & \gamma\delta & \varepsilon & \varepsilon \\ \delta^9 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^{10} & \varepsilon \\ \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma \\ \delta^9 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta^2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \delta^2 & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \delta & \varepsilon & \delta^7 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \gamma\delta \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^7 & \varepsilon & \delta & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^2 & \varepsilon & \delta^{10} & \varepsilon \end{bmatrix}$$

$$B = \begin{bmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

$$C = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta^{10} & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \delta \end{bmatrix}$$

où les entrées u_1, u_2, u_3 représentent les approvisionnements relatifs au pièces 1,2,3 et les sorties y_1, y_2, y_3 représentent les sorties de ces pièces[2] On obtient alors à l'aide de `MAX` la matrice de transfert:

$$H = \begin{bmatrix} \delta^{12} \oplus \gamma\delta^{30}(\gamma\delta^{19})^* & \gamma\delta^{21}(\gamma\delta^{19})^* & \gamma\delta^{13} \oplus \gamma^2\delta^{31}(\gamma\delta^{19})^* \\ \delta^{28}(\gamma\delta^{19})^* & \delta^{19}(\gamma\delta^{19})^* & \gamma\delta^{29}(\gamma\delta^{19})^* \\ \delta^{29}(\gamma\delta^{19})^* & \delta^{20}(\gamma\delta^{19})^* & \delta^4 \oplus \gamma\delta^{30}(\gamma\delta^{19})^* \end{bmatrix}$$

On constate que la période $\gamma\delta^{19}$ correspond à l'unique circuit critique (associé à la pièce 2). Les deux machines menantes, à savoir les machines 1 et 3, dont le temps de cycle est de 12 unités de temps, ne sont donc pas saturées. Nous étudierons des systèmes analogues dans le chapitre IX sur l'optimisation des ressources. Nous montrerons en particulier comment saturer les machines menantes en rajoutant des palettes.

[2] Les matrices (A, B, C) peuvent être produites automatiquement à partir de la matrice des temps de passage des pièces sur les machines à l'aide de l'instruction `array2flex`, décrite au Chapitre suivant.

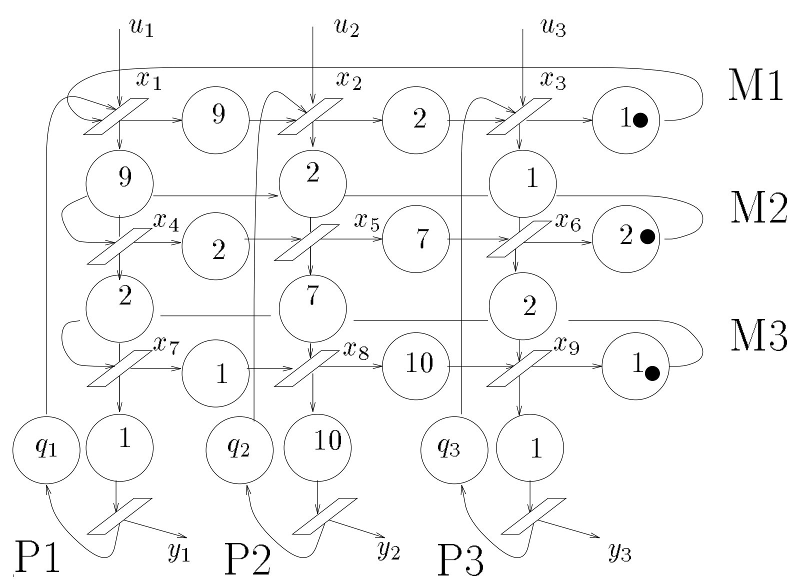

Figure VIII.1: Un atelier flexible à 3-machines et 3-pièces