Notes du cours de **Calcul Formel** MI 2013.

G. H. E. DUCHAMP *

15-02-2013 06:07

Table des matières

1 Préambule **3**

2 Introduction **4**

3 Systèmes de Calcul Formel et structures de données **5**
3.1 Révision numération 5
3.1.1 Quelques exercices 5
3.2 Exemples de sessions 6
3.2.1 Interpréteur 6
3.2.2 Programmation 10
3.3 Quelques exercices et problèmes de combinatoire 11
3.4 Systèmes de Calcul Formel 15
3.5 Prise en main de Maple : T.D. 16
3.6 Séries à une variable (univariées) 17
3.6.1 Généralités 17
3.6.2 Les monômes 17
3.6.3 Les séries à une variable 17
3.7 Séries liées à des statistiques 18
3.7.1 Un exemple : une stastistique bivariée productrice d'un codage 18
3.7.2 Mots de Dyck & Arbres binaires complets 20
3.8 Produit scalaire $\langle$série|polynôme$\rangle$ et premières opérations 21

4 Séries d'une variable ($\mathbb{C}[[z]]$) **21**
4.1 Opérations sur les séries 22
4.2 Les deux produits : Convolution (produit de Cauchy) et produit de Hadamard . 22
4.2.1 Produit de Cauchy 22
4.2.2 Séries rationnelles 23
4.2.3 Séries rationnelles (repésentations linéaires et aspect automatique) . . . 27
4.2.4 Produit de Hadamard 28

* Prière de signaler les erreurs à ADRIAN TANASA ou à moi-même.

5 Génération aléatoire **29**
5.1 Engendrer le hasard . . . 29
5.2 Générateurs à un pas . . . 29
5.2.1 Paramètres . . . 30
5.2.2 Algorithmes de Brent et Floyd . . . 32
5.2.3 Générateurs congruentiels linéaires . . . 33
5.3 Générateurs à deux pas . . . 35
5.3.1 Vectorisation et paramètres . . . 35
5.4 Générateurs du type GL2P . . . 36
5.4.1 Généralités . . . 36
5.4.2 Combinaison de deux générateurs . . . 36
5.4.3 Décomposition et calcul de la période d'un GL2P ($m = p$ premier). . . . 36
5.4.4 Carrés et équations du second degré dans $\mathbb{F}_p$ . . . 37
5.4.5 Calcul de la période d'un GCL2 . . . 37
5.5 Autres générateurs . . . 39
5.6 Généateurs à k pas . . . 39
5.7 Énumérer, classer, indexer . . . 39
5.8 Répartitions équitables et moins équitables . . . 39

6 Représentation des données et calcul **39**
6.1 Généralités . . . 39

7 Premiers pas & Résumé des séances **39**

8 Systèmes et Calcul **40**
8.1 Introduction . . . 40
8.2 Description de la structure d'automate . . . 40
8.2.1 Graphe pondéré . . . 40
8.2.2 Structure et comportement des automates . . . 41
8.2.3 Premiers automates . . . 42
8.2.4 Composition des automates . . . 42

9 Séries **44**
9.1 Introduction . . . 44
9.2 Fonctions : notation fonctionnelle et sommatoire . . . 44

10 Séries de plusieurs variables **45**
10.1 Les scalaires de l'Informatique . . . 45
10.2 Définition d'un semi-anneau . . . 45
10.3 Automates à multiplicités . . . 46
10.3.1 Graphes pondérés (structure de transition) . . . 46
10.3.2 Structure et comportement des automates . . . 47
10.3.3 Premiers automates . . . 48
10.4 Le décalage . . . 48
10.4.1 Formules liant le décalage et les autres opérateurs . . . 48
10.4.2 Espaces stables et matrice de décalage . . . 48
10.5 Types courants (de séries) . . . 49
10.5.1 Composition des automates . . . 50

10.6 Modules 52
10.6.1 Le bi-module des fonctions $X \mapsto K$ 52
10.7 Cas où X est un monoïde 52
10.7.1 Décalages 53
10.7.2 Convolution 53
10.7.3 Monoïdes localement finis 53
10.8 Cas du monoïde libre, théorème de Kleene-Schützenberger 53

11 La structure d'AF **53**
11.1 Définition et fonctionnement 53
11.2 Graphe et représentations linéaires 53

12 Calcul du langage reconnu par un AF **53**
12.1 Automates finis et structures de transition 53
12.2 Étoile d'une matrice-lettres 54

13 Fonctions sur les mots **54**
13.1 Quelques exemples de machines 55

14 Systèmes et Calcul **55**
14.1 Introduction 55
14.2 Description de la structure d'automate 55
14.2.1 Graphe pondéré 55
14.2.2 Structure et comportement des automates 56
14.2.3 Premiers automates 57

15 Résumé des cours et TD. **57**
15.0.4 Composition des automates 58

1 Préambule

Ce cours est à la fois théorique et appliqué.
Il est conçu de façon à permettre aux étudiants des deux sensibilités de pouvoir s'entraîner tant au niveau de la programmation (TP - TD - libre service) qu'au niveau conceptuel : outils analytiques et algébriques de l'informatique théorique &/ou utilisés en informatique moderne (sécurité, simulation, algorithmique rapide, compression audio et video).
Les T.D. se font en Maple et Maxima, mais le contenu de l'enseignement est, bien sûr, indépendant du langage.

Le poly correspond à un programme maximal. Seule une partie de celui-ci a été traitée cette année.
Les exercices qui sont là pour aider à la compréhension du cours.
II se peut que vous ne compreniez pas certains énoncés, dans ce cas choisissez que les questions que vous êtes capables d'aborder ... et contactez-moi rapidement pour le décodage du reste.
(email `gheduchamp@gmail.com`) pour le décodage du reste.

Les parties en petits caractères sont des suppléments et ne sont pas obligatoires.

2 Introduction

Le *Calcul Symbolique* est l'art de manipuler (scientifiquement) les symboles exacts :

$$4/7,\ \pi^2/6,\ \sqrt{10}$$

ou bien littéraux

$$x,y,z,t,u,v$$

selon certaines règles dites "de calcul" (ou bien de dérivation[1]). En fait, cette activité est très ancienne et remonte à la numération puisque celle-ci consiste à symboliser des quantités par des symboles et que les quatre opérations arithmétiques ne sont autres que le calcul symbolique attaché à des problèmes concrets (ajout ou retrait de quantités, calcul de longueur, de surface, de volume, mesure d'une grandeur) et donnent lieu aux alogorithmes de l'arithmétique élémentaire.
De même que la mathématique (quand elle ne se consacre pas à ses propres problèmes) développe des modèles pour les sciences de la nature (équations de la physique, lois ..), de même l'Informatique Théorique a développé des modèles et des concepts pour les ordinateurs (machines de Turing, calculabilité, automates, séries génératrices, complexité, grammaires..).
Le *Calcul Formel*[2] est né dès que l'on a essayé de traiter automatiquement certains calculs trop compliqués ou fastidieux pour être élaborés à la main[3]. Il se démarque du *Calcul Numérique* en ceci qu'il est un calcul exact (c'est à dire sans perte d'information due aux erreurs d'arrondi). Par exemple, si l'on veut faire mécaniquement les quatre opérations avec les fractions et $\sqrt{2}$, il faut utiliser la structure de donnée $a + b\sqrt{2};\ a,b \in \mathbb{Q}$, il n'y a pas là d'arrondi. Le problème principal du Calcul Formel est l'explosion des données en cours de calcul (c'est le prix à payer pour l'exactitude). Ansi toute l'arithmétique et ses applications[4] nécessite le calcul exact, en effet une erreur d'un digit même sur un nombre de 250 chiffres peut transformer un nombre premier en un nombre hautement factorisable (et, dans tous les cas un nombre pair en nombre impair !).qq5R54N Ceci est génant pour les tests de parité, par exemple dans les modems : le *bit de parité*, une forme très rudimentaire de code détecteur d'erreurs.
Le Calcul Symbolique peut ici être vu comme la science qui va traiter à la fois des structures de donnés du Calcul Formel et du comportement de ceux-ci (cf infra, la méthode Symbolique en complexité).

Pour résumer, le voisinage scientifique du Calcul Symbolique se compose ainsi :

- le calcul scientifique exact
- le calcul formel (et, concrètement, les systèmes de Calcul Formel)
- l'informatique théorique
- les systèmes formels
- la combinatoire

les sciences voisines sont l'informatique, les mathématiques, la physique et, depuis peu, la chimie et la biologie.

1. On parle de *règle de dérivation* en logique formelle ou en réécriture et d'*arbre de dérivation* dans les théories des grammaires et des systèmes formels
2. Computer Algebra en anglais.
3. En anglais : hand and paper computation.
4. comme le codage, décodage, cryptage, la sécurité des transmissions par exemple..

3 Systèmes de Calcul Formel et structures de données

3.1 Révision numération

Il est bon, avant d'attaquer, de maîtriser la numération en base quelconque. Voici quelques exercices.

3.1.1 Quelques exercices

Objectifs : Il faut que les étudiants soient maîtres (papier crayon et programmation) des conversions entre bases (avec virgule) et du passage (fraction $\leftrightarrow$ développement périodique). La commande maple pour vérifier les conversions est à chercher dans `convert`, celle pour les développement illimités dans `evalf` (attention à la variable d'environnement `Digits`).

1) Mettre les nombres binaires suivants sous forme décimale
a) $(101101)_2$ b)$(101101101)_2$ c) $(\underbrace{101\cdots101}_{3n \text{ chiffres}})_2$ (pour le (c), on montrera que la réponse dépend de la conversion d'un nombre plus simple)
2) Mettre les fractions suivantes sous forme décimale
a) $(0{,}615)_8$ b) $(12{,}321)_5$ c) $(0,\underbrace{7777777777}_{10 \text{ chiffres}})_8$
3) Calculer
a) $a_n = (0,\underbrace{5\cdots5}_{n \text{ chiffres}})_8$ b) $b_n = (12,\underbrace{1212\cdots12}_{2n \text{ chiffres}})_8$ c) $c_n = (12,\underbrace{2112\cdots2112\cdots}_{n \text{ chiffres}})_8$
d) $f_1 = (0{,}(54321)^\infty)_8$
e) le développement décimal illimité de
i) $23/7$ ii) $7/22$ iii) $1/99999$

4) Conversions Fraction $\leftrightarrow$ Développements illimités en base b (les résultats seront toujours donnés en base dix sauf pour les points c,e).
a) $12{,}(345)^\infty;\ b=10$ b) $12{,}(345)^\infty;\ b=8$
c) $BA/CA;\ b=16$ d) $22/132;\ b=10$ e) $BA{,}(CA)^\infty;\ b=16$ Calcul formel=calcul scientifique exact. Notions reliées: calcul symbolique, algorithmique algébrique, calcul numérique stable. Objets manipulés:

1. Toutes les structures de données classiques:
   arbres, listes, permutations, graphes, ensembles, intervalles, tables, tableaux, mots, partitions, compositions,partitions.
2. Les quantités du calcul scientifique:
   nombres, radicaux, polynômes, fractions rationnelles, matrices, vecteurs, tenseurs, fonctions, séries.
3. Les opérateurs pour ces structures:

   (a) Pour les structures de données : Manipulation d'arbres (sous arbres, etc...), retirer ou ajouter un élément dans une liste ou un ensemble, un sommet ou une arête dans un graphe, concaténer ou simplifiere mots (fusion, réunion, contraction etc...).
   (b) Pour les quantités du calcul scientifique: dérivation, intégration et sommation symboliques. équations diff, développements limités ou en série.

4. Résolution des équations posées dans le cadre précédent.

Bien entendu, un système de calcul formel comprend aussi des interfaces graphiques, texte etc....

On peut représenter toutes sortes de formules et on est très vite confronté aux problèmes d'égalité (i.e. reconnaître les doubles représentations). Par exemple on a

$$\sqrt{2}+\sqrt{3}+\sqrt{5}=\sqrt{10+2\sqrt{10}+2\sqrt{3}\sqrt{7+2\sqrt{10}}} \tag{1}$$

comment le système peut-il s'en appercevoir? Plus difficile, comment le système peut-il donner la premier membre à partir du second? (cf Exo) ($\hookrightarrow$ forme normale, canonique, simplification automatique).

: Remarquer que:

```
> A := sqrt(7+2*sqrt(10));
```

$$A := \sqrt{7+2\sqrt{10}}$$

```
> simplify(A);
```

$$\sqrt{2}+\sqrt{5}$$

Mais

```
> B := sqrt(10+2*sqrt(10)+2*sqrt(3)*sqrt(7+2*sqrt(10))): simlify(B);
```

$$\sqrt{10+2\sqrt{2}\sqrt{5}+2\sqrt{3}\sqrt{5}+2\sqrt{3}\sqrt{2}}$$

Vaut-il mieux représenter une fraction sous la forme $\frac{p}{q}$ ou par son développement décimal (qui est complètement représentable en machine), comment passer d'une représentation à l'autre?

Combinatoire. —

C'est l'art de manipuler ls structures de données (discrètes ou continues, mais toujours représentables en machine et calculables).

Dénombrer, dessiner, calculer, implémenter sont la pain quotidien du combinatoriste-praticien.

3.2 Exemples de sessions

3.2.1 Interpréteur

p17[4]

```
> c:= [1,2,3,2,1];
```

```
c:= [1,2,3,2,1]
```

```
> nops(c);
```

```
5
```

```
> op(2,c);
```

```
2
```

```
> c[2];
```

```
2
```

```
> d:=[2,5,46];
```

```
d:=[2,5,46]
```

```
> e:=[op(c),op(d)];
```

```
e:=[1,2,3,2,1,2,5,46]
```

```
> map(x->x^2,c);
```

```
e:=[1,4,9,4,1]
```

La formule de Rodrigues pour les polynomes d'Hermite est

$$H_n(x) := (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2})$$

On peut les calculer en repetant une même commande à l'aide de `diff`. Soit

$> \; n := 0 : [n,exp(-x^2)];$

$[0,e^{-x^2}]$

$> \; n := n+1 : [n,factor(diff(""[2],x))];$

$$[1, -2\,xe^{-x^2}]$$

$> \; n := n+1 : [n,factor(diff(""[2],x))];$

$$[2,2\,e^{-x^2}\left(-1+2\,x^2\right)]$$

$> \; n := n+1 : [n,factor(diff(""[2],x))];$

$$[3, -4\,xe^{-x^2}\left(-3+2\,x^2\right)]$$

$> \; n := n+1 : [n,factor(diff(""[2],x))];$

$[4,4\,e^{-x^2}\left(3-12\,x^2+4\,x^4\right)]$

$[5, -8\,xe^{-x^2}\left(15-20\,x^2+4\,x^4\right)]$

$[6,8\,e^{-x^2}\left(-15+90\,x^2-60\,x^4+8\,x^6\right)]$

$[7, -16\,xe^{-x^2}\left(-105+210\,x^2-84\,x^4+8\,x^6\right)]$

$[8,16\,e^{-x^2}\left(105-840\,x^2+840\,x^4-224\,x^6+16\,x^8\right)]$

$[9, -32\,xe^{-x^2}\left(945-2520\,x^2+1512\,x^4-288\,x^6+16\,x^8\right)]$

$[10,32\,e^{-x^2}\left(-945+9450\,x^2-12600\,x^4+5040\,x^6-720\,x^8+32\,x^{10}\right)]$

Pour intégrer:

$> \; f := (1+x^4)^{-1};$

$$\frac{1}{x^4+1}$$

$> \; int(f,x);$

$$\sqrt{2}\ln(\frac{x^2+x\sqrt{2}+1}{x^2-x\sqrt{2}+1})1/8 + \frac{\sqrt{2}\arctan(x\sqrt{2}+1)}{4} + \frac{\sqrt{2}\arctan(x\sqrt{2}-1)}{4}$$

combinatoire

$> \; with(combinat);$

$$[Chi,bell,binomial,cartprod,character,choose,composition,conjpart,$$
$$decodepart,encodepart,fibonacci,firstpart,graycode,inttovec,$$
$$lastpart,multinomial,nextpart,numbcomb,numbcomp,numbpart,numbperm,$$

$partition$,$permute$,$powerset$,$prevpart$,$randcomb$,$randpart$,$randperm$,
$stirling1$,$stirling2$,$subsets$,$vectoint$]

$> fibonacci(43);$

$$433494437$$

$> fibonacci(43,x);$

$$1+8855\,x^4+231\,x^2+18156204\,x^{26}+38567100\,x^{24}+64512240\,x^{22}+84672315\,x^{20}+86493225\,x^{18}+$$
$$67863915,x^{16}+40116600\,x^{14}+17383860\,x^{12}+5311735\,x^{10}+1081575\,x^8+134596\,x^6+6724520\,x^{28}$$
$$+1947792\,x^{30}+435897\,x^{32}+73815\,x^{34}+9139\,x^{36}+780\,x^{38}+41\,x^{40}+x^{42}$$

$> sort(");$

$$x^{42}+41\,x^{40}+780\,x^{38}+9139\,x^{36}+73815\,x^{34}+435897\,x^{32}+1947792\,x^{30}+6724520\,x^{28}+18156204\,x^{26}+$$
$$38567100\,x^{24}+64512240\,x^{22}+84672315\,x^{20}+86493225\,x^{18}+67863915\,x^{16}+40116600\,x^{14}+$$
$$17383860\,x^{12}+5311735\,x^{10}+1081575\,x^8+134596\,x^6+8855\,x^4+231\,x^2+1$$

$> subs(x=1,");$

$$433494437$$

$> partition(8);$

[[1,1,1,1,1,1,1,1],[1,1,1,1,1,1,2],[1,1,1,1,2,2],[1,1,2,2,2],[2,2,2,2],
[1,1,1,1,1,3],[1,1,1,2,3],[1,2,2,3],[1,1,3,3],
[2,3,3],[1,1,1,1,4],[1,1,2,4],[2,2,4],[1,3,4],
[4,4],[1,1,1,5],[1,2,5],[3,5],[1,1,6],[2,6],[1,7],[8]]

taylor: Exemple de $((1-x)/(1+x))^{1/2}$

$> f := ((1+x)/(1-x))^{(1/2)};$

$$\frac{\sqrt{1+x}}{\sqrt{1-x}}$$

$> taylor(f,x=0,20);$

$$(1+x+\frac{1}{2}x^2+\frac{1}{2}x^3+\frac{3}{8}x^4+\frac{3}{8}x^5+\frac{5}{16}x^6+\frac{5}{16}x^7+\frac{35}{128}x^8+$$
$$\frac{35}{128}x^9+\frac{63}{256}x^{10}+\frac{63}{256}x^{11}+\frac{231}{1024}x^{12}+\frac{231}{1024}x^{13}+\frac{429}{2048}x^{14}+$$
$$\frac{429}{2048}x^{15}+\frac{6435}{32768}x^{16}+\frac{6435}{32768}x^{17}+\frac{12155}{65536}x^{18}+\frac{12155}{65536}x^{19}+O\left(x^{20}\right))$$

$> f1 := f/(1+x);$

$> taylor(f1,x=0,20);$

$$(1+\frac{1}{2}x^2+\frac{3}{8}x^4+\frac{5}{16}x^6+\frac{35}{128}x^8+\frac{63}{256}x^{10}+$$
$$\frac{231}{1024}x^{12}+\frac{429}{2048}x^{14}+\frac{6435}{32768}x^{16}+\frac{12155}{65536}x^{18}+O\left(x^{20}\right))$$

$> f2 := (1-x)^{(-1/2)};$

$$(1+\frac{1}{2}x+\frac{3}{8}x^2+\frac{5}{16}x^3+\frac{35}{128}x^4+\frac{63}{256}x^5+\frac{231}{1024}x^6+\frac{429}{2048}x^7+$$

$$\frac{6435}{32768}x^8+\frac{12155}{65536}x^9+\frac{46189}{262144}x^{10}+\frac{88179}{524288}x^{11}+\frac{676039}{4194304}x^{12}+\frac{1300075}{8388608}x^{13}+\frac{5014575}{33554432}x^{14}+$$
$$\frac{9694845}{67108864}x^{15}+\frac{300540195}{2147483648}x^{16}+\frac{583401555}{4294967296}x^{17}+\frac{2268783825}{17179869184}x^{18}+\frac{4418157975}{34359738368}x^{19}+O\left(x^{20}\right))$$

$> f3 := subs(x = 4 * x,f2);$
$> taylor(f3,x = 0,20);$

$$(1+2\,x+6\,x^2+20\,x^3+70\,x^4+252\,x^5+924\,x^6+3432\,x^7+$$
$$12870\,x^8+48620\,x^9+184756\,x^{10}+705432\,x^{11}+2704156\,x^{12}+10400600\,x^{13}+40116600\,x^{14}+$$
$$155117520\,x^{15}+601080390\,x^{16}+2333606220\,x^{17}+9075135300\,x^{18}+35345263800\,x^{19}+O\left(x^{20}\right))$$

$> cb := proc(n)\ binomial(2 * n,n)\ end;$

$$proc(n)\ binomial(2 * n,n)\ end$$

$> cb(17);$

$$2333606220$$

$> cb(19);$

$$35345263800$$

$> sum(binomial(2 * n,n) * x^n,n = 0..infinity);$

$$\frac{1}{\sqrt{1-4x}}$$

Solutions approchées de $(x = arctan(2x))$

$> read\ flot;$
$f := proc(n)$
$\ local x,y;$
$\ x := 1;$
$\ y := 2;$
$\ while\ evalf(y - x,n) <> 0\ do\ x := y; y := evalf(2 * arctan(x),n)\ od$
$\ end;$

$> f(10);$

$$2.331122370$$

$> f(20);$

$$2.3311223704144226136$$

$> f(100);$

$$2.331122370414422613667835955917121338269077695386114575109737293393230817432716673842154257104393014$$

Exercice 3.1 *On définit, pour tout complexe* α

$$\binom{\alpha}{k} := \frac{\alpha(\alpha-1)\cdots(\alpha+k-1)}{k!}$$

i) En utilisant le développement de Taylor, montrer que

$$(1+h)^\alpha = \sum_k^\infty \binom{\alpha}{k} h^k$$

ii) En déduire la formule

$$\sum_{n\geq 0} \binom{2n}{n} x^n = \frac{1}{\sqrt{1-4x}}$$

3.2.2 Programmation

Le produit de facteurs

$$\prod_{\epsilon_i\in\{-1,1\},\ i=1..3} (X + \epsilon_1\sqrt{2} + \epsilon_2\sqrt{3} + \epsilon_3\sqrt{5})$$

qui peut servir à montrer l'équation (1).

```
pol2 :=
proc()
local res,i,j,k;
res := 1;
for i from -1 by 2 to 1 do
for j from -1 by 2 to 1 do
for k from -1 by 2 to 1 do
res:= res*(X - i*2^(1/2) - j*3^(1/2) - k*5^(1/2))
od
od
od;
res
end;
```

$> pol2();$

$$\left(X+\sqrt{2}+\sqrt{3}+\sqrt{5}\right)\left(X+\sqrt{2}+\sqrt{3}-\sqrt{5}\right)\left(X+\sqrt{2}-\sqrt{3}+\sqrt{5}\right)\left(X+\sqrt{2}-\sqrt{3}-\sqrt{5}\right)$$
$$\left(X-\sqrt{2}+\sqrt{3}+\sqrt{5}\right)\left(X-\sqrt{2}+\sqrt{3}-\sqrt{5}\right)\left(X-\sqrt{2}-\sqrt{3}+\sqrt{5}\right)\left(X-\sqrt{2}-\sqrt{3}-\sqrt{5}\right)$$

$> expand(");$

$$-960\,X^2 + 352\,X^4 - 40\,X^6 + X^8 + 576$$

$> subs(seq(X^{(2*i)} = x^i, i = 1..4),");$

$$-960\,x + 352\,x^2 - 40\,x^3 + x^4 + 576$$

$> [solve(")];$

$$[10 + 2\sqrt{10} + 2\sqrt{3}\sqrt{7+2\sqrt{10}}, 10 + 2\sqrt{10} - 2\sqrt{3}\sqrt{7+2\sqrt{10}},$$
$$10 - 2\sqrt{10} + 2\sqrt{3}\sqrt{7-2\sqrt{10}}, 10 - 2\sqrt{10} - 2\sqrt{3}\sqrt{7-2\sqrt{10}}]$$

$> op(1,");$

$$10 + 2\sqrt{10} + 2\sqrt{3}\sqrt{7 + 2\sqrt{10}}$$

Équations complexes. Voici l'exemple de résolution des équations du second degré.

```
> read(csolve);
csolve :=
proc(P,z)
local a,b,c,delta,r;
a := coeff(P,z,2);
b := coeff(P,z,1);
c := coeff(P,z,0);
delta := expand(b^2 - 4*a*c);
if Im(delta) = 0 then
if 0 <= delta then r := sqrt(delta) else r :=
I*sqrt(-delta) fi
else
r := sqrt(1/2*Re(delta) + 1/2*sqrt(Re(delta)^2 + Im(delta)^2));
r := r + 1/2*I*Im(delta)/r
fi;
1/2*(-b - r)/a,1/2*(-b + r)/a
end;

> P := z^2 + (-3 - 2*I)*z + 5 + I
> solve(P,z);
```

$$3/2 + \sqrt{-1} + \frac{\sqrt{-15 + 8\sqrt{-1}}}{2}, 3/2 + \sqrt{-1} - \frac{\sqrt{-15 + 8\sqrt{-1}}}{2}$$

```
> csolve(P,z);
```

$$1 - \sqrt{-1}, 2 + 3\sqrt{-1}$$

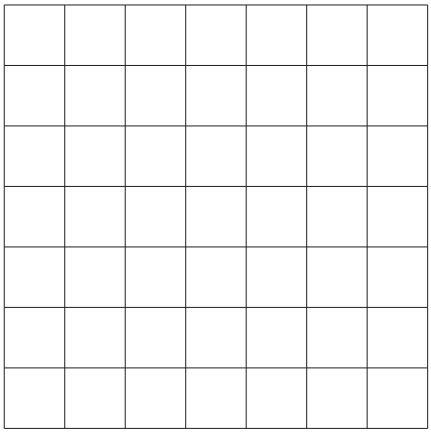
Ce "bug" - ou cette faiblesse - a d'ailleurs disparu de Maple V.4 release 3 qui donne la bonne réponse avec solve.

3.3 Quelques exercices et problèmes de combinatoire

Exercice 3.2 I) Bijections explicte $\mathbb{N} \to \mathbb{N}^2$. —
On dispose les entiers naturels sur le réseau $\mathbb{N}^2$ de la façon suivante

$$0 \to (0,0);\ 1 \to (1,0);\ 2 \to (0,1);\ 3 \to (0,2);\ 4 \to (1,1);\ 5 \to (2,0);\ 6 \to (3,0) \cdots$$

a) Disposer les premiers entiers dans les cases suivantes



pour vérification, le déb

pour vérification les premiers nombres se placent ainsi

3
2 4
0 1 5 6

la règle est

1. un premier pas horizontal
2. pas en diagonale
3. un pas montant dès que l'on touche l'axe des "y"
4. pas en diagonale vers l'axe des "x"
5. etc ...

a) Montrer que le point de coordonnées $(0,2n)$ porte le numéro $2n^2 + 3n$.
b) En déduire une méthode pour donner le numéro porté par le point (p,q).

Exercice 3.3 II Structures informatiques. —

Le but de ce problème est de manipuler les ordres qui permettent de numéroter et d'engendrer des structures informatiques de base.
A) "Balls and bars" (BB). —
On appelle ainsi des structures du type

$$\bullet \,|\, \bullet \bullet \bullet \,|\, \bullet \,|\, \bullet \bullet \tag{2}$$

c'est à dire des points équidistants et horizontaux séparés par des | de façon que :

Toutes les barres ont (au moins) un point qui les précède et un point qui les suit.

1) Montrer que de telles structures peuvent être engendrées par une grammaire que l'on précisera.
2) Se servir de ce qui précède pour montrer que la série génératrice $S = \sum_{n\geq 0} a_n z^n$ (a_n étant le nombre de ces structures avec n points) vérifie

$$S = 1 + z + 2z(S-1) \tag{3}$$

3) Résoudre (3) et montrer que $a_n = 2^{n-1}$ pour $n > 0$.
4) Un vecteur d'entiers (VE) est une liste d'entiers non nuls $\mathbf{I} = [i_1,i_2,\cdots i_k]$ et son poids est la somme de ses coordonnées. En voici quelques uns avec leur poids

I	[1,4]	[3,1,1]	[2,1,3]	[1,2,1,3]
$poids(\mathbf{I})$	*0*	*5*	*6*	*7*

a) Donner une correspondance entre les BB de poids n et les vecteurs d'entiers de poids n.
On attribue à chaque (VE) $\mathbf{I} = [i_1,i_2,\cdots i_k]$ de poids $n > 0$ un nombre binaire sur $n-1$ bits dont les 1 sont aux places $(i_1,i_1+i_2,\cdots,(\sum_{j=1^r} i_j)),\cdots,i_1,i_1+i_2\cdots i_{k-1})$ cette dernière somme étant $n - i_k$. Voici les VE de poids 4 et leurs numéros binaires

I	[4]	[3,1]	[2,2]	[2,1,1]	[1,3]	[1,2,1]	[1,1,2]	[1,1,1,1]
Numéro	*000*	*001*	*010*	*011*	*100*	*101*	*110*	*111*

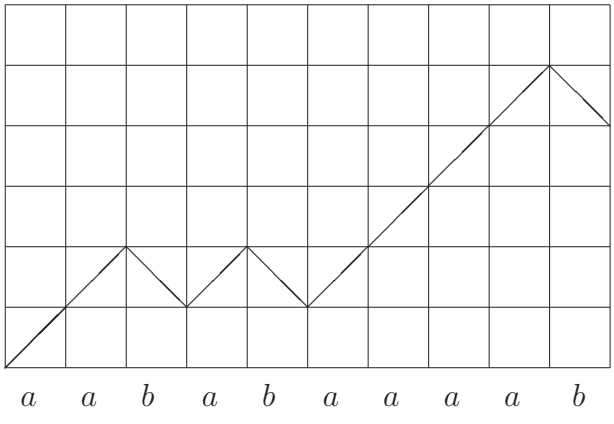
b) Donner les 10 premières compositions de poids 5 et leur numéro binaire, classées (comme dans l'exemple) par ordre croissant de ce numéro.
c) Si on désigne par $\prec_{lex}$ l'ordre lexicographique entre les vecteurs d'un poids donné et $<$, l'ordre entre les entiers binaires, établir une relation entre les deux ordres.
B) (Listes sans répétitions). —
Soit $E = a,b,c\cdots z$ l'alphabet usuel ordonné par l'ordre du dictionnaire et $k \leq 26$. On note SR_E^k l'ensemble des mots de longueur k (ou listes, c'est la même chose) sans répétitions (i.e. les objets sont $x_1x_2\cdots x_k$ avec $i < j \Longrightarrow x_i \neq x_j$). Pour $k = 3$, les premièrs mots, par ordre lexicographique, sont (les traits qui soulignent ne font pas partie de la structure, mais sont là pour aider à la solution des (1)b) et (2)).

$$abc,abd\cdots ab\underline{z},\cdots a\underline{zy}\cdots bac\cdots ba\underline{z},bca,bcd,\cdots ba\underline{z},bda,\cdots b\underline{zy}\cdots \tag{4}$$

1) ($k = 3$) a) Montrer que zyx est le dernier mot de la liste.
b) Pour les mots de longueur 3 différents de zyx, trouver la loi de formation du suivant.
2) Trouver le procédé général pour k quelconque.

Exercice 3.4 III Chemins de Dyck. —

Pour l'alphabet $\{a,b\}$, on donne la fonction de poids $\omega(a) = 1;\ \omega(b) = -1$. Un mot est représenté par un chemin (ligne brisée) avec un pas nord-est pour a et sud-est pour b selon le schéma ci-dessous.



Pour un mot w et $1 \leq i \leq j \leq |w|$, on note $w[i..j]$ le facteur $w[i]w[i+1]\cdots w[j]$ où $w[k]$ est la lettre de w qui est à la place k.
Les mots de Dyck positifs sont définis par

$$D_+ = \{w \in (a+b)^* \mid \omega(w) = 0 \text{ et } (\forall i \in [1..|w|-1])(\omega(w[1..i]) \geq 0)\} \tag{5}$$

a) Montrer que $D_+ = \mathcal{L}(G,X)$ est défini par l'une des règles suivantes
Règle 1
Un mot de Dyck est

- *soit ϵ*
- *soit la concaténation de deux mots de Dyck non vides*
- *soit de la forme awb où w est un mot de Dyck*

pour ceux qui connaissent les grammaires c'est résumé par

$$G_1 \ : \ X \to \epsilon + XX + aXb \ . \tag{6}$$

Règle 2
Un mot de Dyck est

- *soit ϵ*
- *soit de la forme aw_1bw_2 où w_1,w_2 sont des mots de Dyck*

pour ceux qui connaissent les grammaires c'est résumé par

$$G_2 \ : \ X \to \epsilon + aXbX \tag{7}$$

b) Expliquer sur un exemple pourquoi G_1 est ambigüe et montrer que G_2 ne l'est pas.
2) On note D^0_+, l'ensemble suivant

$$D^0_+ = \{w \in (a+b)^+ \mid \omega(w) = 0 \text{ et } (\forall i \in [1..|w|-1])(\omega(w[1..i]) > 0)\} \tag{8}$$

a) Montrer que $D^0_+ \subseteq D_+$.
b) Montrer que tout mot de D_+ se factorise $w = u_1u_2\cdots u_k \ ; \ u_i \in D^0_+$.
Cette factorisation est-elle unique Peut-on l'obtenir de façon automatique ? Si oui, comment ?
c) Montrer que $D^0_+ = \mathcal{L}(G,Y)$ pour la grammaire $G \ : \ X \to \epsilon + aXbX, Y \to aXb$.
3) On note D_- l'ensemble suivant (Dycks négatifs)

$$D_- = \{w \in (a+b)^* \mid \omega(w) = 0 \text{ et } (\forall i \in [1..|w|-1])(\omega(w[1..i]) \leq 0)\} \tag{9}$$

Donner une condition graphique qui permet de décider si un mot est dans D_-.
4) Soit

$$D^0_- = \{w \in (a+b)^+ \mid (\forall i \in [1..|w|-1])(\omega(w[1..i]) < 0)\} \tag{10}$$

a) Reprendre les questions 2) (a) et (b) en échangeant les signes.
Soit Φ, la substitution $(a+b)^* \to (a+b)^*$ définie par $\Phi(a) = b \ ; \ \Phi(b) = a$.
b) Montrer que $\Phi(D_+) = D_-$ et que $\Phi(D^0_+) = D^0_-$.
c) En déduire que tout mot de D_- se factorise $w = u_1u_2\cdots u_k \ ; \ u_i \in D^0_-$.
et que cette factorisation est unique. Peut-on l'obtenir de façon automatique ? Si oui, comment ?
d) Donner une grammaire qui permet d'engendrer D^0_-.
4) Soit $D = \{w \in (a+b)^* \mid \omega(w) = 0\}$.
a) Montrer que $D = \{w \in (a+b)^* \mid |w|_a = |w|_b\}$, donner les mots de longueur ≤ 4 et les

dessiner. Combien y a-t-il, dans D, de mots de longueur $2n$? $2n+1$?
b) Montrer que tout mot de D se factorise

$$w = u_1 u_2 \cdots u_k \ ; \ u_i \in D_+^0 \sqcup D_-^0 \tag{11}$$

c) Cette factorisation est-elle unique ?
5) Déduire de la factorisation (11) une grammaire pour engendrer D, puis de 4) (c) une grammaire non-ambigüe.

3.4 Systèmes de Calcul Formel

Voici, à gros traits, des éléments de l'histoire récente (nous ne parlons pas de la machine arithmétique de Pascal) des Systèmes de Calcul Formel.

1953 : Premier système de dérivation (en LISP).
1958 : LISP (John Mc Carthy au MIT).
1960-70 : SAC-1 (G. Collins) manipulations de polynômes en Fortran.
ALPAK aux Bell Labs (polynômes et fractions rationnelles).
FORMAC à IBM.
Premier programme d'intégration en LISP.
MATHLAB au MIT (utilisation interactive, affichage bidimentionnel).
REDUCE (en Lisp) à Standford.
1970-80 : MACSYMA (En lisp au MIT).
SCRATCHPAD (chez IBM).
1980-90 : MAPLE à Waterloo (en C).
SMP par S. Wolfram.
SCRATCHPAD II (chez IBM).
Systèmes dédiés MACAULAY, GAP, CAYLEY, PARI,...
1990-.. : MATHEMATICA (S. Wolfram), AXIOM (Version commercialisée de SCRATCHPAD I), MuPAD (Développé par des universitaires européens).

Quelques commentaires..

AXIOM, demande une grosse station de travail I.B.M.
MACSYMA, développé au MIT, commercialisé par Symbolics.
REDUCE de Anthony C. Hearn, disponible chez Softworld.
MATHEMATICA, très convivial mais pas toujours très fiable.
muMATH, peut fonctionner sur de petits ordinateurs.
SCRATCHPAD, développé par IBM, langage fortement typé.
CAYLEY, manipule les groupes.
MACAULAY, développé par David Bayer et Michel Stillman, résout bien les systèmes d'équations algébriques.
MAPLE, développé à l'université de Waterloo (Canada), disponible chez Softworld.
PARI Comporte les meilleurs algorithmes actuels de théorie des nombres (développé à Bordeaux).
SYMMETRICA Comporte les meilleurs algorithmes sur la combinatoire du groupe symétrique, les algèbres de Hecke et leurs représentations.
MuPAD Système généraliste qui a des commandes similaires a Maple, mais la communication avec C et C++ est plus simple.

3.5 Prise en main de Maple : T.D.

3.6 Séries à une variable (univariées)

3.6.1 Généralités

Les monômes forment un monoïde et les séries sont juste les fonctions sur ce monoïde.
Exemple des arbres binaires, grammaire et série caractéristique.
Formule de la série génératrice des Dycks (TD).
Calculs : exemple de $\sum_{n\geq 0} n^2 z^2$

3.6.2 Les monômes

On le verra dans la suite, on comprend les opérations entre les polynômes ou les séries quand on a défini la loi de multiplication des monômes. Ici, l'alphabet des variables a un seul élément noté usuellement $\Sigma = \{x\}$ ou $\{z\}$ ou $\{t\}$ ou tout autre élément qui n'est pas une constante. L'ensemble des monômes est une copie de $(\mathbb{N},+)$ en notation multiplicative. Soit, avec z comme variable $z^{n+m} = z^n z^m$ et $z^0 = 1$.

On considère ensuite les sommes $\sum_{n\geq 0} a_n z^n$ où les a_n sont pris dans un semi-anneau, c'est à dire, pour parler de façon imagée *un anneau sans le signe "moins"*.

Définition 3.5 *Un semi-anneau est la donnée d'un ensemble* $(\mathbb{K}, +, \times)$ *muni de deux lois internes telles que :*

1. $(\mathbb{K},+)$ *est un monoïde commutatif (d'élément neutre* $0_\mathbb{K}$*)*
2. $(\mathbb{K},\times)$ *est un monoïde (d'élément neutre* $1_\mathbb{K}$*)*
3. $\times$ *est distributive à droite et à gauche sur* $+$
4. $0_\mathbb{K}$ *est un élément absorbant* $0_\mathbb{K}.x = x.0_\mathbb{K} = 0_\mathbb{K}$

3.6.3 Les séries à une variable

Les opérations suivantes ont été détaillées en cours. Nous les rappelons ici :

Pour $S = \sum_{n\geq 0} f(n)z^n$ et $T = \sum_{n\geq 0} g(n)z^n$, on a

Opération	Remarques	Notation = Formule
Somme	(1)	$S + T = \sum_{n\geq 0}(f(n) + g(n))z^n$
Produits externes ($\lambda \in \mathbb{K}$)	(2)	$\lambda S = \sum_{n\geq 0}(\lambda f(n))z^n$ $S\lambda = \sum_{n\geq 0}(f(n)\lambda)z^n$
Produit (Cauchy)	(3)	$ST = \sum_{n\geq 0}(\sum_{p+q=n} f(p)g(q))z^n$
Décalage	(6)	$\gamma_z^*(S) = \frac{1}{z}(S - S(0))$
Dérivée	(7)	$S' = \sum_{n\geq 1} f(n)nz^{n-1}$
Intégrale	(8)	$\int_0 S(z)dz = \sum_{n\geq 0} f(n)\frac{z^{n+1}}{n+1}$

Remarque 3.6 *i) Les opérations (1) et (2) viennent de ce qu'une série est une fonction.*
ii) Le produit (3) est appellé convolution, concaténation ou produit de CAUCHY.

3.7 Séries liées à des statistiques

3.7.1 Un exemple : une stastistique bivariée productrice d'un codage

Propos de cette section. — Dans ce qui suit, nous allons considérer trois structuresde données : certaines permutations, les tableaux de Young standard à deux lignes égales et les mots de Dyck. Ils ont même statistique monovariée et donc sont en correspondance bijective. Cependant, c'est en raffinant encore celle-ci, en une stastistique bivariée qu'on peut apercevoir un code qui les représente bijectivement.

Permutations sans 123. —

Définition 3.7 *i) On appelera ici* permutation de longueur "n"*, une liste* $[a_1,a_2,\cdots a_n]$ *telle que tous les nombres de* $[1\cdots n]$ *soient pris une fois et une seule. Plus formellement*

$$\{a_1,a_2,\cdots a_n\} = [1\cdots n] \tag{12}$$

ii) L'ensemble des permutations de longueur n *est noté* $\mathfrak{S}_n$.

Remarque 3.8 *Les listes de* $\mathfrak{S}_n$ *codent les bijections* $[1..n] \mapsto [1..n]$ *et donc* $\mathfrak{S}_n$ *est muni d'une structure de groupe pour la composition : c'est le groupe symétrique d'ordre* n.

Il y a $n!$ permutations de longueur n. Voici, par exemple, la liste des permutations de longueur 4.

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

Dans la suite, les permutations seront notées comme des mots. Par exemple $[4,2,3,1]$ sera notée 4231 Pour un mot $w \in A^*$ de longueur n et toute partie $I = \{i_1,i_2,\cdots i_k\} \subset \{1,2,\cdots n\}$, on notera $w[I] = w[i_1]w[i_2]\cdots w[i_k]$. Pour une permutation notée de longueur n comme un mot, il y a 2^n sous-mots (on dit aussi, dans ce cas, sous-suite).

Définition 3.9 *i) Pour une permutation $w \in \mathfrak{S}_n$, le plus grand k tel qu'il existe $I \subset \{1,2,\cdots n\}$ tel que $w[I]$ soit croissante est le* premier invariant de Greene *et est noté $g_1(w)$.*
ii) On notera P_n l'ensemble des permutations w de longueur n telles que $g_1(w) \leq 2$

On peut montrer que

$$\#P_n = c_n = \frac{\binom{2n}{n}}{n+1}$$

le $n^{\text{ième}}$ nombre de Catalan.

Tableaux de Young standard. —

Définition 3.10 *i) On appelle* partition *de n et on note $I = [n_1,n_2,\cdots n_k]$, une vecteur d'entiers non nuls décroissant. Ces entiers représentent le domaine du plan discret $D \subset \mathbb{N} \times \mathbb{N}$ défini par*

$$D = \{(x,y) \mid 0 \leq y \leq k-1, 0 \leq x \leq n_k - 1\} \tag{13}$$

le poids de I, noté $|I|$ est la somme $\sum_{j=1}^{k} n_j$.
ii) Un tableau de Young standard est une application bijective $D \mapsto [1..|I|]$ croissante pour l'ordre produit

8			
5	6	7	
1	2	3	4

7			
3	6	8	
1	2	4	5

5			
3	4	8	
1	2	6	7

TAB. 1 – *Tableaux de Young standard de forme 431*

Ici, on notera T_n l'ensemble des tableaux de Young standard de forme $I = [n,n]$. Pour exemple voici T_3.

4	5	6
1	2	3

3	5	6
1	2	4

2	5	6
1	3	4

3	4	6
1	2	5

2	4	6
1	3	5

TAB. 2 – *Tableaux de Young de T_3*

On montre que $\#T_n = c_n$.

3.7.2 Mots de Dyck & Arbres binaires complets

Exercice 3.1 *On considère l'alphabet $A = \{a,b\}$ et le morphisme $\pi : A^* \mapsto \mathbb{Z}$ définit par $\pi(a) = 1$, $\pi(b) = -1$. Les mots de Dyck sont les mots*

$$GD_+ := \{w \in A^* \mid (\forall u,v \in A^*)(uv = w \implies \pi(u) \geq 0) \text{ et } \pi(w) = 0\} \tag{14}$$

On note D_n, l'ensemble des mots de Dyck de longueur $2n$. On note c_n leur nombre $\#D_n = c_n$.
Q1) Montrer que les mots de Dyck sont définis par la grammaire suivante

$$G_2 \ : \ X \to \epsilon + aXbX \tag{15}$$

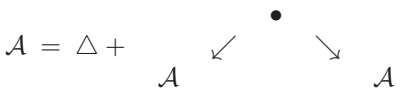
et que, si $w \neq \epsilon$ est un mot de Dyck, la décomposition $w = aubv$ est unique.
Q2) En déduire que la suite c_n est donnée par l'équation suivante sur sa série $S = \sum_{n\geq 0} c_n\, z^n$

$$S = 1 + zS^2 \tag{16}$$

Q3) Montrer que

$$S := \frac{1 - \sqrt{1-4z}}{2z} = \frac{2}{1 + \sqrt{1-4z}} \tag{17}$$

Exercice 3.2 *L'ensemble $\mathcal{A}$ des arbres binaires est construit par la grammaire* **(G1)**



Il y a trois notions de taille classiques : le nombre de nœuds, le nombre de sommets, la profondeur.
1) Montrer que pour toutes ces notions de taille, le nombre d'arbres d'une taille donnée est fini.
2) Donner la série génératrice des arbres binaires complets par nombre de nœuds. (Indication, c'est la série S de l'exercice précédent).
3) Donner la grammaire qui engendre les arbres binaires dont les feuilles sont indexées par un ensemble F fini donné. Donner les premiers éléments pour $F = \{a,b\}$.

Éléments pour la solution (s'entraîner à rédiger l'exercice précédent). —
On peut adopter une notation typographiquement plus rapide : un arbre différent de $\bullet$ sera donc noté $\mathcal{A} = (\mathcal{A}_g,\mathcal{A}_d)$ où $\mathcal{A}_g$ (resp. $\mathcal{A}_d$) est le sous-arbre gauche (resp. droit). La taille (ici nombre de nœuds) peut se définir récursivement par

$$\tau(\bullet) = 1;\ \tau((\mathcal{A}_g,\mathcal{A}_d)) = \tau(\mathcal{A}_g) + \tau(\mathcal{A}_d) + 1 \tag{18}$$

ce qui donne l'équation pour la SGO $\sum_{k\geq 0} a_k x^k$ (où a_k est le nombre d'arbres binaires à k nœuds).

$$T = x + xT^2 \tag{19}$$

soit $xT^2 - T + x = 0$ on résout (19) par la méthode habituelle. Le discriminant est $\Delta = 1 - 4x^2$ et les racines possibles sont

$$T_1 = \frac{1 - \sqrt{1-4x^2}}{2x};\ T_2 = \frac{1 + \sqrt{1-4x^2}}{2x} \tag{20}$$

T_2 ayant un terme de plus bas degré en $1/x$ ne peut pas être retenue. On a donc $T = T_1$. Vérifions ce résultat.
On sait que

$$(1+X)^\alpha = \sum_{k\geq 0} \binom{\alpha}{k} X^k \tag{21}$$

avec

$$\binom{\alpha}{k} := \frac{(\alpha)(\alpha-1)\cdots(\alpha-k+1)}{k!} \tag{22}$$

ceci qui donne

$$\sqrt{1-4x^2} = \sum_{k\geq 0} \binom{1/2}{k} (-4x^2)^k \tag{23}$$

à l'aide de (22), on a, pour $k \geq 1$

$$\binom{1/2}{k} = \frac{(1/2)(1/2-1)(1/2-2)\cdots(1/2-k+1)}{k!} = \frac{(-1).(-3)\cdots(3-2k)}{2^k k!} = \tag{24}$$

$$\frac{(-1)^{k-1}1.3.\cdots(2k-3)}{2^k k!} = \frac{(-1)^{k-1}(2k-2)!}{2^{2k-1}k!(k-1)!} \tag{25}$$

en remplaçant ce résultat dans (23) puis dans (20), il vient

$$T = \sum_{k\geq 1} \frac{\binom{2k-2}{k-1}}{k} x^{2k-1} = \tag{26}$$

$$x + x^3 + 2x^5 + 5x^7 + 14x^9 + 42x^{11} + 132x^{13} + 429x^{15}\cdots \tag{27}$$

Exercice 3.11 *Reprendre l'exercice précédent avec le nombre de sommets comme taille.*

3.8 Produit scalaire $\langle$série$|$polynôme$\rangle$ et premières opérations

Pour suivre l'usage en ce qui concerne les séries citées et quelques autres (polynômes d'exponentielles) nous appelerons *polynôme* une série $M \mapsto k$ à support fini. L'espace des séries sera noté k^M et celui des polynômes $k^{(M)}$.

Définition 3.12 *La dualité entre k^M et $k^{(M)}$ est définie, pour $S \in k^M$ et $P \in k^{(M)}$ par*

$$\langle S|P\rangle = \sum_{m\in M} S(m)P(m) \tag{28}$$

on vérifie facilement que cette somme est à support fini (donc bien définie).

Exercice 3.13 *1) a) Rappeler la structure d'EV de $k^X = \mathcal{F}(X,k)$. Montrer que $k^{(X)} = \{f \in k^X \mid supp(f)$ est fini$\}$ est un SEV de k^X.*
*b) Pour toute partie $Z \subset X$, χ_Z est la fonction caractéristique de Z (avec les notations de l'informatique $\chi_Z(x) = [x \in Z]$ où $[\,]$ est le symbole d'Iverson [**?**]). Montrer que, si Z_1 et Z_2 sont disjointes, on a $\chi_{Z_1} + \chi_{Z_2} = \chi_{Z_1\cup Z_2}$.*
c) Que se passe-t-il si les parties ne sont pas disjointes ?, si $k = \mathbb{Z}_2$?
d) Montrer que pour $S \in k^M$; $m \in M$, on a $\langle S|\chi_{\{m\}}\rangle = S(m)$
2) a) Montrer que pour tout $P \in k^{(M)}$, on a

$$P = \sum_{m\in M} \langle P|\chi_{\{m\}}\rangle \chi_{\{m\}} \tag{29}$$

b) Montrer que l'application $M \mapsto k^M$ définie par $m \mapsto \chi_{\{m\}}$ a son image dans $k^{(M)}$ et qu'elle est injective. On identifiera alors m à χ_m.
c) Comment se réécrit l'équation (29) avec l'identification du (b) ?
3) On veut prolonger aux séries l'écriture (29) ou plutôt sa version simplifiée avec l'identification du 2) b). On dira qu'une famille $(S_i)_{i\in I}$ de séries est sommable *si, pour tout $m \in M$, la fonction $I \mapsto k$ define par $i \mapsto S_i(m)$ est à support fini. Dans ce cas, on dira que la famille $(S_i)_{i\in I}$ est de somme S définie par $S(m) = \sum_{i\in I} S_i(m)$. On notera $S = \sum_{i\in I} S_i$*
Soit $S \in k^M$ une série. Montrer que la famille $(\langle S|m\rangle m)_{m\in M}$ est sommable et que $S = \sum_{m\in M}\langle S|m\rangle m$.

4 Séries d'une variable ($\mathbb{C}[[z]]$)

Faute de temps, nous n'aborderons pas les opérations sur les séries en général, mais nous concentrerons sur celles à une variable de façon que l'étudiant sache calculer dans $\mathbb{C}[[z]]$.

4.1 Opérations sur les séries

Tableau des opérations usuelles (décalage, intégration, dérivation).

Nom	Opérateur	Effet sur $S = \sum_{n=0}^{\infty} a_n z^n$
Décalage	γ_z^*	$\sum_{n=0}^{\infty} a_{n+1} z^n$
Dérivation	$\frac{d}{dz}$	$\sum_{n=1}^{\infty} a_n n z^{n-1}$
Intégration	$\int_0$	$\sum_{n=0}^{\infty} a_n \frac{z^{n+1}}{n+1}$
Sommes cumulées	$\gamma_{\frac{1}{1-z}}$	$\sum_{n=0}^{\infty} \left(\sum_{j=0}^{n} a_j \right) z^n$
Étoile ($a_0 = 0$)		$\frac{1}{1-S} = \sum_{n=0}^{\infty} S^n$

Exercice 4.1 *Fontaines de pièces (Wilf)*

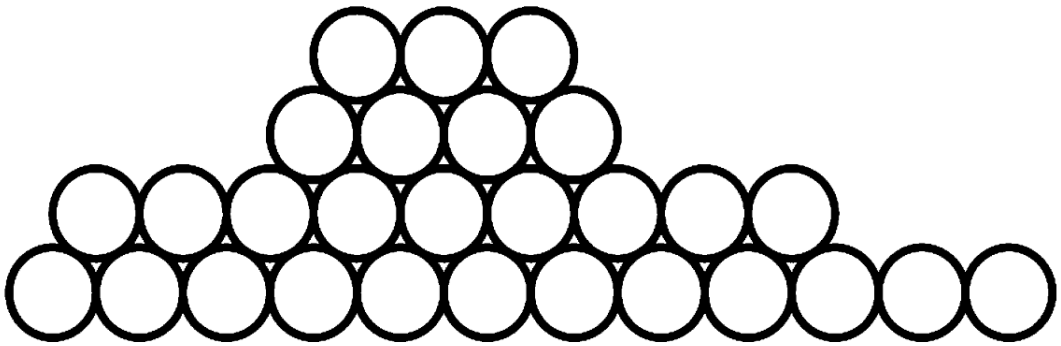


Fig 1. — *Une fontaine de pièces.*

On définit les fontaines de pièces comme ci-dessus. Donner la série génératrice de ces objets par nombre de pièces "à la base".

Exercice 4.2 *Prendre ces opérateurs deux par deux et donner des formules pour la composition et l'échange (si elles existent). Par exemple prouver la formule* $\gamma_z^* \frac{d}{dz} - \frac{d}{dz}\gamma_z^* = (\gamma_z^*)^2$.

4.2 Les deux produits : Convolution (produit de Cauchy) et produit de Hadamard

4.2.1 Produit de Cauchy

Ce produit s'obtient facilement à partir de la notation sommatoire. On multiplie formellement les sommes. Avec $S = \sum_{n=0}^{\infty} \langle S|z^n\rangle z^n$; $T = \sum_{m=0}^{\infty} \langle T|z^m\rangle z^m$ on a

$$ST = \sum_{n,m=0}^{\infty} \langle S|z^n\rangle \langle T|z^m\rangle z^{n+m} \tag{30}$$

puis, en regroupant par degrés,

$$ST = \sum_{r=0}^{\infty} \Big(\sum_{n+m=r} \langle S|z^n\rangle \langle T|z^m\rangle \Big) z^r \tag{31}$$

Définition 4.3 *Soient* S,$T \in \mathbb{C}[[z]]$. *On définit le produit (de Cauchy) ou convolution de ces deux séries par la formule (31).*

$$ST = \sum_{r=0}^{\infty} \Big(\sum_{n+m=r} \langle S|z^n\rangle \langle T|z^m\rangle \Big) z^r \tag{32}$$

Exercice 4.4 SUPPORT D'UN PRODUIT. —
On rappelle que le support d'une série $R = \sum_{n\in\mathbb{N}} \langle R|z^n\rangle z^n$ est l'ensemble des $n \in \mathbb{N}$ pour lesquels $\langle R|z^n\rangle \neq 0$. Donner les supports des séries suivantes

$$a)\ \frac{1}{1-z^2} \quad b)\ \frac{z^k}{1+\beta z} \quad c)\ \frac{1}{1+z+z^2}$$

$$d)\ \frac{1}{1-z^3} \quad e)\ \frac{1-z}{1-z^2} \quad f)\ \frac{1}{1-z} + \frac{1}{1+z}$$

Exercice 4.5 ÉTOILE D'UNE SÉRIE. —
La notion de sommabilité définie dans l'exercice (3.13) est reprise ici dans le cadre plus simple des séries complexes à une variable.
On dira qu'une famille $(S_i)_{i\in I}$ de séries est sommable *si, pour tout $n \in \mathbb{N}$, la fonction $I \mapsto \mathbb{C}$ define par $i \mapsto \langle S_i|z^n\rangle$ est à support fini. Dans ce cas, on dira que la famille $(S_i)_{i\in I}$ est de somme S définie par $\langle S|z^n\rangle = \sum_{i\in I}\langle S_i|z^n\rangle$. On notera $S = \sum_{i\in I} S_i$*

1) Soit $T \in \mathbb{C}[[z]]$ et S_i sommable. Montrer que TS_i est sommable et que $\sum_{i\in I} TS_i = T\sum_{i\in I} S_i$. 2) a) Soit $S \in \mathbb{C}[[z]]$ une série sans terme constant (formellement $\langle S|z^0\rangle = 0$), montrer que la famille $(S^k)_{k\in\mathbb{N}}$ est sommable. b) Montrer que $\sum_{k\in\mathbb{N}} S^k = (1-S)^{-1}$ dans $\mathbb{C}[[z]]$.
3) En posant $S = a_0 + S^+$ où $a_0 = \langle S|z^0\rangle$ et $S^+ = S - a_0 = \sum_{n>0}\langle S|z^n\rangle z^n$, montrer la proposition (4.6).

Proposition 4.6 SÉRIES INVERSIBLES. — *Soit $S \in \mathbb{C}[[z]]$, S est inversible dans $\mathbb{C}[[z]]$ ssi son terme constant $\langle S|z^0\rangle$ l'est.*

Exercice 4.7 *1) Fontaine de pièces de Wilf (donné en cours).*
2) Polyominos tas stricts (donné dans la présentation du cours).

4.2.2 Séries rationnelles

Définition 4.8 *Une série $S \in \mathbb{C}[[z]]$ est dite rationnelle si elle peut s'exprimer sous la forme $S = \frac{P}{Q};\ P,Q \in C[z];\ Q(0) \neq 0$.*

Nous allons voir trois caractérisations (très importantes) des séries rationnelles :

- **Rat**. — Fraction rationnelle $S = \frac{P}{Q};\ Q(0) \neq 0$
- **Coeff**. — Coefficients $\langle S|z^n\rangle = \sum_{(s,\lambda)\in F} \alpha(s,\lambda)n^s\lambda^n$, avec F fini.
- **Rec**. — Récurence linéaire

$$(\exists(\alpha_0,\alpha_1,\cdots\alpha_k) \in \mathbb{C}^{k+1})(\forall n \in \mathbb{N})(\langle S|z^{n+k}\rangle = \sum_{j=0}^{k-1} \alpha_j\langle S|z^{n+j}\rangle)$$

les étudiants devront être maîtres du passage de l'une des ces formes à l'autre. Nous allons les détailler maintenant.
Rat→**Rec**. — Comme $Q(0) \neq 0$, on met la fraction $\frac{P}{Q}$ sous la forme

$$\frac{N(z)}{1 - \sum_{j=1}^{m} \beta_j z^j} \tag{33}$$

(en divisant haut et bas par $Q(0)$). Une fois ceci fait, on a

$$S(1 - \sum_{j=1}^{m} \beta_j z^j) = S - S(\sum_{j=1}^{m} \beta_j z^j) = N(z) \tag{34}$$

soit

$$S = S(\sum_{j=1}^{m} \beta_j z^j) + N(z) \tag{35}$$

Maintenant, pour $k \geq sup(\deg(N(z)),m)$, on a $\langle N|z^{n+k}\rangle = 0$ d'où

$$\langle S|z^{n+k}\rangle = \sum_{j=1}^{n} \beta_j\langle z^j S|z^{n+k}\rangle = \sum_{j=1}^{n} \beta_j\langle S|z^{n+k-j}\rangle \tag{36}$$

qui est la récurrence linéaire cherchée.

Exercice 4.9 *Les nombres de Fibonacci sont donnés par la récurrence*

$$F_0 = 0,\, F_1 = 1;\; F_{n+2} = F_n + F_{n+1} \tag{37}$$

a) Redémontrer que la série génératrice des nombres de Fibonacci est $S(z) = \frac{z}{1-z-z^2}$.
b) En déduire que celle des nombres de Fibonacci impairs $T(z) = \sum_{n\geq 0} F_{2n+1} z^n$ *est* $\frac{1}{1-3z+z^2}$
c) En déduire la relation de récurrence satisfaite par ces nombres (les $a_n = F_{2n+1}$*).*

Rec→Rat. — Soit S qui vérifie la relation de récurrence linéaire, donnée par des coefficients $(\alpha_0,\alpha_1,\cdots\alpha_k) \in \mathbb{C}^k$,

$$(\forall n \in \mathbb{N})(\langle S|z^{n+k}\rangle = \sum_{j=0}^{k-1} \alpha_j \langle S|z^{n+j}\rangle) \tag{38}$$

on a , pour tout n,

$$\langle (\gamma_z^*)^k S|z^n\rangle = \sum_{j=0}^{k-1} \alpha_j (\gamma_z^*)^j \langle S|z^n\rangle \tag{39}$$

soit

$$(\gamma_z^*)^k S - \sum_{j=0}^{k-1} \alpha_j (\gamma_z^*)^j S = 0 \tag{40}$$

si on fait $z^m(\gamma_z^*)^m S$ on obtient la série privée de ses m premiers termes, soit le reste d'ordre m $S - \mathrm{trunc}(S,m-1)$ (où $\mathrm{trunc}(S,l) = \sum_{n=0}^{l} \langle S|z^n\rangle z^n$ est l'opérateur de troncature). En multipliant l'équation précédente par z^k on obtient

$$0 = z^k(\gamma_z^*)^k S - \sum_{j=0}^{k-1} \alpha_j z^{k-j} z^j (\gamma_z^*)^j S = S - \mathrm{trunc}(S,k-1) - \sum_{j=0}^{k-1} \alpha_j z^{k-j}(S - \mathrm{trunc}(S,j-1)) \tag{41}$$

soit

$$S\Big(1 - \sum_{j=0}^{k-1} \alpha_j z^{k-j}\Big) = \mathrm{trunc}(S,k-1) - \sum_{j=0}^{k-1} \alpha_j z^{k-j} \mathrm{trunc}(S,j-1) \tag{42}$$

dont le second membre est un polynôme. Ce qui est la forme cherchée

$$S = \frac{\mathrm{trunc}(S,k-1) - \sum_{j=0}^{k-1} \alpha_j z^{k-j} \mathrm{trunc}(S,j-1)}{1 - \sum_{j=0}^{k-1} \alpha_j z^{k-j}} \tag{43}$$

Exercice 4.10 *En calculant* $F_{n+3}^2 - F_{n+2}^2$ *trouver une relation de récurrence entre les* $a_n = F_n^2$*. Calculer la fraction rationnelle* $\sum_n F_n^2 z^n$

Rat→Coeff. — On utilise la décomposition en éléments simples

$$\frac{P}{Q} = E(x) + \sum_{\lambda \in \mathcal{O}_Q} \sum_{j=1}^{n_\lambda} \frac{\alpha(\lambda,j)}{(z-\lambda)^j} \tag{44}$$

il suffit donc de savoir calculer les coefficients du développement de chaque $\frac{1}{(z-\lambda)^j}$ avec $\lambda \neq 0$.

Exercice 4.11 *1) a) Montrer que* $\frac{d^k}{dz^k}(1-\beta z)^{-1} = k!\beta^k(1-\beta z)^{-(k+1)}$.
b) Montrer que $\frac{d^k}{dz^k}(1-\beta z)^{-1} = \sum_{n=k}^{\infty} n(n-1)\cdots(n-k+1)\beta^n z^{n-k}$.
On adopte les notations commodes suivantes
`http://mathworld.wolfram.com/FallingFactorial.html`
Factorielle descendante $(x)_k = x(x-1)\cdots(x-k+1)$
Factorielle ascendante $x^{(k)} = x(x+1)\cdots(x+k-1)$

Binomial généralisé $\binom{x}{k} = \frac{1}{k!}(x)_k = \frac{x(x-1)\cdots(x-k+1)}{k!}$

2) Déduire du (1.b) que

$$\frac{1}{(1-\beta z)^{k+1}} = \sum_{n=k}^{\infty} \binom{n}{k} \beta^{n-k} z^{n-k} = \sum_{m=0}^{\infty} \frac{m^{(k)}}{k!} \beta^m z^m \tag{45}$$

Exercice Machine 4.12 *On veut convertir les factorielles montantes sur la base des puissances. La clef est le triangle des nombres de Stirling de 1ère espèce.*
1) a) Lire et interpréter les résultats du programme suivant

```
> with(combinat);
```

[Chi, *bell*, binomial, *cartprod*, *character*, *choose*, *composition*, *conjpart*, *decodepart*, *encodepart*, *fibonacci*, *firstpart*, *graycode*, *inttovec*, *lastpart*, *multinomial*, *nextpart*, *numbcomb*, *numbcomp*, *numbpart*, *numbperm*, *partition*, *permute*, *powerset*, *prevpart*, *randcomb*, *randpart*, *randperm*, *setpartition*, *stirling1*, *stirling2*, *subsets*, *vectoint*]

```
> rf:=proc(x,k) product(x+j,j=0..k-1) end;
```

$$rf := \textbf{proc}(x,k)\,\mathrm{product}(x+j, j=0..k-1)\ \textbf{end proc}$$

```
> rf(x,5);
```

$$x\,(x+1)\,(x+2)\,(x+3)\,(x+4)$$

```
> seq(print(expand(rf(x,k))),k=0..7);
```

$$1$$
$$x$$
$$x^2+x$$
$$x^3+3\,x^2+2\,x$$
$$x^4+6\,x^3+11\,x^2+6\,x$$
$$x^5+10\,x^4+35\,x^3+50\,x^2+24\,x$$
$$x^6+15\,x^5+85\,x^4+225\,x^3+274\,x^2+120\,x$$
$$x^7+21\,x^6+175\,x^5+735\,x^4+1624\,x^3+1764\,x^2+720\,x$$

```
> seq(print(seq(stirling1(n,k),k=0..n)),n=0..7);
```

$$1$$
$$0, 1$$
$$0, -1, 1$$
$$0, 2, -3, 1$$
$$0, -6, 11, -6, 1$$
$$0, 24, -50, 35, -10, 1$$
$$0, -120, 274, -225, 85, -15, 1$$
$$0, 720, -1764, 1624, -735, 175, -21, 1$$

b) Se renseigner sur les nombres de Stirling de première espèce (par exemple dans les livres ou dans `http://mathworld.wolfram.com/StirlingNumberoftheFirstKind.html`*) et donner l'expression de* $x^{(k)}$ *en fonction des puissances de* x.
2) On appelle Série Quasi-Polynôme (SQP), une série $P(l,\alpha,z) = \sum_{n=0}^{\infty} n^l \alpha^n z^n$ *avec* $\alpha \neq 0$.
a) Calculer le produit $P(l_1,\alpha_1,z)P(l_2,\alpha_2,z)$.
b) Exprimer la SGO des nombres de Fibonacci en fonction de deux SQP.
c) Déduire de la question (1) une expression des éléments simples $\frac{1}{(1-\alpha z)^{k+1}}$ *en fonction des SQP (*$P(l,\alpha,z) = \sum_{n=0}^{\infty} n^l \alpha^n z^n$*).*

Note 4.13 *On peut montrer que les fonctions $P(l,\alpha,z)$ sont linéairement indépendantes et forment une base des fractions rationnelles sans pôle nul ni partie entière. Ce qui précède montre qu'elles forment une* base multiplicative *de cet espace.*

Coeff$\rightarrow$**Rat**. —

Exercice 4.14 *On suppose qu'à partir d'un certain rang les coefficients de la série S sont une combinaison linéaire d'expressions du type $n^l\alpha^n$.*
1) Montrer que S se décompose de façon unique comme

$$S = P(z) + \sum_{(l,\alpha)\in F} \beta(l,\alpha)P(l,\alpha,z) \tag{46}$$

où P est un polynôme.

Exercice Machine 4.15 *1) a) Lire et interpréter le programme suivant.*

```
> S1:=matrix(10,10,(i,j)->stirling1(i-1,j-1));
```

$$S1 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & -3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -6 & 11 & -6 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 24 & -50 & 35 & -10 & 1 & 0 & 0 & 0 & 0 \\ 0 & -120 & 274 & -225 & 85 & -15 & 1 & 0 & 0 & 0 \\ 0 & 720 & -1764 & 1624 & -735 & 175 & -21 & 1 & 0 & 0 \\ 0 & -5040 & 13068 & -13132 & 6769 & -1960 & 322 & -28 & 1 & 0 \\ 0 & 40320 & -109584 & 118124 & -67284 & 22449 & -4536 & 546 & -36 & 1 \end{bmatrix}$$

```
> S2:=matrix(10,10,(i,j)->stirling2(i-1,j-1));
```

$$S2 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 7 & 6 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 15 & 25 & 10 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 31 & 90 & 65 & 15 & 1 & 0 & 0 & 0 \\ 0 & 1 & 63 & 301 & 350 & 140 & 21 & 1 & 0 & 0 \\ 0 & 1 & 127 & 966 & 1701 & 1050 & 266 & 28 & 1 & 0 \\ 0 & 1 & 255 & 3025 & 7770 & 6951 & 2646 & 462 & 36 & 1 \end{bmatrix}$$

```
> multiply(S1,S2);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

b) Les nombres de Stirling de deuxième espèce $stirling2(n,k)$ avec $n,k \in \mathbb{N}$ forment la matrice inverse de celle des nombres de première espèce. On peut voir cela come une relation entre patrice infinies ou bien comme une infinité de relations au sens suivant.
Soient les matrices de $\mathbb{Z}^{(N+1)\times(N+1)}$,

$$S1(N) = \big(stirling1(n,k)\big)_{0\le n,k\le N} \text{ et } S2(N) = \big(stirling2(n,k)\big)_{0\le n,k\le N} \tag{47}$$

alors $S1S2 = I_{(N+1)\times(N+1)}$.
2) Soit $\underline{P}(l,\alpha,z) = \sum_{n=l}^{\infty}(n)_l\ \alpha^n z^n$.
a) Exprimer les $\underline{P}(l,\alpha,z)$ *en fonction des* $P(l',\alpha',z)$. *Écrire soigneusement la matrice de passage.*
b) À l'aide des nombres de Stirling de deuxième espèce, exprimer les $P(l,\alpha,z)$ *en fonction des* $\underline{P}(l',\alpha',z)$.
3) Déduire de tout ce qui précède une méthode pour exprimer S *comme une fraction rationnelle.*

Remarque 4.16 *Les passages* **Rec**→**Coeff** *et* **Coeff**→**Rec** *se font par composition des méthodes précédentes. Si l'on tient à des méthodes directes (mais pas nécessairement plus courtes ni plus élégantes), on peut aussi, pour le premier, utiliser une réduction de Jordan de la matrice compagnon associée à la récurrence et utiliser des produits de Hadamard (cf paragraphe (4.2.4)) pour le second.*

4.2.3 Séries rationnelles (repésentations linéaires et aspect automatique)

Avant de généraliser la théorie des séries rationnelles à plusieurs variables (non-commutatives), il est utile de voir comment elles peuvent se représenter par un automate (unaire et à mltiplicités). On a la proposition suivante (énoncée dans le cas général où l'ensemble des scalaires K est un corps commutatif quelconque)

Proposition 4.17 *Soit* $S\sum_{n\in N} a_n z^n \in K\langle\langle z\rangle\rangle$ *une série. Les conditions suivantes sont équivalentes*
i) S *est rationnelle, c'est à dire* $S = P(Q)^{-1}$ *où* P,$Q \in K\langle z\rangle$ *et* $Q(0) \neq 0$
ii) Les coefficients de S *vérifient une récurrence linéaire*

$$(\exists(\alpha_j)_{0\leq j<k} \in K^k)(\forall n \in \mathbb{N})(a_{n+k} = \sum_{j=0}^{k-1} \alpha_j a_{n+j}) \tag{48}$$

iii) Il existe $\lambda \in K^{1\times n}$, $T \in K^{n\times n}$,$\gamma \in K^{n\times 1}$ *tels que*

$$(\forall n \in \mathbb{N})(a_n = \lambda T^n \gamma) \tag{49}$$

Preuve — ii) $\Longrightarrow$ iii). —
La relation de récurrence linéaire implique

$$(a_{n+1},a_{n+2}\cdots,a_{n+k}) = (a_n,a_{n+1}\cdots,a_{n+k-1}) \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 & \alpha_0 \\ 1 & 0 & \ddots & & \vdots & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & \vdots \\ \vdots & \vdots & & \ddots & 0 & \alpha_{k-2} \\ 0 & 0 & \cdots & \cdots & 1 & \alpha_{k-1} \end{pmatrix} \tag{50}$$

soit, en posant T, la matrice et $v_n = (a_n,a_{n+1}\cdots,a_{n+k-1})$, $v_{n+1} = v_n T$, d'où $v_n = v_0 T^n$. On a finalement

$$a_n = v_n \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = (a_0,a_1\cdots,a_{k-1})T^n \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{51}$$

iii) $\Longrightarrow$ i). —
En effet, $S = \sum_n a_n z^n = \sum_n \lambda T^n \gamma z^n = \lambda(\sum_n T^n z^n)\gamma = \lambda(I - zT)^{-1}\gamma$. Mais

$$(I - zT)^{-1} = \frac{1}{det(I - zT)}\text{comatrice}(I - zT) \tag{52}$$

et les $Q(z) = det(I - zT)$ est un polynôme en z tel que $Q(0) = 1$ et comatrice(I-zT) est une matrice de polynômes en z (exercice !!). D'où le résultat.
i) $\Longrightarrow$ ii). —
On peut poser $Q(z) = \sum_{j=0}^{k-1} \beta_j z^j$ avec $\beta_0 = 1$. Comme $P = SQ$, pour tout $n \in \mathbb{N}$, on a $\langle P|z^n\rangle = \sum_{p+q=n} \beta_p a_q$,

soit, si $n > N = max(\deg(P),k)$, $\sum_{p+q=n} \beta_p a_q = 0$ ce qui peut encore s'écrire $a_n = -\sum_{j=1}^{k} \beta_j a_{n-j}$, ce qui donne, pour tout $n \in \mathbb{N}$,

$$a_{n+N} = -\sum_{j=1}^{k} \beta_j a_{n+N-j} \tag{53}$$

ce qui entraine (ii). ♣

Définition 4.18 *Un triplet $\mathcal{T} = (\lambda,T,\gamma)$ tel que $\langle S|z^n\rangle = \lambda T^n \gamma$ est appellé* repésentation linéaire *de dimension n de S. De même S est appelée* comportement *de $\mathcal{T}$.*

Note 4.19 *Une série rationnelle admet en général plusieurs représentations linéaires. La dimension minimale de ces représentations est appellée* rang *de S. C'est aussi la dimension de l'espace vectoriel engendré par les décalées de S.*

4.2.4 Produit de Hadamard

C'est juste le produit des fonctions (fonctions "coefficient"), il sera noté $\odot$. Par exemple pour les séries d'une variable on a

$$\sum_{n=0}^{\infty} a_n z^n \odot \sum_{n=0}^{\infty} b_n z^n = \sum_{n=0}^{\infty} a_n b_n z^n \tag{54}$$

Exercice 4.20 *Effectuer les produits de Hadamard suivants*

a) $\frac{1}{1-z^2} \odot \frac{1}{1-2z}$ *b)* $\frac{1}{1+z+z^2} \odot \frac{1}{1-2z}$ *c)* $\frac{1}{1+z+z^2} \odot f(z)$

d) $\frac{1}{1-z^2} \odot e^z$ *e)* $\frac{z}{1-z^2} \odot e^z$ *f)* $\frac{1}{1-z} \odot f(z)$

Montrer que le résultat de (c) est rationnel si $f(z)$ est rationnelle i.e. si $f(z) = \frac{P(z)}{Q(z)}$; $Q(0) \neq 0$ **indication**. — *Pour le (b), décomposer en éléments simples. Pour le (c) on pourra remarquer que $\frac{1}{1+z+z^2} = \frac{1-z}{1-z^3}$.*

Théorème 4.21 PRODUIT DE HADAMARD DE SÉRIES RATIONNELLES. —
Soient $S,T \in \mathbb{C}[[z]]$ rationnelles. Alors $S \odot T$ est rationnelle.

Preuve — On peut remarquer qu'une série $R \in \mathbb{C}[[z]]$ est rationnelle ssi l'ensemble de ses décalées $(\gamma_z^*)^k R$ est de rang fini. Comme $\gamma_z^*(U \odot V) = \gamma_z^*(U) \odot \gamma_z^*(V)$, on a le résultat.

5 Génération aléatoire

5.1 Engendrer le hasard

Dès les premiers temps des ordinateurs, on les a appliqués à la simulation de phénomènes trop complexes pour être décrits exactement (on dit “bien modelisés”). Ils ont alors l’air d’être régis par le hasard (foudre, finance, météo).
La question est alors :
“Comment concevoir des générateurs de tirages automatiques qui soient équirépartis ?”
Notons qu’à partir de l’équirépartition on peut simuler les autres lois.

Exercice 5.1 *Soit une loi discrète donnée par un tableau*

k	$\cdots$	k_i	$\cdots$	k_n
$p(X=k)$	$\cdots$	p_i	$\cdots$	p_n

a) On suppose d’abord que $p_i = \frac{n_i}{d_i}$ et $d = ppcm(d_i)$, montrer comment simuler X avec un générateur équiréparti sur $[1..d]_N$.
b) Les p_i sont réels quelconques, expliquer sur un exemple (cf. la cible) pourquoi il vaut mieux utiliser les fractions continues pour approximer les p_i.

Le premier générateur de nombres aléatoires a été conçu selon la méthode du milieu du carré (middle square method), dont la recette est la suivante :
a) Prendre un nombre à 10 chiffres.
b) L’élever au carré.
c) Prendre les 10 chiffres du milieu.
Par exemple, avec le mini-programme suivant, on peut avoir la middle-square method.

```
>MSM:=proc(n)
local i,LL,S;
      LL:=convert(n^2,base,10);
      S:=0;
      for i from 6 to 15 do S := S + LL[i] * 10^(i-6) od;
      S
end;


>MSM(1234567890);
                          1578750190
>MSM(");
                          4521624250
>MSM(");
                          868581880
```

Question. — Comment évaluer les performances d’un tel générateur ?
La première idée est de faire une statistique sur un grand nombre de tirages.
La seconde est d’examiner les orbites.
ReponseSi le générateur est bon :
a) On met un certain temps avant de revoir une valeur (cf théorème de la période max).
b) Les tirages sont (semblent ??) indépendants (cf les générateurs à deux pas).

Exerc. Machine 5.2 *1) a) Faire 10^6 tirages par la* `MSM` *avec longueurs 3, 4, 8. Sont-ils équirépartis ?*
b) Faire calculer les orbites. Que constate-t-on dans ce cas ?
c) La situation s’améliore-t-elle quand on augmente la longueur ?
2) Quelles sont les plus petites périodes, les plus grandes, combien y a-t-il de cycles ?

5.2 Générateurs à un pas

Nous pouvons formaliser cette classe de générateurs comme suit.

Définition 5.3 *Soit F un ensemble fini, $x_0 \in F$ et $f : F \to F$ une application, On appelle générateur (à un pas) le triplet (F,f,x_0).*

Exemple 5.4 *A) Soit $E_{n+1} = [0,10^n - 1[$, l'ensemble des entiers à $n+1$ chiffres (écrits en base 10). On considère l'application f définie par les règles suivantes:*
1) Si $N = (a_n a_{n-1} \cdots a_0)_1 0$ est pair (i.e. si $a_0 = 0,2,4,6,8$) $f(N) = N/2$.
2 Sinon $f(N) = ((a_0 - 1)a_n a_{n-1} \cdots a_1)_1 0$.
On a par exemple, avec $n = 2;\ x_0 = 91$,

$$91 \to 09 \to 80 \to 40 \to 20 \to \mathbf{10} \to 05 \to 40 \to 20 \to \mathbf{10} \to \cdots$$

B) Soit m, un module. Considérer les générateurs à un pas avec $F = \mathbb{Z}/m\mathbb{Z}$ et f une fonction polynôme. Par exemple $x^2 + 1$.

Exerc. Machine 5.5 (EVELYN NELSON). — *On considère la fonction de transition suivante dans E_4 :*
i) Pour un nombre $N \in E_4$, $c(N)$ (resp. $d(N)$) désigne le réarrangement croissant (resp. décroissant) des chiffres de N.
ii) $f(N) = c(N) - d(N)$.
Soit C_4, l'ensemble des chiffres de la forme $k(1111)$ avec $0 \leq k \leq 9$ (ce sont les nombres qui ont leurs quatre chiffres égaux).
1) Montrer que $f(C_4) \subset C_4$ et que si $N \in E_4 - C_4$, on a $f(N) \in E_4 - C_4$.
On posera $D_4 = E_4 - C_4$.
2) Implémenter le générateur (x_0, f).
3) Faire la liste des cycles, que remarque-t-on ?

Exercice 5.6 *Faire dessiner les graphes de différentes fonctions.*

5.2.1 Paramètres

Les caractéristiques d'un générateur sont données par la proposition suivante :

Proposition 5.7 *Soit (F, f, x_0) un générateur à un pas. Alors :*
i) La suite $(x_n)_{n \geq 0}$ définie par

$$x_0;\ x_{n+1} = f(x_n)$$

est ultimement périodique, plus précisément,
ii) Il existe un plus petit indice μ dont la valeur est prise deux fois (indice d'entrée dans la période) et un plus petit entier λ (période) tel que, pour $n \geq \mu;\ k \geq 0$

$$x_n = x_{n+k\lambda}$$

L'ensemble des valeurs de la suite est $\{x_j\}_{0 \leq j \leq \mu+\lambda-1}$

Voici l'exemple d'un petit programme qui calcule l'orbite d'un élément dans les entiers modulo N.

```
>orb1:=proc(x0,f,B)
local i,LL,x;
      x:=x0:LL:=NULL:
      for i to N while not member(x,LL) do LL:=LL,x: x:=f(x) od
      [LL,x]
end;
```

```
> f := x- > x^2 + x + 1  mopt41;
                              f := x- > x^2 + x + 1;
> orb1(0,f,41);
                              [0,1,3,13,19,12,34,2,7,16,27,19]
> orb1(5,f,41);
                              [5,31,3,9,9]
> orb1(8,f,41);
                              [8,32,32]
> orb1(11,f,41);
                              [11,10,29,10]
```

```
> orb1(14,f,41);
                          [14,6,2,7,16,27,19,12,34,2]
```

Les paramètres sont

x_0	0	5	8	11	14
μ	4	3	1	1	2
λ	7	1	1	2	7

L'orbite de 0 a une période de 7 son cycle est $19{,}12{,}34{,}2{,}7{,}16{,}27$, à ce cycle se "raccrochent" d'autres branches, comme celle de l'orbite de 14 ($14{,}6{,}\mathbf{2}{,}\cdots$). L'orbite de 5 a une période de 1 (orbite apériodique).

Applications 5.8 *Pour la loi uniforme sur $[1{,}n]_{\mathbb{N}}$ une urne équitable a n boules peut bien faire l'affaire. Informatiquement, l'urne est remplacée par une fonction aléatoire du type* `rand()`. *Nous verrons le type de générateur comme en utilise Maple (c'est un générateur congruentiel linéaire, cf paragraphe suivant). Par exemple, on a :*

```
>rand(1000);
proc()
local t
global seed;
       seed:=irem(427419669081 *_seed,999999999989; t:=_seed;irem(t,1000)
end;
>_seed;
                          1
>rand(1000)();
                          81
>_seed;
                          427419669081
```

Noter que l'on obtient ainsi un "hasard faible" qui est, en général bon (voir toutefois les triplets [9]) pour la simulation parce qu'il est équiréparti (mais il n'est pas aléatoire)

Remarque 5.9 *La compréhension de ces paramètres est fondamentale et est à l'origine de nombreuses applications (notamment en factorisation : attaques de systèmes sécurisés du type RSA).*

Plus généralement, on a la notion de suite ultimement périodique.

Définition 5.10 *On dit qu'une suite x_n est ultimement périodique ssi il existe un certain rang N à partir duquel elle est périodique. Soit*

$$(\exists N)(\exists t>0)(\forall n\geq N)(x_{n+t}=x_n)$$

Note 5.11 *Une telle suite peut donc se mettre sous la forme*

$$(\cdots(x_N{,}x_{N+1}\cdots x_{N+t-1})^{\infty})$$

(comme $a(ba)^{\infty}$ par exemple), mais on constate qu'il y a une façon plus compacte que les autres de l'écrire sous cette forme (dans l'exemple précédent, c'est $(ab)^{\infty}$). Ceci détermine l'indice d'entrée dans le cycle et la période.

- paramètres
- opérations (produit cartésien,image)
- celles qui proviennent d'un G1P (rappel) et autres
- fonctions réversibles et orbites

Exercice 5.12 *La suite ultimement périodique $10(100)^{\infty}$ provient-elle d'un générateur à un pas ?*

Exerc. Machine 5.13 *1) a) Tester l'amplitude du générateur de hasard standard de Maple. Est-ce $[0..10^{12}-11]_{\mathbb{N}}$, comme semble l'indiquer l'aide ?*
b) Avec cette connaissance, former un tirage aléatoire de Bernouilli à deux dimensions.
c) Expérimenter et comparer avec le tirage précédent.
2) Faire une statistique sur la `MSM` *pour $N=2{,}3$ $(\lambda{,}n(\lambda))$ où $n(\lambda)$ est la nombre de points de départ qui aboutissent sur un cycle de longueur λ). (On aménagera la statistique de la façon la plus parlante possible en prenant par ex. λ dans des intervalles d'amplitude 100, et on raffinera certains intervalles.)*

Remarque 5.14 *L'image d'un générateur à un pas n'est pas toujours un générateur à un pas comme le montrent les exemples ci-dessous.*
i) Projection d'un générateur à deux pas :

$$\begin{pmatrix} x_{n+1} \\ x_{n+2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_n \\ x_{n+1} \end{pmatrix} \tag{55}$$

dans $(\mathbb{Z}/5\mathbb{Z})^2$ *et la première projection.*
On obtient la suite $[0,1,1,2,3,0,3,3,1,4,0,4,4,3,2,0,2,2,4,1]^\infty$*, on observe que le retour d'une valeur n'entraine pas nécessairement le retour de la suivante.*
ii) Cascade de congruences (technique utilisée par Maple) : On considère un générateur GCL1 $x_{n+1} = ax_n + b \bmod m$ *de période maximale (cf le théorème (5.17)). On le réduit modulo une autre congruence (*$m_1 < m$*) par exemple on peut montrer (et l'observer à la machine) que, pour que le résultat* $y_k \equiv x_k[m_1]$ *est aussi ultimement périodique et de période* λ_y *qui divise* $m = \lambda_x$*. Mais pour que* y_k *soit équiréparti, il faut et il suffit que* m_1 *divise* m*. Pourquoi ? Expliquer alors pourquoi* y_k *provient d'un générateur à un pas.*

Exercice 5.1 *A) Suites ultimement périodiques, méthodes de* BRENT*, de* FLOYD *[12] pp 308, 337, 346.*
B) Implémenter les différentes méthodes et comparer leurs performances sur les générateurs suivants:

$$\begin{aligned} x_{n+1} &= 3141692621x_n + 2718281829 \bmod 10^{10} \\ x_{n+1} &= x_n^2 + 5x_n + 1 \bmod 10^{32} \end{aligned}$$

5.2.2 Algorithmes de Brent et Floyd

Il est facile de déterminer les paramètres de la suite définie par un générateur à un pas lorsque celle-ci

- n'est pas trop longue
- est composée de nombres pas trop grands

mais, dans la pratique, on peut être amené à tester des suites de nombres de plus de 100 chiffres dont on ne sait pas la longueur de la période, il est alors hors de question de tout stocker, il faut se contenter de méthodes "locales" qui ne demandent de stocker, à chaque instant que deux valeurs. C'est le cas des algorithmes de Brent et Floyd.

Brent. —
On procède aux comparaisons suivantes:

x_0	x_1	$\cdots$	x_{2^k-1}	$\cdots$
x_1	x_2,x_3	$\cdots$	$x_{2^k}, \cdots x_{2^{k+1}-1}$	$\cdots$

jusqu'à ce que l'on trouve une coïncidence $x_{2^l-1} = x_m;\ m \in [2^l \cdots 2^{l+1} - 1]$. Ceci se produit forcément dès que $2^l - 1 \geq e$ (indice d'entrée dans le cycle, inconnu rappellons-le) et que $2^l \geq \lambda$. Soit donc $x_{2^l-1} = x_m;\ m \in [2^l, \cdots 2^{l+1} - 1]$, la coïncidence trouvée, on a $\lambda = m - (2^l - 1)$. Maintenant que l'on connait λ, on procède aux comparaisons

x_0	x_1	$\cdots$	x_k	$\cdots$
x_λ	$x_{\lambda+1}$	$\cdots$	$x_{\lambda+k}$	$\cdots$

la première a lieu pour $k = e$. L'étude de complexité de cet algorithme dépasse le cadre de ce cours, mais peut être trouvée dans [9, 12].

Floyd. —

On procède aux comparaisons suivantes

x_0	x_1	$\cdots$	x_k	$\cdots$
x_1	x_2	$\cdots$	x_{2k}	$\cdots$

la première coïncidence se produit pour le premier k multiple de λ qui dépasse e. L'indice k trouvé est un multiple de λ (mais pas nécéssairement λ), on procède alors aux comparaisons suivantes

x_0	x_1	$\cdots$	x_l	$\cdots$
x_d	x_{d+1}	$\cdots$	x_{d+l}	$\cdots$

la première conïncidence se produit pour $l = \mu$. On finit en déterminant λ par

x_μ	x_μ	$\cdots$	x_μ	$\cdots$
$x_{\mu+1}$	$x_{\mu+2}$	$\cdots$	$x_{\mu+k}$	$\cdots$

le premier k qui produit une égalité est $k = \lambda$.

5.2.3 Générateurs congruentiels linéaires

On appelle ainsi (en abr. GCL) les générateurs à un pas définis par $x_0;\ x_{n+1} = ax_n + b \text{ mopt } m$ où m est un module. Quand ils sont utilisés comme générateurs de hasard (benin) on cherche que leur période soit maximum (soit m). Voici quelques exemples donnés dans [12].

Exemple 5.15 *1)* $x_0 = 0;\ x_{n+1} = 4x_n + 1 \text{ mopt } 9$.
$(0 \to 1 \to 5 \to 3 \to 4 \to 8 \to 6 \to 7 \to 2 \to)^\infty$
2) $x_0 = 0;\ x_{n+1} = 2x_n + 1 \text{ mopt } 48$.
$0 \to 1 \to 3 \to 7 \to (15 \to 31 \to)^\infty$
3) $x_0 = 0;\ x_{n+1} = 3x_n + 1 \text{ mopt } 20$.
$(0 \to 1 \to 4 \to 12 \to)^\infty$
4) $x_0 = 0;\ x_{n+1} = 2x_n + 1 \text{ mopt } 5$.
$(0 \to 1 \to 3 \to 2)^\infty$ *et, si* $x_0 = 4$, $(4 \to)^\infty$
C'est ce type de générateur qu'utilisent les langages de programmation. Par exemple en Maple, on a:

```
>rand(1000);
proc()
local t
global seed;
       seed:=irem(427419669081 *_seed,999999999989; t:=_seed;irem(t,1000)
end;
>_seed;
                          1
>rand(1000)();
                          81
>_seed;
                          427419669081
```

Noter que l'on obtient ainsi un "hasard faible" qui est, en général bon (voir toutefois les triplets [9]) pour la simulation parce qu'il est équiréparti (mais il n'est pas aléatoire).

Exercice 5.16 *Vérifier la période du générateur précédent par* BRENT *et* FLOYD. *Comparer les performances de ces méthodes.*

On le théorème suivant qui indique comment fabriquer des générateurs de période maximale.

Théorème 5.17 *Pour qu'un GCL $x_{n+1} = ax_n + b \bmod m$ soit de période maximale il faut et il suffit que les conditions suivantes soient vérifées:*
a) b est inversible mopt m.
b) $a \equiv 1\ [p]$ pour tout p premier divisant m.
c) Si $4|m$ alors $a \equiv 1[4]$

Preuve —
Période maximale → Critère. —

- On a $x_n = a^n.x_0 + [n]_a.b$
- S'il y a une période maximale, elle passe par zéro. En posant $y_0 = x_{n_0} = 0$ on a $y_n = [n]_a.b$ et la période est max pour y_n, il existe n_1 tel que $[n_1]_a.b = 1$ d'où (a).
- Soit $p|m$, on a une projection $\mathbb{Z}_m \to \mathbb{Z}_p$, si la période est maximale dans $\mathbb{Z}_m$, c'est aussi la cas dans $\mathbb{Z}_p$ (on note $\mathbb{Z}_N$ une classe de restes modulo N, classique ou centrée par exemple).
- Si $a \not\equiv 1\ [p]$, on a, dans $\mathbb{Z}_p$, le point fixe $\frac{b}{1-a}$, ce qui est incompatible avec la période maximale, d'où (b).
- Si $4|m$, comme $a \equiv 1\ [2]$, on a $a \equiv 1{,}3\ [4]$. Si $a \equiv 3 \equiv -1\ [4]$, on a $x_{n+2} = -(-x_n + b) + b = x_n$, il n'y a donc pas de période maximale d'où (c).

Critère → Période maximale. —

- (Réduction du problème)
  1. Si $m = \prod p^{\alpha(p)}$, il suffit de démontrer que la période est maximale dans tous les $\mathbb{Z}_{p^\alpha}$
  2. Regardons le cycle de 0, qui est engendré par le générateur

     $$y_0 = 0; y_{n+1} \equiv a.y_n + b$$

     on a, comme précédement, $y_n \equiv [n]_a.b\ [m]$, et puisque b est inversible modulo m, il suffit de montrer que le critère entraine $[n]_a$ est de période m dans $\mathbb{Z}_m$.
- On montre que la période de $[n]_a$ est maximale à l'aide de l'identité

  $$[sq]_a = [s]_a.[q]_{a^s}$$

♣

Exerc. Machine 5.18 *Soit (F,f,x_0), un générateur. Au lieu d'une orbite, on peut vouloir tracer tout le graphe (la pieuvre) de f. Pour cela, il faut savoir "raccrocher" les éléments aux arbres qui se raccrochent aux cycles. La première méthode à laquelle on pense est de "scanner" les éléments non visités un par un. Mais, il est des cas (nous allons en voir deux) ou l'on peut créer un fonction antécédent qui prend un élément y et retourne l'ensemble des solutions de l'équation $y = f(x)$.*
1er Cas *: Les GCL à un pas $x_0;\ x_{n+1} = ax_n + b \bmod m$.*
1.1) $pgcd(a,m) = 1$ alors on calcule un inverse de a modulo m par la méthode de l'algorithme d'Euclide étendu qui donne les coefficients de Bezout $au + mv = 1$ `gcdex` *en Maple et MuPAD. On a donc $au \equiv 1$. Ainsi $y \equiv ax + b[m]$ est équivalent à $y - b \equiv ax$ et par multiplication par u, on a $u(y - b) \equiv x\ [m]$ solution unique.*
1.2) a et m ont des diviseurs communs. On calcule $d = pgcd(a,m)$ et par `gcdex` *on calcule des coefficients u,v tels que $au + mv = d$. Soit maintenant à résoudre $y \equiv ax + b[m]$. Il y a deux cas :*
1.2.1) $y - b \not\equiv 0[d]$ pas de solution en x et on se trouve à une feuille de l'arborescence.
1.2.2) $y - b = kd$ alors, en posant $a_1 = a/d$, $m_1 = m/d$ on a, de manière équivalente, $k \equiv a_1 x[m_1]$. Comme $pgcd(a_1,m_1) = 1$, on est ramené au cas précédent.

2ème Cas *: Les générateurs quadratiques du type $x_{n+1} \equiv ax_n^2 + bx_n + c[p]$ (p premier). Il faut résoudre $y \equiv ax^2 + bx + c[p]$ soit $ax^2 + bx + (c - y) \equiv 0[p]$. La discussion se fait classiquement par $\Delta := b^2 - 4a.(c - y)$ (voir le détail, utilisé dans les générateurs à deux pas (5.4.4)).*

5.3 Générateurs à deux pas

Q1 Si on voit sortir une suite ultimement pérodique d'une boîte noire, comment reconnaitre qu'elle provient d'un générateur à un pas ?
Q2 On constate que les "tirages" ne sont pas indépendants, peut-on améliorer cette situation ?
Q3 Comment appliquer BRENT et FLOYD au nouveau type de générateurs ?
Repartons de la suite $x_0,x_1;\ x_{n+2} = x_n + x_{n+1} \bmod 5$, c'est une suite périodique de période 20, telle qu'une valeur soit "fonction" des deux précédentes, ceci peut se formaliser de la façon suivante.

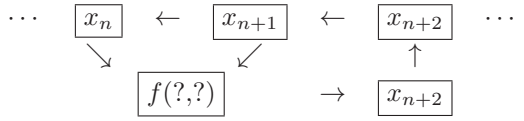
Définition 5.19 *Soit F un ensemble fini, $x_0,x_1 \in F$ et $f : F^2 \to F$ une application, On appelle générateur (à deux pas) le triplet $(F,f,(x_0,x_1))$. La suite associée au générateur est donnée par $x_0,x_1;\ x_{n+2} = f(x_n,x_{n+1})$.*

Exercice 5.20 *Montrer que la suite des "Fibonacci impairs" $a_n = F_{2k-1}$ vérifie la récurrence*

$$a_{n+2} = 3a_{n+1} - a_n$$

Exemple 5.21 *i) Soit $F = \mathbb{Z}/7\mathbb{Z}$ $f(x,y) = x^2 + y$ et $(x_0,x_1) = (1,2)$ On a la suite $1 \to 2 \to 3 \to 0 \to 2 \to 2 \to -1 \to 3 \to 4 \to -1 \to 1 \to 2$*
ii) $x_0 = 1$, $x_1 = 2$, $x_{n+2} \equiv 3x_n + x_{n+1} \bmod 9$

Note 5.22 *En utilisant la notation d'un "linear shift register" (utilisés abondamment en codage), on a*



Exercice 5.23 *i) Faire tourner* BRENT *et* FLOYD *sur des générateurs quadratiques du type de (5.18).*
ii) Faire tourner BRENT *et* FLOYD *sur les générateurs suivants (a,b=main; c,d=machine).* *a)* $(x_0,x_1) = (0,1)$ $x_{n+2} \equiv 2(x_n + x_{n+$ *c)* $(x_0,x_1) = (1,1)$ $x_{n+2} \equiv x_n^2 + x_{n+1}^2)$

La vectorisation (qui consiste à considérer la suite (x_n,x_{n+1})) permet de ramener l'étude d'un générateur à deux pas à celle d'un générateur à un pas.

5.3.1 Vectorisation et paramètres

Proposition 5.24 *i) La suite engendrée par un générateur à deux pas $(F,f,(x_0,x_1))$ est la première projection de la suite engendrée par le générateur à un pas $(F^2,g,(x_0,x_1))$ où g est donnée par $g(x,y) = (y,f(x,y))$. En particulier :*
ii) Cette suite est ultimement périodique, ses paramètres sont ceux de la suite "vectorisée" $\big((x_n,x_{n+1})\big)_{n\geq 0}$.

Remarque 5.25 *i) Le résultat précédent permet d'appliquer les algorithmes de* BRENT *et* FLOYD *aux générateurs à deux pas.*
ii) Avec les notations de Mupad, en notant $c := (x,y)$, on aurait

```
g(c)=(op(c,2),f(op(c,1),op(c,2)))
```

Exemple 5.26 *La suite de Fibonacci dans $\mathbb{Z}/5\mathbb{Z}$ se vectorise en*

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix};\ \begin{pmatrix} x_{n+2} \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_{n+1} \\ x_n \end{pmatrix} \tag{56}$$

(voir remarque (5.14)).
En utilisant le produit par blocs on a que

$$\begin{pmatrix} x_{n+1} & x_n \\ x_{n+2} & x_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}$$

ce qui prouve que la période est celle de $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ (modulo 5) et que l'indice est 0.

Un exemple dont on maîtrise les paramètres est celui des générateurs linéaires à deux pas (GL2P), dont la fonction de transition est $f(x,y) = ax + by$. Par exemple, la suite de Fibonacci modulo p $F_0 = 0,F_1 = 1;\ F_{n+2} = F_n + F_{n+1} \bmod p$ (en faire l'étude).

5.4 Générateurs du type GL2P

5.4.1 Généralités

La suite fournie par un GL2P est donnée par

$$\begin{cases} x_0,x_1 \\ x_{n+2} = ax_n + bx_{n+1} \quad \text{(RL2)} \end{cases}$$

la relation (RL2) s'appelle récurrence linéaire d'ordre deux.
À cette récurrence est attachée une l'équation $r^2 = a + br$ appellée équation caractéristique et qui provient de la considération suivante

Proposition 5.27 *Pour qu'une suite de puissances $(r^n)_{n\geq 0}$ (suite géométrique de premier terme 1) vérifie RL2, il faut et il suffit qu'elle la vérifie pour $n = 0$, soit*

$$r^2 = a + br \tag{57}$$

Pour concevoir des générateurs efficaces, il faut pouvoir les paramétrer de façon qu'ils aient une grande période. Pour cela il est nécéssaire de les décomposer en "petits" générateurs dont on maîtrise la période. Ce seront les suites $(r^n)_{n\geq 0}$ et $(nr^n)_{n\geq 0}$.

5.4.2 Combinaison de deux générateurs

La suite $(r^n)_{n\geq 0}$ mopt$p$ est produite par un GL1P et $(nr^n)_{n\geq 0}$ mopt$p$ est produite par un GL2P.

Proposition 5.28 *i) Soit $m > 0$ un entier et $\alpha,\beta,r_1,r_2 \in \mathbb{N}$ alors la suite de nombres $(\alpha r_1^n + \beta r_2^n)_{n\geq 0}$ est produite par le générateur*

$$\begin{cases} \alpha + \beta,\alpha r_1 + \beta r_2 \\ x_{n+2} = (r_1 + r_2)x_{n+1} - (r_1 r_2)x_n \quad \text{(RL2)} \end{cases}$$

ii) Dans les mêmes conditions que précedemment, la suite de nombres $(\alpha r^n + \beta n r^n)_{n\geq 0}$ est produite par le générateur

$$\begin{cases} \alpha + \beta,\alpha r_1 + \beta r_2 \\ x_{n+2} = 2rx_{n+1} - (r^2)x_n \quad \text{(RL2)} \end{cases}$$

Remarque 5.29 *En fait, il suffit de remarquer que, dans les deux cas, l'équation caractéristique doit avoir r_1,r_2 comme racines ($r = r_1 = r_2$) dans le deuxième cas. Celle-ci est alors $(r - r_1)(r - r_2) = 0$.*

5.4.3 Décomposition et calcul de la période d'un GL2P ($m = p$ premier).

Voici comment on procède:

1. On résout si possible l'équation caractéristique
2. On combine les résultats de façon à trouver les mêmes conditions initiales.

Par exemple pour Fibonnacci, on trouve $r = 3$ comme solution unique pour $m = p = 5$. L'équation caractéristique est $r^2 = 1 + r$ et ses racines dans différents "moduli" sont ($\{\} = \emptyset$ signifie qu'il n'y a pas de racine). On peut montrer d'ailleurs que les seuls p premiers tels que l'équation ait des racines sont de la forme $10k \pm 1$.

p	3	5	7	11	13	17	19	23	29	31	37	41
r	{}	{3}	{}	{4,8}	{}	{}	{5,15}	{}	{6,24}	{9,13}	{}	{7,35}

Dans le cas où il y a deux racines distinctes r_1,r_2 il existe toujours des coefficients u,v tels que l'on ait les deux premiers termes de le suite analysée soit $x_k = ur_1^k + vr_2^k;\ k = 0,1$ en ce cas la relation précédente reste vraie pour tout k.
Dans le cas où il n'y a qu'une racine r, il existe toujours des coefficients u,v tels que l'on ait les deux premiers termes de le suite analysée soit $x_k = (uk + v)r^k;\ k = 0,1$ en ce cas la relation précédente reste vraie pour tout k.

On peut énoncer :

Proposition 5.30 *i) Dans le cas où l'équation caractéristique admet deux racines distinctes r_1,r_2 il existe toujours des coefficients u,v tels que l'on ait les deux premiers termes de le suite analysée soit*

$$x_k = ur_1^k + vr_2^k;\ k = 0{,}1$$

en ce cas la relation précédente reste vraie pour tout k.
ii) Dans le cas où l'équation caractéristique n'admet qu'une racine r il existe toujours des coefficients u,v tels que l'on ait les deux premiers termes de le suite analysée soit $x_k = (uk + v)r^k;\ k = 0{,}1$ en ce cas la relation précédente reste vraie pour tout k.
iii) Les couples (u,v) déterminés précédemment sont uniques.

Nous ne traitons pas cette année le cas où il n'y a pas de racine. Voyons maintenant, avec plus de détails, la technique utilisée pour résoudre les équations du second degré modulo p (premier impair).

5.4.4 Carrés et équations du second degré dans $\mathbb{F}_p$

L'équation du second degré dans $\mathbb{Z}/p\mathbb{Z} = \mathbb{F}_p;\ p \neq 2$ se discute comme dans le cas classique (i.e. réel) en tenant compte des carrés. *Dans tout ce paragraphe, p désigne un nombre premier $\neq 2$.*

Proposition 5.31 *L'équation du second degré $ax^2 + bx + c = 0;\ a \neq 0$ se discute et résout selon le discriminant $\Delta = b^2 - 4ac$.*
i) Si $\Delta = \delta^2$ est un carré dans $\mathbb{F}_p$ alors l'équation admet les racines

$$\frac{-b-\delta}{2a};\ \frac{-b+\delta}{2a}$$

ii) Si Δ n'est pas un carré, l'équation n'a pas de solution.

Remarque 5.32 *Si $\Delta = 0$, les deux racines du cas (i) se confondent en une racine dite double $\frac{-b}{2a}$.*

Les carrés se calculent sur la "première moitié" de $\mathbb{F}_p$.

Proposition 5.33 *L'application donnée par $x -> x^2$ définit une bijection entre $[0..\frac{p-1}{2}]$ et l'ensemble des carrés de $\mathbb{F}_p$.*

$p=$	3			5				7					11						
			2			4	3			6	5	4			10	9	8	7	6
x	0	1	x	0	1	2		x	0	1	2	3	x	0	1	2	3	4	5
x^2	0	1	x^2	0	1	4		x^2	0	1	4	2	x^2	0	1	4	2	5	3

Exercice 5.34 *Résoudre et discuter dans $\mathbb{Z}/p\mathbb{Z}$:*
a) $x^2 + x + 1 = 0;\ p = 11{,}23{,}31$ *b)* $x^2 + 4x + 2 = 0;\ p = 17{,}29$
c) $x^2 + mx + 1 = 0;\ p = 7{,}11{,}13$ *d)* $x^3 + 1 = 0;\ p = 19{,}23$
e) $x^4 = m;\ p = 29{,}31$ *f)* $x^4 + x^2 + 1 = 0;\ p = 11{,}23{,}31$

5.4.5 Calcul de la période d'un GCL2

Rappelons ici la discussion.
La suite fournie par un GL2P est donnée par

$$\begin{cases} x_0,x_1 \\ x_{n+2} = \alpha x_{n+1} + \beta x_n \quad \text{(RL2)} \end{cases}$$

la relation (RL2) s'appelle récurrence linéaire d'ordre deux.
Voici comment on procède :

1. On résout si possible l'équation caractéristique

2. On combine les résultats de façon à trouver les mêmes conditions initiales.

Exercice 5.35 *i) ($p = 5$) Résoudre et discuter les 25 équations*

$$r^2 = \alpha r + \beta$$

pour $(\alpha,\beta) \in \mathbb{F}_5$ (essayer de regrouper des cas, voir des symétries etc..)
ii) Constater que si on tire les paramètres $(\alpha,\beta) \in \mathbb{F}_5^2$ "au hasard" (équidistribué), la probabilité que l'équation(57) ait $0,1,2$ racines est :

Nb. rac.	*0*	*1*	*2*
Proba.	$\frac{2}{5}$	$\frac{1}{5}$	$\frac{2}{5}$

iii) (Cas général) On considère maintenant $p \neq 2$ quelconque et on tire les paramètres $(\alpha,\beta) \in \mathbb{F}_p^2$ au hasard (équidistribué). Montrer que la probabilité que l'équation(57) ait $0,1,2$ racines est :

Nb. rac.	*0*	*1*	*2*
Proba.	$\frac{p-1}{2p}$	$\frac{1}{p}$	$\frac{p-1}{2p}$

iv) Faire des expériences machine sur différentes valeurs de p.
a) Exhaustives : examiner **tous** *les couples (α,β).*
b) Aléatoires (grandes valeurs de p) : utiliser un générateur de hasard.

On rappelle que, pour p premier, un élément non nul $g \in \mathbb{Z}/p\mathbb{Z}$ vérifie toujours $g^{p-1} = 1$ (Fermat). Mais il peut y avoir des puissances plus petites. Le plus petit $k > 0$ telle que $g^k = 1$ est appellé l'ordre de g. Voici à titre d'exemple les ordres des $g \neq 0$ dans $\mathbb{Z}/19\mathbb{Z}$.

g	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ord(g)	1	18	18	9	9	9	3	6	9	18	3	6	18	18	18	9	9	2

Les périodes sont données par la tableau suivant:

racines		$x_n =$	période
0 est rac. ou $0 \in \{u,v\}$		$(x_n)_{n\geq 1}$ vient d'un GCL1.	
$r_1,r_2 \neq 0$	$r_1 \neq r_2$	$x_n = ur_1^n + vr_2^n$	$ppcm(ord(r_1),ord(r_2))$
$u,v \neq 0$	$r_1 = r_2 = r$	$x_n = (un+v)r^n$	$p \times ord(r)$

Par exemple, pour Fibonacci mod 5, on a

$$x_n \equiv 3^n + n.3^n \mathrm{mopt} 5$$

ce qui explique la période de $20 = 5.4$. Pour Fibonacci modulo 11, on a

$$x_n \equiv 10 * 4^n + 2 * 8^n \equiv -4^n + 2 * 8^n \mathrm{mopt} 11$$

ce qui explique la période de 10 car $ord(4) = 5;\ ord(8) = 10$

Exercice 5.36 *a) Concevoir un programme qui donne les ordres des éléments modulo p.*
b) Faire, pour différentes valeurs de p, la statistique $p \to$ `(maxord(p),numaxord(p))` *où* `maxord` *est l'ordre maximal et* `nummaxord` *est le nombre des éléments d'ordre maximal. Que remarque-t-on ?*

La notion de générateur a déjà été utilisée pour RSA, nous la rappellons ici.

Proposition/Definition 5.37 *i) Pour tout p premier, il existe des éléments $g \in \mathbb{Z}_p$ tels que*

$$\mathbb{Z}_p - \{0\} = \{g^k\}_{0\leq k\leq p-2} = \{1,g,g^2,g^3,\cdots g^{p-2}\}$$

c'est à dire que les puissances de g décrivent tous les éléments non-nuls.
ii) De tels éléments (exactement ceux qui sont d'ordre maximal soit $p-1$) sont appellés génŕateurs.
iii) (Calcul de l'ordre d'un élément quelconque) Si g est un générateur, on a

$$ord(g^k) = \frac{p-1}{pgcd(k,p-1)}$$

la conclusion de cette étude est que la période d'un générateur à deux pas avec solutions de l'EC est : soit un diviseur de $p-1$ (cas où il y a deux racines), soit $p \times ord(r)$ (cas où il n'y a qu'une seule racine r) et donc la plus longue période réalisable avec un générateur "à racine(s)" est $p(p-1)$. On verra aussi que la plus longue période d'un générateur "sans racine" est p^2-1. Soit pour les premiers nombres premiers $\neq 2$

p	3	5	7	11	13	17	19	23	29	31	37	41	43	47
$p(p-1)$	6	20	42	110	156	272	342	506	812	930	1332	1640	1806	2162
p^2-1	8	24	48	120	168	288	360	528	840	960	1368	1680	1848	2208

5.5 Autres générateurs

Knuth donne dans [9] (p26), un générateur de D.J. Mitchell et D.P Moore

$$X_n = (X_{n-24} + X_{n-55}) \text{ mopt } m$$

où les termes initiaux $X_0 \cdots X_{54}$ sont des entiers arbitraires non tous pairs. Il a pour période $2^f(2^{55}-1)$ pour $m = 2^e$ et $0 \leq f < e$ (cf [9] p34 Ex. 11) et se programme à l'aide d'une liste cyclique à 55 pas.

5.6 Généateurs à k pas

5.7 Énumérer, classer, indexer

5.8 Répartitions équitables et moins équitables

6 Représentation des données et calcul

6.1 Généralités

Nous connaissons déjà, avec les mathématiques ou plus simplement avec l'arithmétique élémentaire le probmlème de la représentation des données. Dans le premier cas, toutes les notations en sont un exemple, avec le déploiement kaleïdoscopique des multiples possibilités dont la puissance et le champ d'application varie autant que la multiplicité des languages et des disciplines : qu'on considère seulement la notation algébrique et trigonométrique des complexes, les angles d'EULER etc..., de l'autre côté on a déjà les différentes représentations d'un nombre par une suite de symboles (numération) et les algorithmes qui leurs sont associés. Il est connu qu'une notation appropriée peut rendre un résultat limpide et qu'une mauvaise notation peut rendre un résultat ou la pratique d'un calcul impossible (qu'on s'amuse seulement à additionner deux complexes sous forme trigonométrique). Ainsi, on peut calculer sur les représentations au lieu de calculer sur les quantités et on espère que les notations sont suffisament bien choisies pour donner des algorithmes simples.
Les structures de données de base ("briques") les plus courantes sont :

- Listes, vecteurs (d'entiers, de scalaires etc..)
- Matrices
- Mots
- Fonctions
- Diagrammes (graphes, chemins - de Dyck etc.. -, DAG, arbres)

7 Premiers pas & Résumé des séances

Les premiers rappels concernent les notions suivantes

- Alphabet (de commande)
- mots (y compris le mot vide)
- exemples de statistiques sur les mots (degrés partiels),
  les langages (distributions de longueurs)

- concaténation
- longueur, degrés partiels (nombre d'occurrence)
- langages
- codes rationnels : le premier exemple de $(a^*b)^*$
- premières identités rationnelles $(a^*b)^* = (a+b)^*b$ et $(a+b)^* = (a^*b)^*a^*$

Les dernières séances.

- le 18.03 : exemple de l'automate à deux états qui reconnait $(a^*b)^*a^*$, réalisation de la factorisation par les retours à l'état $\boxed{2}$. Début des multiplicités
- le 25.03 :
  - $\mathcal{A}(w)$
  - forme "sandwich" du calcul
  - comportement, exemples
  - étoile d'une matrice
  - "lemme des oreilles de Mickey"
  - exemples de calcul
  - expressions rationnelles
- le 08.04 :
  - Retour sur l'étoile d'un langage, d'une série
- le 15.04 :
  - exemples détoiles,
  - semi-anneaux par les diagrammes

8 Systèmes et Calcul

8.1 Introduction

Exemples d'automates booléens, stochastiques, de comptage, de plus courts chemins. Les semi-anneaux associés sont : $\mathbb{B}$, $\mathbb{R}_+$,$\mathbb{N}$,$([0,+\infty]$,min,$+)$.

8.2 Description de la structure d'automate

8.2.1 Graphe pondéré

L'élément de base de ces graphes est la flêche $A = q_1 \overset{a|\alpha}{\to} q_2$ avec $q_i \in Q$, $a \in \Sigma$, $\alpha \in k$ où Q est un ensemble d'états, Σ un alphabet et k, un semi-anneau[5]. Pour un tel objet, on définit, selon les conventions générales de la théorie des graphes,

- $t(A) := q_1$ ("tail": queue, source, origine)
- $h(A) := q_2$ ("head" tête, but, extrémité)
- $l(A) := a$ ("label" étiquette)
- $w(A) := \alpha$ ("weight" poids).

Un *chemin* est une suite d'arêtes $c = A_1A_2\cdots A_n$ (c'est un mot en les arêtes et sa longueur est n) telle que $h(A_k) = t(A_{k+1})$ pour $1 \le k \le n-1$ pour un tel chemin $t(c) = t(A_1), h(c) = h(A_n), l(c) = l(A_1)l(A_2)\cdots l(A_n)$ (concaténation), $w(c) = w(A_1)w(A_2)\cdots w(A_n)$ (produit dans le semi-anneau).
Par exemple pour le chemin de longueur 3 suivant ($k = \mathbb{N}$),

$$u = p \overset{a|2}{\to} q \overset{b|3}{\to} r \overset{c|5}{\to} s \tag{58}$$

on a $t(u) = p$, $h(u) = s$, $l(u) = abc$, $h(u) = 30$.

5. Nous verrons plus bas que les axiomes de la structure de semi-anneau sont contraints par la définition même du système de transitions ainsi obtenu.

Le poids d'un ensemble de chemins de même source, but et étiquette est la somme des poids des chemins de cet ensemble. Ainsi, si

$$\mathbf{q}1 \quad \begin{array}{c} \overset{u|\alpha}{\rightarrow} \\ \overset{u|\beta}{\rightarrow} \end{array} \quad \mathbf{q}2 \tag{59}$$

le poids de cet ensemble de chemins est $\alpha + \beta$. On a donc que les poids se multiplient en série et s'aditionnent en parallèle. Les diagrammes suivants montrent la nécessité des axiomes de semi-anneau.

Diagramme	Identité	Nom
$\begin{array}{c} \overset{a\|\alpha}{\rightarrow} \\ p \overset{a\|\beta}{\rightarrow} q \\ \overset{a\|\gamma}{\rightarrow} \end{array}$	$\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$	Associativité de $+$
$p \begin{array}{c} \overset{a\|\alpha}{\rightarrow} \\ \underset{a\|\beta}{\rightarrow} \end{array} q$	$\alpha + \beta = \beta + \alpha$	Commutativité de $+$
$p \begin{array}{c} \overset{a\|\alpha}{\rightarrow} \\ \underset{a\|0}{\rightarrow} \end{array} q$	$\alpha + 0 = \alpha$	Élément neutre (droite) de $+$
$p \begin{array}{c} \overset{a\|0}{\rightarrow} \\ \underset{b\|\beta}{\rightarrow} \end{array} q$	$0 + \beta = \beta$	Élément neutre (gauche) de $+$
$p \overset{a\|\alpha}{\rightarrow} q \overset{b\|\beta}{\rightarrow} r \overset{c\|\gamma}{\rightarrow} s$	$\alpha(\beta\gamma) = (\alpha\beta)\gamma$	Associativité de $\times$
$\begin{array}{c} \overset{a\|\alpha}{\rightarrow} \\ p \underset{a\|\beta}{\rightarrow} q \end{array} \overset{b\|\gamma}{\rightarrow} r$	$(\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma$	Distributivité (droite) de $\times$ sur $+$
$p \overset{a\|\alpha}{\rightarrow} \begin{array}{c} \overset{b\|\beta}{\rightarrow} \\ q \underset{b\|\gamma}{\rightarrow} \end{array} r$	$\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$	Distributivité (gauche) de $\times$ sur $+$
$p \overset{a\|\alpha}{\rightarrow} q \overset{b\|1_k}{\rightarrow} r$	$\alpha \times 1_k = \alpha$	Élément neutre (droite) de $\times$
$p \overset{a\|1_k}{\rightarrow} q \overset{b\|\beta}{\rightarrow} r$	$1_k \times \beta = \beta$	Élément neutre (gauche) de $\times$

8.2.2 Structure et comportement des automates

Un automate à poids ou pondéré ("automaton with weights") est la donnée de trois éléments vectoriels (I,M,T):

- Un vecteur d'entrée $I \in k^{1\times Q}$
- Une famille (indexée à A) de matrices de transition $M : A \to k^{Q\times Q}$
- Un vecteur de sortie $T \in k^{Q\times 1}$

La donnée des transitions (M) est équivalente à celle d'un graphe pondéré dont les sommets sont Q, l'alphabet A et les poids sont pris dans k. De plus celle de I (resp. T) correspond à la donnée de flêches rentrantes (resp. sortantes) marquées avec des poids. Dans tout ce processus, on peut ne pas indiquer les flêches de poids nul.
Ce type d'automates généralise les automates (booléens) de la théorie des langages (que l'on obtient alors pour $k = \mathbb{B}$) est une machine qui prend un mot en entrée et retourne un coefficient (dans k) en sortie. Son comportement est donc une fonction $\mathcal{A} : A^* \to k$ (que l'on peut noter, de manière équivalente, comme une série $\mathcal{A} = \sum_{w\in A^*} \mathcal{A}(w)w$).
Calcul du poids $\mathcal{A}(w)$. —
On étend d'abord la fonction de transition M à A^* par

$$M(\epsilon) = I_{Q\times Q},\ M(w) = M(a_1a_2\cdots a_n) = M(a_1)M(a_2\cdots M(a_n) \tag{60}$$

où $I_{Q\times Q}$ est la matrice identité de format $Q \times Q$. Le calcul du poids d'un mot est alors, par définition,

$$\mathcal{A}(w) := IM(w)T \tag{61}$$

d'après la règle de multiplication des matrices, on a bien que $IM(w)T$ est une matrice de format 1×1 et donc un élément de k. Le lien avec le graphe de l'automate est donné par la proposition suivante :
Proposition 8.1 *Soit, pour deux états r,s et un mot $w \in A^*$*

$$\mathcal{A}^{r,s}(w) := I_r \left(\sum_{\substack{c,\ \text{chemin}\ l(c)=w \\ t(c)=r,\ h(c)=s}} weight(c) \right) T_s \tag{62}$$

alors

$$\mathcal{A}(w) = \sum_{r,s\in Q} \mathcal{A}^{r,s}(w) \tag{63}$$

Cette proposition a le sens intuitif suivant :

1. l'équation (89) donne le poids calculé comme au paragraphe précédent
   - on fait le bilan parallèle (c'est à dire une somme) des poids des chemins qui joignent r à s
2. on multiplie (à gauche si c'est non commutatif) par le poids d'entrée en r
3. on multiplie (à droite si c'est non commutatif) par le poids de sortie en s

8.2.3 Premiers automates

1. Longueur totale $\sum_{w\in A^*} |w|w$
2. Comptage des a, $\sum_{w\in A^*} |w|_a w$ et des b, $\sum_{w\in A^*} |w|_b w$
3. Produit des degrés partiels $\sum_{w\in A^*} |w|_a|w|_b w$
4. Autres produits $\sum_{w\in A^*} F_{|w|}|w|w$, $\sum_{w\in A^*} F_{|w|_a}|w|_b w$

8.2.4 Composition des automates

Somme et multiplication par un coefficient constant

Produit de Hadamard

Produit (de concaténation)

Nous avons vu que nous pouvions coder de "l'infini dans du fini" en considérant les suites ultimement périodiques que sont les développements illimités des rationnels. Nous allons voir qu'il en est de même pour la production des automates finis, en effet, un automate fini, dès qu'il possède un chemin réussi qui comporte un boucle, reconnaît un langage infini.

Exercice 8.2 *Montrer que cette condition est suffisante, autrement dit, si aucun chemin réussi ne comporte de boucle, alors le langage reconnu par l'automate est fini.*

Commençons par un exemple : On considère un automate (booléen), d'ensemble d'états Q et dont les transitions sont étiquetées par un alphabet A. Cet automate, via la correspondance (graphes $\leftrightarrow$ matrices) peut être vu comme un triplet (I,T,M) avec :

$$\begin{cases} \bullet \text{ Un vecteur d'entrée } I \in k^{1\times Q} \\ \bullet \text{ Une famille de matrices de transition } M : A \to k^{Q\times Q} \\ \bullet \text{ Un vecteur de sortie } T \in k^{Q\times 1} \end{cases}$$

Dans les automates usuels, les scalaires sont pris dans $\{0,1\}$. Si on considère ces nombres comme des entiers naturels, l'opération $w \to IM(w)T$ donne le nombre de chemins réussis. Une expression rationnelle du comportement de l'auromate (tenant compte des multiplicités) résulte du calcul suivant

$$\sum_{w\in\Sigma^*} (IM(w)T)w = I(\sum_{w\in\Sigma^*} M(w)w)T = I(Id_n - \sum_{a\in\Sigma} M(a)a)^{-1}T$$

si on note $M_\Sigma = \sum_{a\in\Sigma} M(a)a$, on a $M_\Sigma^* = (Id_n - \sum_{a\in\Sigma} M(a)a)^{-1}$. C'est la matrice dont l'entrée d'adresse (i,j) est la somme

$$\sum_{\substack{w \text{ étiquette} \\ \text{un chemin de } i \text{ vers } j}} (\text{nb de chemins } i \to j \text{ d'étiquette } w)w$$

par exemple la matrice

$$M_\Sigma = \begin{pmatrix} a & a \\ b & 0 \end{pmatrix}$$

a pour étoile

$$M_\Sigma^* = \begin{pmatrix} (a+ab)^* & (a+ab)^*a \\ b(a+ab)^* & (ba^*a)^* \end{pmatrix}$$

il est facile de voir que les séries associées sont sans multiplicité (i.e. pour (i,j) et w donnés il existe au plus un chemin d'étiquette w), mais ce n'est pas le cas pour

$$Q_\Sigma = \begin{pmatrix} a & a \\ b & a \end{pmatrix}$$

qui a pour étoile

$$Q_\Sigma^* = \begin{pmatrix} (a+aa^*b)^* & (a+aa^*b)a^*a \\ a^*b(a+aa^*b)^* & (a+ba^*a)^* \end{pmatrix}$$

Exercice 8.3 *1) Dessiner les automates (sans vecteurs d'entrée et sortie) associés aux matrices M_Σ,Q_Σ.*
2) a) Montrer, en utilisant un raisonnement sur les chemins dans un graphe étiqueté convenable, que pour deux lettres, on a $(a+b)^* = (a^*b)a^*$ (élimination de Lazard monoïdale).
b) Appliquer cette identité pour trouver une autre forme de $(a+aa^*b)^*$.
c) Montrer que $a^*aa^* = a\frac{1}{(1-a)^2} = \sum_{n\geq 1} na^n$.
d) Si un mot ne se termine pas par b, sa multiplicité dans $(a^*aa^*b)^*$ est nulle, mais s'il s'écrit $w = a^{n_1}ba^{n_2}b\cdots a^{n_k}b$, on a $(w,(a^*aa^*b)^*) = n_1+n_2+\cdots n_k$. En déduire le développement (i.e. les multiplicités des mots) de $(a^*aa^*b)^*a^*$, puis des 4 coefficients de la matrice Q_Σ^*.
3) a) Soit l'alphabet à quatre lettres $\Sigma = \{a_{11},a_{12},a_{21},a_{22}\}$, montrer directement en raisonnant sur les chemins, que si

$$G = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

on a

$$G^* = \begin{pmatrix} A_{11} & A_{11}a_{12}a_{22}^* \\ a_{22}^*a_{21}A_{11} & A_{22} \end{pmatrix};\ avec\ A_{11} = (a_{11}+a_{12}a_{22}^*a_{21})^*,\ A_{22} = (a_{22}+a_{21}a_{11}^*a_{12})^*$$

b) Expliquer en quoi ces formules fournissent un algorithme permettant de calculer l'étoile de toute matrice de séries propres.

Exercice 8.4 *(L'lphabet est $\{a,b\}$) Soit L_n le langage fini formé des mots w tels que*

$$|w|_a + 2|w|_b = n\ .$$

Pour vérification, on a

$$L_0 = \{\epsilon\},\ L_1 = \{a\},\ L_2 = \{aa,b\},\ L_3 = \{aaa,ab,ba\}$$

a) Écrire les termes L_4 et L_5.
b) Calculer $|L_n|$ à l'aide d'une récurrence simple.
c) Montrer que $SG = \sum_n |L_n|t^n = (t+t^2)^* = \frac{1}{1-t-t^2}$.
d) Faire le lien avec le nombre de pavages d'un rectangle $2\times n$ par des dominos 2×1 ([11] pp 321) comment coder les pavages, les énumérer, les générer.
e) À l'aide des décalages, former l'automate qui reconnait la série S.

On a un analogue parfait de ce qui se passe pour les rationnels positifs. Plus précisément :

Exercice 8.5 *A) On considère les arbres $1-2$ qui sont les arbres à 1 ou deux fils.*
À chaque arbre $1-2$, dont les noeuds internes sont signés par "$+$" s'ils ont deux fils et "$()^{-1}$" s'ils en ont un on fait correspondre une fraction (i.e. son évaluation avec les feuilles en 1) donnée par la règle récursive

$$ev(\bullet) = 1;\ ev((\mathcal{A}_1,\mathcal{A}_2)) = ev((\mathcal{A}_1)+ev(\mathcal{A}_2);\ ev((\mathcal{A})) = \frac{1}{ev(\mathcal{A})}$$

montrer que l'ensemble des valeurs obtenues est $\mathbb{Q}_+^*$. Est-ce que la représentation est unique ? Est-ce qu'elle englobe les fractions continues ? Comment caractériser les arbres qui les donnent ?
B) On considère les séries sur un alphabet A (i.e. fonctions $A^* \to k$ où k est un semi-anneau (i.e. suffisant pour

faire la calcul matriciel).
a) Montrer que les conditions suivantes sont équivalentes :
i) La série S est l'évaluation d'une expression rationnelle.
ii) La série S est combinaison linéaire d'un ensemble de séries S_1,S_2,$\cdots S_n$ qui est (linéairement) stable par décalages soit

$$(\forall x \in A)(\forall i \in [1..n])(x^{-1}S_i = \sum_{0 \leq j \leq n} \mu_{i,j}(x)S_j$$

iii) Il existe $\lambda \in K^{1\times n}$, $\mu : A \to K^{n\times n}$,$\gamma \in K^{1\times n}$ tels que pour tout $w \in A^*$, $(S,w) = \lambda\mu(w)\gamma$ (où $\mu()$ dénote encore l'extention de μ à A^*).

Lorsque l'on a une partie $X \in A^*$, on peut se demander :
Quel est le langage $L(X)$ engendré par X ?
c'est à dire les suites finies d'instructions (i.e. le sous-monoïde engendré). On a $L(X) = \sum_{n\geq 0} X^n$ à coefficients dans $\mathbb{B}$. La même somme à coefficients dans $\mathbb{N}$ contient plus d'informations (soit le nombre de façons d'obtenir w comme produit de facteurs dans X).
Automates. Automates à multiplicité (notion de coût). Comportement d'un automate. Séries (exemples), rationnelles.
Passage SGO $\leftrightarrow$ Aut $\leftrightarrow$ Exp. rat.
Exemples de $\mathbb{N}$ et $\mathbb{Z}$ automates.
Séries génératrices (rationnelles -arbres de Fibonacci- et non rationnelles -arbres binaires, chemins de Dyck-). Résolution des premières récurrences, décalage et Δ. Complexité du comptage des boucles. Arbres $1-2$.

9 Séries

9.1 Introduction

Les séries jouent le rôle d'un outil très important en MATHÉMATIQUES (algèbre : réalisation explicites de complétés, analyse : développement de fonctions analytiques, développements asympotiques, géométrie : classes de singularités, probabilités : séries génératrices de probabilités), en INFORMATIQUE (combinatoire - algébrique, énumérative, analytique -, analyse d'algorithmes, grammaires d'objets, théorie des langages), en PHYSIQUE (probléme des moments, développements perturbatifs, solutions d'ED) et dans l'art de l'ingénieur (électronique : transformée en "z", séries de Fourier, développement de caractéristiques non linéaires, Linear Shift Register) pour ne citer que quelques unes de leurs applications.

9.2 Fonctions : notation fonctionnelle et sommatoire

Les séries se présentent sous forme d'une somme (finie ou infinie)

$$\sum_{m\in M} \text{coefficient}(m)m \tag{64}$$

l'ensemble M pouvant être un ensemble de monômes ou un ensemble de fonctions bien choisies (séries d'exponentielles, séries de Fourier, de Dirichlet). Dans ce cours, nous nous limiterons au cas des monômes. Dans ce cadre rentrent

1. les séries et polynômes d'une ou plusieurs variables (commutatives, non commutatives ou partiellement commutatives)
2. les fonctions symétriques
3. les séries et polynômes de Laurent, de Malcev
4. les séries d'exponentielles, de Bertrand et de Dirichlet

le caractère commun de ces séries est que l'ensemble des monoômes M est fermé (stable en français) pour la multiplication.

10 Séries de plusieurs variables

10.1 Les scalaires de l'Informatique

La théorie des langages traite d'ensembles de mots (i.e. de parties d'un monoïde libre), la théorie des systèmes traite des actions, celle des tranducteurs des correspondances (avec ou sans poids), celle des chaînes de Markov considère des proabailités de transition. En général, l'Informatique Théorique développe des calculs

- dans le cadre logique (où l'on a volontiers $x + x = x$)
- dans le cadre modulaire (i.e. modulo p où $\underbrace{x + x + \cdots x}_{p \text{ fois}} = 0$)
- dans le cadre probabiliste et celui des coûts (souvent réels)
- avec les traitement du signal (où les fonctions peuvent être à valeur dans $\mathbb{C}$)
- dans le cadre des transducteurs (coefficients non-commutatifs).

Voilà pourquoi on généralise la structure d'anneau en celle de semi-anneau (intuitivement, un anneau sans l'opération $x \mapsto -x$).

10.2 Définition d'un semi-anneau

Définition 10.1 *Soit K, un ensemble. Une structure de semi-anneau sur K est définie par la donnée de deux lios de composition internes sur K, $(+,\times)$ telles que*
SR1) $(K,+)$ est un monoïde commutatif d'élément neutre 0_K
SR2) $(K,\times)$ est un monoïde d'élément neutre 1_K
SR3) $\times$ est distributif à droite et à gauche sur l'addition
SR4) 0_K est un annulateur (ou élément absorbant) $0_k x = x 0_k = 0_k$

Un semi-anneau sera dit commutatif ssi $\times$ l'est.

Remarque 10.2 *SR4) n'est pas conséquence des autres, comme le montre l'exemple de $(\mathbb{N},sup,+)$*

Exemple 10.3 *i) $(\mathbb{B},+,.)$ (semi-anneau booléen).*
ii) $(\mathbb{N},+,.)$ (entiers naturels)
iii) $(\mathbb{R} \cup \{-\infty\},max,+)$
ces semi-anneaux ne sont pas des anneaux.
iv) Tous les anneaux (et en particulier les corps) sont des semi-anneaux.

Définition 10.4 *i) Un morphisme de semi-anneau $\phi : (K_1,+,.) \mapsto (K_2,+,.)$ est une application $\phi : K_1 \mapsto K_2$ qui est un morphisme pour les structures de monoïdes (additifs et multiplicatifs) de K_i; $i = 1,2$.*
ii) Si $K_1 \subset K_2$ et que ϕ est l'injection canonique, on dit que K_1 est un sous-semi-anneau de K_2. Si ϕ est surjective, on dit que K_2 est un quotient de K_1.

Exemple 10.5 *i) L'application $s : \mathbb{N} \mapsto \mathbb{B}$ telle que $s(0) = 0$; $s(n) = 1$ pour $n > 0$ est un morphisme de semi-anneaux. Ce morphisme est non prolongeable à $\mathbb{Z}$.*
ii) Semi-anneaux d'Eilenberg. — On peut munir $\mathbb{K}_{e,p} = \{0,1,\cdots e,\cdots e+p-1\}$ d'une seule structure de semi-anneau telle que l'application $r : \mathbb{N} \mapsto \mathbb{K}_{e,p}$ définie par

$$r(n) = \begin{cases} n \text{ si } 0 \leq n \leq e+p-1 \\ (n - e \ mod \ p) + e \text{ sinon} \end{cases} \tag{65}$$

soit un morphisme de semi-anneaux.

Exercice 10.6 *Soit $(K,+,.)$, un semi-anneau. Montrer que K est trivial (c'est à dire $|K| = 1$) ssi $1_K = 0_K$. Donner alors ses tables d'addition et de multiplication.*

Exercice 10.1 *Soit $(K,+,.)$ un semi-anneau.*
a) Montrer qu'il existe un seul morphisme de semi-anneaux $\phi : \mathbb{N} \mapsto K$ et que son image est le sous monoïde de $(K,+)$ engendré par 1_K.
b) Montrer que $\phi(\mathbb{N})$ est un sous semi-anneau de K et qu'il est soit isomorphe à $\mathbb{N}$ soit isomorphe à un, et un seul, des $\mathbb{K}_{e,p}$.

Exercice 10.2 *1) a) Soit $(M,+)$, un monoïde commutatif*

$$End(M) = \{\phi \in M^M | \phi(0) = 0 \text{ et } (\forall x,y \in M)\big(\phi(x+y) = \phi(x) + \phi(y)\big)\} \tag{66}$$

montrer que $(End(M), +, \circ)$ est un semi-anneau.
b) Soit $(K, +, \times)$ un semi-anneau et, pour tout $\alpha \in K$, soit $h_\alpha : x \mapsto \alpha x$ de K dans K. Montrer que $\alpha \mapsto h_\alpha$ est un morphisme de semi-anneaux $K \mapsto End(K,+)$.
2) Soit $(M,*)$ un monoïde tel que $(\forall x \in M)(x*x = x)$. Montrer que le relation $\prec$ définie par $x \prec y \Longleftrightarrow x*y = y$ est une relation d'ordre.
3) On suppose M de plus commutatif et on note $+$ sa loi (Commutative band).
a) Montrer que M est semi-réticulé supérieurement (toute paire admet une borne supérieure) et que $x + y = sup(x,y)$
b) Montrer que, dans M, on a identiquement $sup(x + t, y + t) = sup(x,y) + t$.
*4) [**?**] VI, ¶1 Ex 13). —*
On dit qu'un monoïde M (commutatif) est semi-réticulé inférieurement (ou semi-réticulé pour simplifier), si c'est un monoïde ordonné, si $inf(x,y)$ existe pour tout couple d'éléments x, y de M et si l'on a

$$inf(x + z, y + z) = inf(x,y) + z$$

quels que soient x,y,z dans M. Démontrer qu'on a alors les identités :

$$inf(x,z) + inf(y,z) = inf(x + y, z + inf(x,y,z))$$

$$inf(x,y,z) + inf(x + y, y + z, z + x) = inf(x,y) + inf(y,z) + inf(z,x)$$

*Déduire de la première de ces relations que $x \leq z$ et $y \leq z$ entraînent $x + y \leq z + inf(x,y)$. Montrer que la prop. 11 de [**?**] VI, ¶1 p. 14 et ses corollaires sont valables dans un monoïde semi-réticulé.*
b) Soient M un monoïde semi-réticulé, N un sous-monoïde de M tel que $inf_M(x,y) = inf_N(x,y)$ pour x,y dans N et que les éléments de N soient symétrisables dans M.
Montrer que le sous-groupe G de M formé des éléments $x - y$, pour x,y dans N, est réticulé (c'est à dire que toute paire admet un borne supérieure et inférieure).
5) Soit $(M,*)$, un monoïde semi-réticulé supérieurement, c'est à dire que l'on ait identiquement

$$sup(t * x, t * y) = t * sup(x,y);\ sup(x * t, y * t) = sup(x,y) * t \tag{67}$$

Quelles sont les conditions pour que $(M,sup,*)$ soit un semi-anneau ?

10.3 Automates à multiplicités

10.3.1 Graphes pondérés (structure de transition)

L'élément de base de ces graphes est la flèche $A = q_1 \overset{a|\alpha}{\to} q_2$ avec $q_i \in Q$, $a \in \mathbb{A}$, $\alpha \in k$ où Q est un ensemble d'états, $\mathbb{A}$ un alphabet et k, un semi-anneau[6]. Pour un tel objet, on définit, selon les conventions générales de la théorie des graphes,

- $t(A) := q_1$ ("tail": queue, source, origine)
- $h(A) := q_2$ ("head" tête, but, extrémité)
- $l(A) := a$ ("label" étiquette)
- $w(A) := \alpha$ ("weight" poids).

Poids d'un chemin $\mathcal{A}(w)$. —

Un *chemin* est classiquement une suite d'arêtes consécutives, formellement, une suite $c = A_1 A_2 \cdots A_n$ (c'est un mot en les arêtes et sa longueur est n) telle que $h(A_k) = t(A_{k+1})$ pour $1 \leq k \leq n-1$ pour un tel chemin $t(c) = t(A_1)$, $h(c) = h(A_n)$, $l(c) = l(A_1)l(A_2)\cdots l(A_n)$ (concaténation), $w(c) = w(A_1)w(A_2)\cdots w(A_n)$

6. Nous verrons plus bas que les axiomes de la structure de semi-anneau sont contraints par la définition même du système de transitions ainsi obtenu.

(produit dans le semi-anneau).
Par exemple pour le chemin de longueur 3 suivant ($k = \mathbb{N}$),

$$u = p \overset{a|2}{\to} q \overset{b|3}{\to} r \overset{c|5}{\to} s \tag{68}$$

on a $t(u) = p$, $h(u) = s$, $l(u) = abc$, $h(u) = 30$.
Le poids d'un ensemble de chemins de même source, but et étiquette sera la somme des poids des chemins de cet ensemble. Ainsi, si

$$\mathbf{q}1 \quad \begin{matrix} \overset{u|\alpha}{\to} \\ \overset{u|\beta}{\to} \end{matrix} \quad \mathbf{q}2 \tag{69}$$

le poids de cet ensemble de chemins est $\alpha + \beta$. On a donc que les poids se multiplient en série et s'aditionnent en parallèle. Les diagrammes suivants montrent la nécessité des axiomes de semi-anneau.

Diagramme	Identité	Nom
$p \begin{matrix} \overset{a|\alpha}{\to} \\ \overset{a|\beta}{\to} \\ \overset{a|\gamma}{\to} \end{matrix} q$	$\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$	Associativité de $+$
$p \begin{matrix} \overset{a|\alpha}{\to} \\ \overset{a|\beta}{\to} \end{matrix} q$	$\alpha + \beta = \beta + \alpha$	Commutativité de $+$
$p \begin{matrix} \overset{a|\alpha}{\to} \\ \overset{a|0}{\to} \end{matrix} q$	$\alpha + 0 = \alpha$	Élément neutre (droite) de $+$
$p \overset{a|0}{\to} q \overset{b|\beta}{\to} r$	$0 + \beta = \alpha$	Élément neutre (gauche) de $+$
$p \overset{a|\alpha}{\to} q \overset{b|\beta}{\to} r \overset{c|\gamma}{\to} s$	$\alpha(\beta\gamma) = (\alpha\beta)\gamma$	Associativité de $\times$
$p \begin{matrix} \overset{a|\alpha}{\to} \\ \overset{a|\beta}{\to} \end{matrix} q \overset{b|\gamma}{\to} r$	$(\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma$	Distributivité (droite) de $\times$ sur $+$
$p \overset{a|\alpha}{\to} q \begin{matrix} \overset{b|\beta}{\to} \\ \overset{b|\gamma}{\to} \end{matrix} r$	$\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$	Distributivité (gauche) de $\times$ sur $+$
$p \overset{a|\alpha}{\to} q \overset{b|1_k}{\to} r$	$\alpha \times 1_k = \alpha$	Élément neutre (droite) de $\times$
$p \overset{a|1_k}{\to} q \overset{b|\beta}{\to} r$	$1_k \times \beta = \beta$	Élément neutre (gauche) de $\times$

10.3.2 Structure et comportement des automates

Un automate à poids ou pondéré ("automaton with weights") est la donnée de trois éléments vectoriels (I,M,T):

$$\begin{cases} \bullet \text{ Un vecteur d'entrée } I \in k^{1\times Q} \\ \bullet \text{ Une famille (indexée à A) de matrices de transition } M : A \to k^{Q\times Q} \\ \bullet \text{ Un vecteur de sortie } T \in k^{Q\times 1} \end{cases}$$

où Q est un ensemble fini.

La donnée des transitions (M) est équivalente à celle d'un graphe pondéré dont les sommets sont Q, l'alphabet A et les poids sont pris dans k. Pour deux états q_1,q_2 et une lettre, on construit a flêche du graphe pondéré de l'automate par $q_1 \overset{M(a)_{q_1 q_2}}{\to} q_2$.
De plus, les vecteurs I (resp. T) correspondent à la donnée de flêches rentrantes (resp. sortantes) marquées avec des poids. Dans tout ce processus, on peut ne pas indiquer les flêches de poids nul.
Ce type d'automates, qui généralise les automates (booléens) de la théorie des langages (que l'on obtient alors pour $k = \mathbb{B}$) est une machine qui prend un mot en entrée et retourne un coefficient (dans k) en sortie. Son comportement est donc une fonction $\mathcal{A} : A^* \to k$ (que l'on peut noter, de manière équivalente, comme une série $\mathcal{A} = \sum_{w\in A^*} \mathcal{A}(w)w$). Le comportement se calcule de la façon suivante

Définition 10.7 *Soit, pour deux états r,s et un mot $w \in A^*$, on définit le comportement local entre les états r et s, $\mathcal{A}^{r,s}(w)$ comme la somme des poids des chemins d'étiquette w qui joignent r à s. Soit*

$$\mathcal{A}^{r,s}(w) := I_r \left(\sum_{\substack{c,\ \text{chemin}\ l(c)=w \\ t(c)=r,\ h(c)=s}} weight(c) \right) T_s \tag{70}$$

Le comportement de $\mathcal{A}$ se définit par les formules

$$\mathcal{A}(w) = \sum_{r,s\in Q} \mathcal{A}^{r,s}(w) \tag{71}$$

$$behaviour(\mathcal{A}) = \sum_{w\in A^*} \mathcal{A}(w)w \tag{72}$$

La définition (10.7) a le sens intuitif suivant :

- Pour le comportement "local" ($\mathcal{A}^{r,s}(w)$) entre deux états donnés r,s selon le mot w, l'équation (89) donne le poids selon la règle suivante :
  1. on fait le bilan parallèle (c'est à dire une somme) des poids des chemins de même étiquette w qui joignent r à s
  2. on multiplie (à gauche) par le poids d'entrée en r
  3. on multiplie (à droite) par le poids de sortie en s
- Pour le comportement "global" ($\mathcal{A}(w)$) selon le mot w, on fait la somme de tous les comportements locaux pour tous les couples d'états

Calcul matriciel du poids $\mathcal{A}(w)$. —

On étend d'abord la fonction de transition M à A^* par

$$M(\epsilon) = I_{Q\times Q},\ M(w) = M(a_1a_2\cdots a_n) = M(a_1)M(a_2\cdots M(a_n) \tag{73}$$

où $I_{Q\times Q}$ est la matrice identité de format $Q\times Q$. Le lien avec le graphe de l'automate est donné par la proposition suivante :

Proposition 10.8

$$\mathcal{A}(w) := IM(w)T \tag{74}$$

d'après la règle de multiplication des matrices, on a bien que $IM(w)T$ est une matrice de format 1×1 et donc un élément de k.

10.3.3 Premiers automates

1. Longueur totale $\sum_{w\in A^*} |w|w$
2. Comptage des a, $\sum_{w\in A^*} |w|_a w$ et des b, $\sum_{w\in A^*} |w|_b w$
3. Produit des degrés partiels $\sum_{w\in A^*} |w|_a|w|_b w$
4. Autres produits $\sum_{w\in A^*} F_{|w|}|w|w$, $\sum_{w\in A^*} F_{|w|_a}|w|_b w$

10.4 Le décalage

10.4.1 Formules liant le décalage et les autres opérateurs

- Le décalage est linéaire ($\sigma(S+T) = \sigma(S)+\sigma(T);\ \sigma(\alpha S) = \alpha\sigma(S);\ \sigma(S\alpha) = \sigma(S)\alpha$)
- $\sigma(S\odot T) = \sigma(S)\odot\sigma(T)$
- $\sigma(ST) = \sigma(S)T + S(0)\sigma(T)$
- $\sigma(S^*) = \sigma(S)S^*$

10.4.2 Espaces stables et matrice de décalage

Définition 10.9 *i) On dit qu'une famille finie $\mathcal{F} = (f_r)_{r\in Q}$ est stable par décalage si l'ensemble des ses combinaisons linéaires $lin(\mathcal{F})$ l'est. De manière équivalente, pour tout $r\in Q$*

$$\sigma(f_r) = \sum_{s\in Q} \alpha_{r,s} f_s$$

la matrice $(\alpha_{r,s})$ de format $Q\times Q$ s'appelle la matrice du décalage par rapport à la famille $\mathcal{F}$. On dit alors que l'espace engendré par $\mathcal{F}$ est un espace stable de type fini.

ii) La familler des décalées $\sigma^n(f)$ d'une série f engendre un espace de type fini ssi celle-ci admet une relation de récurrence linéaire.

$$\sigma^k(f) = \sum_{j=0}^{n} c_j \sigma^j(f)$$

On appelle alors matrice du décalage pour f, la matrice précédente pour la famille $(f,\sigma(f),\cdots\sigma^{k-1}(f)$.

Exercice 10.10 *I) Écrire la matrice du décalage pour les séries suivantes*

$$\begin{array}{lll} \sum_{n\geq 0} F_n z^n & \sum_{n\geq 0} n z^n & \sum_{n\geq 0} n^2 z^n \\ \sum_{n\geq 0} n F_n z^n & \sum_{n\geq 0} (F_n)^2 z^n & \sum_{n\geq 0} z^{2n+1} \end{array}$$

Indication
On itère σ sur la série S jusqu'à ce que l'on trouve une récurrence linéaire. Ou bien, on identifie des séries $[S_1,S_2,\cdots S_n]$ stables par décalage soit

$$\sigma(s_i) = \sum_j \alpha_{ij} S_j$$

II) (Formule de Dobinsky).
1) On considère les opérateurs sur les séries notés z et $\frac{d}{dz}$ définis comme sur les développemùents de Taylor.
a) Montrer que $\frac{d}{dz} z^k = k z^{k-1} + z^k \frac{d}{dz}$.
b) En déduire que $(z\frac{d}{dz})^n = \sum_{k=1}^{n} \alpha(n,k) z^k (\frac{d}{dz})^k$.
II) Ici $\mathbb{K} = \mathbb{C}$. Le but de cet exercice est de montrer qu'on peut définir des puissances généralisées pour les séries de terme constant 1.
On définit, pour $(\alpha,k) \in \mathbb{C} \times \mathbb{N}$ le coefficient binomial généralisé par

$$\binom{\alpha}{k} = \frac{\alpha(\alpha-1)\cdots(\alpha-k+1)}{k!}$$

Soit une série S de terme constant 0. On définit la série $(1+S)^\alpha$ par

$$(1+S)^\alpha = \sum_{n\geq 0} \binom{\alpha}{k} S^k$$

(cf remarque 3.6 (iv)).
1) Montrer que, si $\alpha \in \mathbb{Z}$, cette notion de puissance correspond bien à la notion usuelle.
2) Étendre cette vérification aux puissances fractionnaires $\alpha \in \mathbb{Q}$.
3) (**) Comment étendre cette vérification aux complexes algébriques sur $\mathbb{Q}$?
III) 1) Trouver, s'ils existent, le premier et le 10^{ième} n tel que: (on appliquera les séries)

a) $3^n \equiv 5^{2n}\ [7]$	*b)* $n3^{7n+1} \equiv 2^{3n+1}\ [7]$	*c)* $5n3^n \equiv 2n5^{2n}\ [7]$
d) $3^n \equiv 7^{2n}\ [11]$	*e)* $3^{7n+1} \equiv 7^{3n+1}\ [11]$	*f)* $5n3^n \equiv 2n7^{2n}\ [11]$
g) $3^n \equiv n^2+1\ [11]$	*h)* $n \equiv 5^n\ [11]$	*i)* $n \equiv 2^{2n}\ [11]$

2) Donner deux autres méthodes pour trouver la récurrence linéaire.
3) Et le 100^{ième} ? le $m^{ième}$?
IV) Exercices sur les séries génératrices vues en cours (fontaines de pièces etc..)

10.5 Types courants (de séries)

Pour pouvoir faire des opérations sur les séries, il faut supposer que l'on sait opérer sur les coefficients. Pour simplifier notre approche, nous supposerons que les coefficients sont réels ou complexes (i.e. $k = \mathbb{R}$ ou $\mathbb{C}$)[7].

7. En fait une telle restriction est inutile en pratique et l'espace des coefficients peut être restreint aux anneaux et même - avec encore plus de succès en Informatique - aux semi-anneaux qui sont des structures qui vérifient les axiomes des anneaux sauf l'existence d'un opposé pour tout élément

D'autre part, on se limitera aux séries pour lesquelles l'ensemble des monômes M est stable pour une certaine opération associative et comporte un élément neutre (qui servira de base aux termes constants).

Définition 10.11 *i) On dit que $(M,*)$ est un monoïde si M est muni d'une loi associative admettant un élément neutre.*
ii) Soit k un ensemble semi-anneau (de coefficients) et $(M,*)$ un monoïde. On appelle série sur M à coefficients dans k, toute fonction $M \overset{coeff}{\mapsto} k$.

Pour les semi-anneaux, on consultera l'annexe.
Les séries de toutes sortes sont des fonctions (avec ou sans restrictions) sur les monoïdes constitués par les monômes. Voyons quelques exemples.

Séries	Monômes	Restrictions	Remarques
Univariées en z	$z^{\mathbb{N}} = \{z^n\}_{n\in\mathbb{N}}$	Aucune	Espace noté $k[[z]]$
Polynômes en z	$z^{\mathbb{N}} = \{z^n\}_{n\in\mathbb{N}}$	Support fini	Espace noté $k[z]$
Plusieurs variables commutatives ($\mathbb{X}$)	$\mathbb{N}^{(\mathbb{X})}$, fonctions $X \mapsto \mathbb{N}$ à support fini	Aucune	Espace noté $k[[\mathbb{X}]]$
Polynômes à plusieurs variables commutatives ($\mathbb{X}$)	$\mathbb{N}^{(\mathbb{X})}$, fonctions $X \mapsto \mathbb{N}$ à support fini	Support fini	Espace noté $k[\mathbb{X}]$
Plusieurs variables noncommutatives ($\mathbb{X}$)	$\mathbb{X}^*$, monoïde libre sur l'alphabet $\mathbb{X}$	Aucune	Espace noté $k\langle\langle\mathbb{X}\rangle\rangle$
Polynômes à plusieurs variables noncommutatives	$\mathbb{X}^*$, monoïde libre sur l'alphabet $\mathbb{X}$	Support fini	Espace noté $k\langle\mathbb{X}\rangle$
Séries de Laurent	$z^{\mathbb{Z}} = \{z^n\}_{n\in\mathbb{Z}}$	$n \geq N;\ N \in \mathbb{Z}$	Espace noté $k((z))$
Polynômes de Laurent	$z^{\mathbb{Z}} = \{z^n\}_{n\in\mathbb{Z}}$	Support fini	Espace noté $k(z,z^{-1})$
Séries de Puiseux	$z^{\mathbb{Q}_+} = \{z^\alpha\}_{\substack{\alpha\in\mathbb{Q}\\ \alpha>0}}$	Aucune	
Séries de Malcev	$(\Gamma, \prec)$ groupe totalement ordonné	Support bien ordonné	Espace noté $k((\Gamma))$
Séries de Taylor	$z^{\mathbb{N}} = \{z^n\}_{n\in\mathbb{N}}$	Rayon de convergence	Espace noté $k[[\{z\}]]$ $k = \mathbb{R}$ ou $\mathbb{C}$
Exponentielles	$\{e^{nz}\}_{n\in\mathbb{N}}$		
Dirichlet	$\{n^{-z}\}_{n>0 \text{ entier}}$		
Bertrand	$e^{\alpha z} ln(z)^\beta n^\gamma$		

10.5.1 Composition des automates

Somme et multiplication par un coefficient constant

Produit de Hadamard

Produit (de concaténation)

Nous avons vu que nous pouvions coder de "l'infini dans du fini" en considérant les suites ultimement périodiques que sont les développements illimités des rationnels. Nous allons voir qu'il en est de même pour la production des automates finis, en effet, un automate fini, dès qu'il possède un chemin réussi qui comporte un boucle, reconnaît un langage infini.

Exercice 10.12 *Montrer que cette condition est suffisante, autrement dit, si aucun chemin réussi ne comporte de boucle, alors le langage reconnu par l'automate est fini.*

Commençons par un exemple : On considère un automate (booléen), d'ensemble d'états Q et dont les transitions sont étiquetées par un alphabet A. Cet automate, via la correspondance (graphes $\leftrightarrow$ matrices) peut être vu comme un triplet (I,T,M) avec :

$$\begin{cases} \bullet \text{ Un vecteur d'entrée } I \in k^{1\times Q} \\ \bullet \text{ Une famille de matrices de transition } M : A \to k^{Q\times Q} \\ \bullet \text{ Un vecteur de sortie } T \in k^{Q\times 1} \end{cases}$$

Dans les automates usuels, les scalaires sont pris dans $\{0,1\}$. Si on considère ces nombres comme des entiers naturels, l'opération $w \to IM(w)T$ donne le nombre de chemins réussis. Une expression rationnelle du comportement de l'auromate (tenant compte des multiplicités) résulte du calcul suivant

$$\sum_{w\in\Sigma^*}(IM(w)T)w = I(\sum_{w\in\Sigma^*}M(w)w)T = I(Id_n - \sum_{a\in\Sigma}M(a)a)^{-1}T$$

si on note $M_\Sigma = \sum_{a\in\Sigma}M(a)a$, on a $M_\Sigma^* = (Id_n - \sum_{a\in\Sigma}M(a)a)^{-1}$. C'est la matrice dont l'entrée d'adresse (i,j) est la somme

$$\sum_{\substack{w \text{ étiquette}\\ \text{un chemin de } i \text{ vers } j}} (\text{nb de chemins } i \to j \text{ d'étiquette } w)w$$

par exemple la matrice

$$M_\Sigma = \begin{pmatrix} a & a \\ b & 0 \end{pmatrix}$$

a pour étoile

$$M_\Sigma^* = \begin{pmatrix} (a+ab)^* & (a+ab)^*a \\ b(a+ab)^* & (ba^*a)^* \end{pmatrix}$$

il est facile de voir que les séries associées sont sans multiplicité (i.e. pour (i,j) et w donnés il existe au plus un chemin d'étiquette w), mais ce n'est pas le cas pour

$$Q_\Sigma = \begin{pmatrix} a & a \\ b & a \end{pmatrix}$$

qui a pour étoile

$$Q_\Sigma^* = \begin{pmatrix} (a+aa^*b)^* & (a+aa^*b)a^*a \\ a^*b(a+aa^*b)^* & (a+ba^*a)^* \end{pmatrix}$$

Exercice 10.13 *1) Dessiner les automates (sans vecteurs d'entrée et sortie) associés aux matrices M_Σ,Q_Σ.*
2) a) Montrer, en utilisant un raisonnement sur les chemins dans un graphe étiqueté convenable, que pour deux lettres, on a $(a+b)^* = (a^*b)a^*$ (élimination de Lazard monoïdale).
b) Appliquer cette identité pour trouver une autre forme de $(a+aa^*b)^*$.
c) Montrer que $a^*aa^* = a\frac{1}{(1-a)^2} = \sum_{n\geq 1}na^n$.
d) Si un mot ne se termine pas par b, sa multiplicité dans $(a^*aa^*b)^*$ est nulle, mais s'il s'écrit $w = a^{n_1}ba^{n_2}b\cdots a^{n_k}b$, on a $(w,(a^*aa^*b)^*) = n_1+n_2+\cdots n_k$. En déduire le développement (i.e. les multiplicités des mots) de $(a^*aa^*b)^*a^*$, puis des 4 coefficients de la matrice Q_Σ^*.
3) a) Soit l'alphabet à quatre lettres $\Sigma = \{a_{11},a_{12},a_{21},a_{22}\}$, montrer directement en raisonnant sur les chemins, que si $G = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ on a

$$G^* = \begin{pmatrix} A_{11} & A_{11}a_{12}a_{22}^* \\ a_{22}^*a_{21}A_{11} & A_{22} \end{pmatrix};\ avec A_{11} = (a_{11}+a_{12}a_{22}^*a_{21})^*,\ A_{22} = (a_{22}+a_{21}a_{11}^*a_{12})^*$$

b) Expliquer en quoi ces formules fournissent un algorithme permettant de calculer l'étoile de toute matrice de séries propres.

Exemple 10.14 *Soit L_n le langage fini formé des mots w tels que $|w|_a + 2|w|_b = n$.*
a) Écrire les premiers termes L_0,L_1,$L_2\cdots$.
b) Calculer $|L_n|$ à l'aide d'une récurrence simple.
c) Montrer que $SG = \sum_n |L_n|t^n = (t+t^2)^* = \frac{1}{1-t-t^2}$.
d) Faire le lien avec le nombre de pavages d'un rectangle $2\times n$ par des dominos 2×1 ([11] pp 321) comment coder les pavages, les énumérer, les générer.
e) À l'aide des décalages, former l'automate qui reconnait la série S.

On a un analogue parfait de ce qui se passe pour les rationnels positifs. Plus précisément :

Exercice 10.15 *A) On considère les arbres $1-2$ qui sont les arbres à 1 ou deux fils.*
À chaque arbre $1-2$, dont les noeuds internes sont signés par "$+$" s'ils ont deux fils et "$()^{-1}$" s'ils en ont un on fait correspondre une fraction (i.e. son évaluation avec les feuilles en 1) donnée par la règle récursive

$$ev(\bullet) = 1;\ ev((\mathcal{A}_1,\mathcal{A}_2)) = ev((\mathcal{A}_1) + ev(\mathcal{A}_2);\ ev((\mathcal{A})) = \frac{1}{ev(\mathcal{A})}$$

montrer que l'ensemble des valeurs obtenues est $\mathbb{Q}_+^*$. Est-ce que la représentation est unique ? Est-ce qu'elle englobe les fractions continues ? Comment caractériser les arbres qui les donnent ?
B) On considère les séries sur un alphabet A (i.e. fonctions $A^* \to k$ où k est un semi-anneau (i.e. suffisant pour faire la calcul matriciel).
a) Montrer que les conditions suivantes sont équivalentes :
i) La série S est l'évaluation d'une expression rationnelle.
ii) La série S est combinaison linéaire d'un ensemble de séries S_1,S_2,$\cdots S_n$ qui est (linéairement) stable par décalages soit

$$(\forall x \in A)(\forall i \in [1..n])(x^{-1}S_i = \sum_{0 \le j \le n} \mu_{i,j}(x)S_j$$

iii) Il existe $\lambda \in K^{1\times n}$, $\mu : A \to K^{n\times n}$,$\gamma \in K^{1\times n}$ tels que pour tout $w \in A^*$, $(S,w) = \lambda\mu(w)\gamma$ (où $\mu()$ dénote encore l'extention de μ à A^*).

Lorsque l'on a une partie $X \in A^*$, on peut se demander :
Quel est le langage $L(X)$ engendré par X ?
c'est à dire les suites finies d'instructions (i.e. le sous-monoïde engendré). On a $L(X) = \sum_{n\ge 0} X^n$ à coefficients dans $\mathbb{B}$. La même somme à coefficients dans $\mathbb{N}$ contient plus d'informations (soit le nombre de façons d'obtenir w comme produit de facteurs dans X).
Automates. Automates à multiplicité (notion de coût). Comportement d'un automate. Séries (exemples), rationnelles.
Passage SGO $\leftrightarrow$ Aut $\leftrightarrow$ Exp. rat.
Exemples de $\mathbb{N}$ et $\mathbb{Z}$ automates.

10.6 Modules

Soit $(M,+)$, un monoïde commutatif. Une structure de K-module à gauche sur est définie par la donnée de ϕ : $K \mapsto End(M)$ avec la condition :
MG) ϕ un morphisme.
En général, on note αx, lélément $\phi(\alpha)(x)$, ce qui définit, de façon équivalente, une loi externe $K \times M \mapsto M$. La condition (MG) est alors équivalente aux axiomes suivants

$$(\forall \alpha,\beta \in K)(\forall x,y \in M)$$

1. $\alpha(x+y) = \alpha x + \alpha y;\ \alpha 0_M = 0_M$
2. $(\alpha+\beta)x = \alpha x + \beta x;\ 0_K x = 0_M$
3. $\alpha(\beta x) = (\alpha\beta)x;\ 1_K x = x$

De façon duale, une structure de module à droite sur K est définie par la donnée d'un morphisme ϕ : $K^\circ \mapsto End(M)$, pour rendre naturels les axiomes, on crée alors la loi externe $M \times K \mapsto M$ définie par $(x,\alpha) \mapsto \phi(\alpha,x)$. Les axiomes qui en résultent s'écrivent sans difficulté.

Les notions de morphisme de module et de sous-module et de bi-module se définissent comme en algèbre linéaire standard.

10.6.1 Le bi-module des fonctions $X \mapsto K$

Soit X, un ensemble et K, un semi-anneau non trivial ($0_k \neq 1_K$ (10.6)). On appelle K-sou ensemble de X (ou ensemble avec multiplicités dans K) toute application $X \stackrel{f}{\mapsto} K$.
Sommabilité d'une famille de fonctions. Notation sommatoire.

10.7 Cas où X est un monoïde

exemples et fonctions sur les monoïdes, fonctions caractéristiques

10.7.1 Décalages

Th de Abe

10.7.2 Convolution

Produit de convolution : sommabilité des $\langle S|u\rangle\langle T|v\rangle uv$ et condition (D) (Bourbaki)

10.7.3 Monoïdes localement finis

Sommabilité de $(M - \{e_M\})^n$ (Eilenberg) et fonction de Möbius. Exemples.

10.8 Cas du monoïde libre, théorème de Kleene-Schützenberger

Clôture rationnelle.

11 La structure d'AF

11.1 Définition et fonctionnement

Définition, qualités d'un automate (déterministe, complet). Compléter un automate non complet grâce à un état puits.

11.2 Graphe et représentations linéaires

Graphe d'un automate, origine et extrémité d'une transition (fonctions h "head" et t "tail"), chemins et prolongement de h et t aux chemins, chemin réussi, lagage reconnu par un automate.
Représentation linéaire : entrées, sorties, matrices de transition.

12 Calcul du langage reconnu par un AF

12.1 Automates finis et structures de transition

Rappelons qu'un **AF** (automate fini est une machine définie par un 5-uplet $\mathcal{A} = (Q,A,\bullet\ ,I,F)$ où

- Q est un ensemble fini d'états
- A est un alphabet fini
- $\bullet : Q \times A \mapsto \mathcal{P}(Q)$ est une application
- $I \subseteq Q$ est l'ensemble des entrées (i.e. états initiaux)
- $F \subseteq Q$ est l'ensemble des sorties (i.e. états finaux)

Note 12.1 *Si on oublie les états initiaux et finaux, on a une* structure de transition.
On appelle donc structure de transition *la donnée d'un triplet $\mathcal{A} = (Q,A,\bullet\ )$ avec les caractéristiques du paragraphe précédent.*

D'autre part, on va considérer des matrices (carrées) de langages. Le produit se fait comme d'habitude : si $U = (L_{ij})_{1\leq i,j\leq n}$ et $V = (M_{ij})_{1\leq i,j\leq n}$ on a $UV = (N_{ij})_{1\leq i,j\leq n}$ avec

$$N_{ij} := \sum_{1\leq k\leq n} M_{ik}N_{kj} \tag{75}$$

Rappel. — Dans $U = (L_{ij})_{1\leq i,j\leq n}$, on a i qui est l'adresse de ligne, j l'adresse de colonne. Ainsi pour une matrice 3×3 on a

$$\begin{pmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \tag{76}$$

Définition 12.2 *La* matrice de transtition "lettres" *(ou* matrice-lettres *tout court) d'une structure de transition est une matrice de langages (qui sont des sous-alphabets). C'est la matrice de format $Q \times Q$ (ses lignes et ses colonnes sont indexées par Q)*

$$T := \left(\sum_{q_2 \in q_1.a} a \right)_{q_1, q_2 \in Q} \tag{77}$$

étant entendu (par convention générale) que la somme est nulle s'il n'y pas d'arête entre q_1 et q_2.

12.2 Étoile d'une matrice-lettres

Comme (dans un premier temps), on fait du calcul boooléen (i. e. union et concaténation), on a sur les fonctions caractéristiques (voir [1] Ch 4 par 4 num. 9)

$$\phi_{A \cup B} = \phi_A + \phi_B - \phi_A \phi_B \tag{78}$$

et,

$$\phi_{A.B}(w) = \begin{cases} 1 & \textit{s'il existe une factorisation } w = uv \textit{ telle que } \phi_A(u)\phi_B(v) = 1 \\ 0 & \textit{sinon} \end{cases} \tag{79}$$

On adopte donc les nouvelles opérations (booléennes) suivantes

$+_{TL}$	0	1
0	0	1
1	1	1

$\times_{TL}$	0	1
0	0	0
1	0	1

On remarquera que ces opérations sont données par les formules

$$X +_{TL} Y = X + Y - XY \; ; \; X \times_{TL} Y = XY \tag{80}$$

c'est à dire qu'on a le produit ordinaire, mais la "nouvelle" somme est donnée conformément à la formule (79).

Les puissances de la matrice-lettres d'une structure de transition ont une propriété remarquable.

Proposition 12.3 *Soit T, la matrice-lettres d'une structure de transition $\mathcal{T} = (Q, A, \bullet)$. Pour toute paire d'états (q_1, q_2), on a*

$$(T^k)[q_1, q_2] = \{w \in A^k | q_1.w = q_2\} \tag{81}$$

autrement dit, l'entrée d'adresse $[q_1, q_2]$ de la puissance T^k est l'ensemble des mots de longueur k qui font passer de q_1 à q_2.

13 Fonctions sur les mots

Dans ce chapitre, nous nous intéresserons plus spécialement aux fonctions sur A^* (A est un alphabet). C'est à dire, en ce qui concerne les sysèmes (informatiques ou électroniques) aux machines qui acceptent un mot en entrée et retournent un coefficient (un scalaire: un nombre, un booléen, un réel, un complexe ou même une matrice) en sortie.

$$w \rightarrow \boxed{\text{MACHINE}} \rightarrow M(w) \tag{82}$$

On appellera *comportement* de la machine cette fonction $A^* \rightarrow K$. L'ensemble K est celui des scalaires (ce sont les nombres qui sont en entrée des matrices), dans ce cours d'introduction, on considère soit les Boooléens $\mathbb{B}$ (avec la règle $1 + 1 = 1$) soit les naturels $\mathbb{N}$ (avec la règle $1 + 1 = 2$) et deux machines seront dites équivalentes ssi elles définisent la même fonction.

On peut composer les machines à l'aide des fonctions classiques (additionneurs, multiplieurs).

$$w \rightarrow \begin{array}{c} \nearrow \boxed{\text{MACHINE 1}} \searrow \\ \rightarrow \boxed{\text{MACHINE 2}} \rightarrow \end{array} \bigoplus \rightarrow M1(w) + M2(w)$$

$$w \rightarrow \begin{array}{c} \nearrow \boxed{\text{MACHINE 1}} \searrow \\ \rightarrow \boxed{\text{MACHINE 2}} \rightarrow \end{array} \bigotimes \rightarrow M1(w) \times M2(w)$$

c'est le produit (ou la somme) ponctuel(le) des fonctions correspondantes.
Il y a aussi un autre type de produit qui est très utile (nous verrons qu'il généralise la concaténation et qu'il éclaire les opérations sur les parties), c'est le produit défini par le formule

$$M1 * M2(w) = \sum_{uv=w} M1(u)M2(v) \tag{83}$$

il peut être réalisé par le système suivant

$$\begin{array}{cccccc} & \nearrow & u \to & \boxed{\text{MACHINE 1}} & \searrow & \\ w = uv & \to & v \to & \boxed{\text{MACHINE 2}} & \to \otimes & \bigoplus_{uv=w} M1(u)M2(v) \end{array}$$

13.1 Quelques exemples de machines

Toutes les machines considérées ici acceptent des mots en entrée et retournent des coefficients en sortie. Tout d'abord quelques machines de type "compteur".

Exemple 1: LONGUEUR D'UN MOT. — C'est une machine (ou un programme) qui lit un mot de gauche à droite (ou de droite à gauche peu importe ici) et qui incémente un compteur de +1 à chaque lettre (le compteur est initialisé à 0).
Le résultat est la longueur du mot.
Lorsque le mot est vide le compteur rest bien à son initialisation et le résultat est 0 (longueur du mot vide).

Exemple 2: NOMBRES D'OCCURENCES D'UNE LETTRE. — Soit $A = \{a,b\}$. C'est une machine (ou un programme) qui lit un mot de gauche à droite (ou de droite à gauche peu importe ici) et qui incémente un compteur de +1 à chaque lecture de a (le compteur est initialisé à 0).
Le résultat est le nombre d'occurences de a.
Lorsque le mot est vide le compteur est bien à son initialisation et le résultat est 0 (longueur du mot vide).

Cette machine peut se réaliser matricellement. On pose

$$M(a) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix};\ M(b) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix};\ I = \begin{pmatrix} 1 & 0 \end{pmatrix};\ T = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{84}$$

puis $M(w) = M(a_1a_2\cdots a_n) = M(a_1)M(a_2)\cdots M(a_n)$ le résultat de la lecture d'un mot w est $IM(w)T$.
Montrons que cette machine compte bien le degré partiel en a

Exercice 13.1 *Montrer que cette machine compte bien le degré partiel en a.*

Dans la suite, une fonction $f : A^* \to K$ pourra aussi être noée $\sum_{u\in A^*} f(u)u$ (notation sommatoire). Cette notation permet de manipuler les fonctions comme des séries et les parties comme des sommes de mots.

14 Systèmes et Calcul

14.1 Introduction

Exemples d'automates booléens, stochastiques, de comptage, de plus courts chemins. Les semi-anneaux associés sont : $\mathbb{B}$, $\mathbb{R}_+$,$\mathbb{N}$,$([0,+\infty],min,+)$.

14.2 Description de la structure d'automate

14.2.1 Graphe pondéré

L'élément de base de ces graphes est la flêche $A = q_1 \stackrel{a|\alpha}{\to} q_2$ avec $q_i \in Q$, $a \in \Sigma$, $\alpha \in k$ où Q est un ensemble d'états, Σ un alphabet et k, un semi-anneau[8]. Pour un tel objet, on définit, selon les conventions générales de la théorie des graphes,

– $t(A) := q_1$ ("tail": queue, source, origine)

8. Nous verrons plus bas que les axiomes de la structure de semi-anneau sont contraints par la définition même du système de transitions ainsi obtenu.

- $h(A) := q_2$ (“head” tête, but, extrémité)
- $l(A) := a$ (“label” étiquette)
- $w(A) := \alpha$ (“weight” poids).

Un *chemin* est une suite d'arêtes $c = A_1 A_2 \cdots A_n$ (c'est un mot en les arêtes et sa longueur est n) telle que $h(A_k) = t(A_{k+1})$ pour $1 \leq k \leq n-1$ pour un tel chemin $t(c) = t(A_1), h(c) = h(A_n), l(c) = l(A_1)l(A_2)\cdots l(A_n)$ (concaténation), $w(c) = w(A_1)w(A_2)\cdots w(A_n)$ (produit dans le semi-anneau).
Par exemple pour le chemin de longueur 3 suivant ($k = \mathbb{N}$),

$$u = p \xrightarrow{a|2} q \xrightarrow{b|3} r \xrightarrow{c|5} s \tag{85}$$

on a $t(u) = p$, $h(u) = s$, $l(u) = abc$, $h(u) = 30$.

Le poids d'un ensemble de chemins de même source, but et étiquette est la somme des poids des chemins de cet ensemble. Ainsi, si

$$\mathbf{q}1 \quad \begin{array}{c} \xrightarrow{u|\alpha} \\ \xrightarrow{u|\beta} \end{array} \quad \mathbf{q}2 \tag{86}$$

le poids de cet ensemble de chemins est $\alpha + \beta$. On a donc que les poids se multiplient en série et s'aditionnent en parallèle. Les diagrammes suivants montrent la nécessité des axiomes de semi-anneau.

Diagramme	Identité	Nom
$p \begin{array}{c} \xrightarrow{a\|\alpha} \\ \xrightarrow{a\|\beta} \\ \xrightarrow{a\|\gamma} \end{array} q$	$\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$	Associativité de $+$
$p \begin{array}{c} \xrightarrow{a\|\alpha} \\ \xrightarrow{a\|\beta} \end{array} q$	$\alpha + \beta = \beta + \alpha$	Commutativité de $+$
$p \begin{array}{c} \xrightarrow{a\|\alpha} \\ \xrightarrow{a\|0} \end{array} q$	$\alpha + 0 = \alpha$	Élément neutre (droite) de $+$
$p \xrightarrow{a\|0} q \xrightarrow{b\|\beta} r$	$0 + \beta = \alpha$	Élément neutre (gauche) de $+$
$p \xrightarrow{a\|\alpha} q \xrightarrow{b\|\beta} r \xrightarrow{c\|\gamma} s$	$\alpha(\beta\gamma) = (\alpha\beta)\gamma$	Associativité de $\times$
$p \begin{array}{c} \xrightarrow{a\|\alpha} \\ \xrightarrow{a\|\beta} \end{array} q \xrightarrow{b\|\gamma} r$	$(\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma$	Distributivité (droite) de $\times$ sur $+$
$p \xrightarrow{a\|\alpha} q \begin{array}{c} \xrightarrow{b\|\beta} \\ \xrightarrow{b\|\gamma} \end{array} r$	$\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$	Distributivité (gauche) de $\times$ sur $+$
$p \xrightarrow{a\|\alpha} q \xrightarrow{b\|1_k} r$	$\alpha \times 1_k = \alpha$	Élément neutre (droite) de $\times$
$p \xrightarrow{a\|1_k} q \xrightarrow{b\|\beta} r$	$1_k \times \beta = \beta$	Élément neutre (gauche) de $\times$

14.2.2 Structure et comportement des automates

Un automate à poids ou pondéré (“automaton with weights”) est la donnée de trois éléments vectoriels (I,M,T):

$\left\{ \begin{array}{l} \bullet \text{ Un vecteur d'entrée } I \in k^{1\times Q} \\ \bullet \text{ Une famille (indexée à A) de matrices de transition } M : A \to k^{Q\times Q} \\ \bullet \text{ Un vecteur de sortie } T \in k^{Q\times 1} \end{array} \right.$

La donnée des transitions (M) est équivalente à celle d'un graphe pondéré dont les sommets sont Q, l'alphabet A et les poids sont pris dans k. De plus celle de I (resp. T) correspond à la donnée de flêches rentrantes (resp. sortantes) marquées avec des poids. Dans tout ce processus, on peut ne pas indiquer les flêches de poids nul.
Ce type d'automates généralise les automates (booléens) de la théorie des langages (que l'on obtient alors pour $k = \mathbb{B}$) est une machine qui prend un mot en entrée et retourne un coefficient (dans k) en sortie. Son comportement est donc une fonction $\mathcal{A} : A^* \to k$ (que l'on peut noter, de manière équivalente, comme une série $\mathcal{A} = \sum_{w\in A^*} \mathcal{A}(w)w$).

Calcul du poids $\mathcal{A}(w)$. —
On étend d'abord la fonction de transition M à A^* par

$$M(\epsilon) = I_{Q\times Q},\, M(w) = M(a_1a_2\cdots a_n) = M(a_1)M(a_2\cdots M(a_n) \tag{87}$$

où $I_{Q\times Q}$ est la matrice identité de format $Q \times Q$. Le calcul du poids d'un mot est alors, par définition,

$$\mathcal{A}(w) := IM(w)T \tag{88}$$

d'après la règle de multiplication des matrices, on a bien que $IM(w)T$ est une matrice de format 1×1 et donc un élément de k. Le lien avec le graphe de l'automate est donné par la proposition suivante :

Proposition 14.1 *Soit, pour deux états r,s et un mot $w \in A^*$*

$$\mathcal{A}^{r,s}(w) := I_r \left(\sum_{\substack{\text{c, chemin } l(c)=w \\ t(c)=r,\ h(c)=s}} weight(c) \right) T_s \tag{89}$$

alors

$$\mathcal{A}(w) = \sum_{r,s\in Q} \mathcal{A}^{r,s}(w) \tag{90}$$

Cette proposition a le sens intuitif suivant :

1. l'équation (89) donne le poids calculé comme au paragraphe précédent
   - on fait le bilan parallèle (c'est à dire une somme) des poids des chemins qui joignent r à s
2. on multiplie (à gauche si c'est non commutatif) par le poids d'entrée en r
3. on multiplie (à droite si c'est non commutatif) par le poids de sortie en s

14.2.3 Premiers automates

1. Longueur totale $\sum_{w\in A^*} |w|w$
2. Comptage des a, $\sum_{w\in A^*} |w|_a w$ et des b, $\sum_{w\in A^*} |w|_b w$
3. Produit des degrés partiels $\sum_{w\in A^*} |w|_a|w|_b w$
4. Autres produits $\sum_{w\in A^*} F_{|w|}|w|w$, $\sum_{w\in A^*} F_{|w|_a}|w|_b w$

15 Résumé des cours et TD.

- Opération sur les laangages : somme, somme infinie, concaténation, itération.
- fin des opérations sur les langages : intersection, shuffle, infiltration, décalage. Notation produit scalaire.
- Équations et systèmes d'équations en langages (discussion de l'unicité) p16 du poly. Équation reproductrice, étoile d'une matrice.

- TD : Fin du générateur aléatoire de mots. Test sur des automates à facteurs exclus. Début de la minimisation.
- compléments sur l'étoile d'un matrice - lettre
- solution de $MX + I = M$
- étoile par blocs
- TD : test des automates contre des compteurs, tests aléatoires
- rationalité des solutions, expressions rationnelles
- langages rationnels
- nombre de chemins, factorisations
- langages rationnels
- types d'automates finis (AF) : AFD, AFC, AFDC, rôle des sorties
- TD : implémentation des produits d'automates

15.0.4 Composition des automates

Somme et multiplication par un coefficient constant

Produit de Hadamard

Produit (de concaténation)

Nous avons vu que nous pouvions coder de "l'infini dans du fini" en considérant les suites ultimement périodiques que sont les développements illimités des rationnels. Nous allons voir qu'il en est de même pour la production des automates finis, en effet, un automate fini, dès qu'il possède un chemin réussi qui comporte un boucle, reconnaît un langage infini.

Exercice 15.1 *Montrer que cette condition est suffisante, autrement dit, si aucun chemin réussi ne comporte de boucle, alors le langage reconnu par l'automate est fini.*

Commençons par un exemple : On considère un automate (booléen), d'ensemble d'états Q et dont les transitions sont étiquetées par un alphabet A. Cet automate, via la correspondance (graphes $\leftrightarrow$ matrices) peut être vu comme un triplet (I,T,M) avec :

$$\begin{cases} \bullet & \text{Un vecteur d'entrée } I \in k^{1\times Q} \\ \bullet & \text{Une famille de matrices de transition } M : A \to k^{Q\times Q} \\ \bullet & \text{Un vecteur de sortie } T \in k^{Q\times 1} \end{cases}$$

Dans les automates usuels, les scalaires sont pris dans $\{0,1\}$. Si on considère ces nombres comme des entiers naturels, l'opération $w \to IM(w)T$ donne le nombre de chemins réussis. Une expression rationnelle du comportement de l'automate (tenant compte des multiplicités) résulte du calcul suivant

$$\sum_{w\in\Sigma^*}(IM(w)T)w = I(\sum_{w\in\Sigma^*}M(w)w)T = I(Id_n - \sum_{a\in\Sigma}M(a)a)^{-1}T$$

si on note $M_\Sigma = \sum_{a\in\Sigma}M(a)a$, on a $M_\Sigma^* = (Id_n - \sum_{a\in\Sigma}M(a)a)^{-1}$. C'est la matrice dont l'entrée d'adresse (i,j) est la somme

$$\sum_{\substack{w \text{ étiquette} \\ \text{un chemin de } i \text{ vers } j}} (\text{nb de chemins } i \to j \text{ d'étiquette } w)w$$

par exemple la matrice

$$M_\Sigma = \begin{pmatrix} a & a \\ b & 0 \end{pmatrix}$$

a pour étoile

$$M_\Sigma^* = \begin{pmatrix} (a+ab)^* & (a+ab)^*a \\ b(a+ab)^* & (ba^*a)^* \end{pmatrix}$$

il est facile de voir que les séries associées sont sans multiplicité (i.e. pour (i,j) et w donnés il existe au plus un chemin d'étiquette w), mais ce n'est pas le cas pour

$$Q_\Sigma = \begin{pmatrix} a & a \\ b & a \end{pmatrix}$$

qui a pour étoile

$$Q_\Sigma^* = \begin{pmatrix} (a+aa^*b)^* & (a+aa^*b)a^*a \\ a^*b(a+aa^*b)^* & (a+ba^*a)^* \end{pmatrix}$$

Exercice 15.2 *1) Dessiner les strucures de transition (i. e. automates sans vecteurs d'entrée et sortie) associés aux matrices M_Σ,Q_Σ.*
2) a) Montrer, en utilisant un raisonnement sur les chemins dans un graphe étiqueté convenable, que pour deux lettres, on a $(a+b)^* = (a^*b)a^*$ (élimination de Lazard monoïdale).
b) Appliquer cette identité pour trouver une autre forme de $(a+aa^*b)^*$.
c) Montrer que $a^*aa^* = a\frac{1}{(1-a)^2} = \sum_{n\geq 1}na^n$.
d) Si un mot ne se termine pas par b, sa multiplicité dans $(a^*aa^*b)^*$ est nulle, mais s'il s'écrit $w = a^{n_1}ba^{n_2}b\cdots a^{n_k}b$,

on a $(w,(a^*aa^*b)^*) = n_1+n_2+\cdots n_k$*. En déduire le développement (i.e. les multiplicités des mots) de* $(a^*aa^*b)^*a^*$*, puis des 4 coefficients de la matrice* Q^*_Σ*.*
3) a) Soit l'alphabet à quatre lettres $\Sigma = \{a_{11},a_{12},a_{21},a_{22}\}$*, montrer directement en raisonnant sur les chemins, que si* $G = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ *on a*

$$G^* = \begin{pmatrix} A_{11} & A_{11}a_{12}a_{22}^* \\ a_{22}^*a_{21}A_{11} & A_{22} \end{pmatrix};\ A_{11} = (a_{11} + a_{12}a_{22}^*a_{21})^*,\ A_{22} = (a_{22} + a_{21}a_{11}^*a_{12})^*$$

b) Expliquer en quoi ces formules fournissent un algorithme permettant de calculer l'étoile de toute matrice de séries propres.

Exercice 15.3 *a) Dessinez la structure de transition associée à la matrice-lettre.*

$$\begin{pmatrix} 0 & a & 0 \\ b & 0 & a \\ a+b & 0 & 0 \end{pmatrix}$$

b) Calculez de deux manières l'étoile de cette matrice.

Exercice 15.4 *Soit* L_n *le langage fini formé des mots* w *tels que* $|w|_a + 2|w|_b = n$*.*
a) Écrire les premiers termes $L_0,L_1,L_2\cdots$*.*
b) Calculer $|L_n|$ *à l'aide d'une récurrence simple.*
c) Montrer que $SG = \sum_n |L_n|t^n = (t+t^2)^* = \frac{1}{1-t-t^2}$*.*
d) Faire le lien avec le nombre de pavages d'un rectangle $2\times n$ *par des dominos* 2×1 *comment coder les pavages, les énumérer, les générer.*
e) À l'aide des décalages, former l'automate qui reconnait la série S*.*

On a un analogue parfait de ce qui se passe pour les rationnels positifs. Plus précisément :

Exercice 15.5 *A) On considère les arbres* $1-2$ *qui sont les arbres à* 1 *ou deux fils.*
À chaque arbre $1-2$*, dont les noeuds internes sont signés par "*$+$*" s'ils ont deux fils et "*$()^{-1}$*" s'ils en ont un on fait correspondre une fraction (i.e. son évaluation avec les feuilles en* 1*) donnée par la règle récursive*

$$ev(\bullet) = 1;\ ev((\mathcal{A}_1,\mathcal{A}_2)) = ev((\mathcal{A}_1) + ev(\mathcal{A}_2);\ ev((\mathcal{A})) = \frac{1}{ev(\mathcal{A})}$$

montrer que l'ensemble des valeurs obtenues est $\mathbb{Q}^*_+$*. Est-ce que la représentation est unique ? Est-ce qu'elle englobe les fractions continues ? Comment caractériser les arbres qui les donnent ?*
B) On considère les séries sur un alphabet A *(i.e. fonctions* $A^* \to k$ *où* k *est un semi-anneau (i.e. suffisant pour faire la calcul matriciel).*
a) Montrer que les conditions suivantes sont équivalentes :
i) La série S *est l'évaluation d'une expression rationnelle.*
ii) La série S *est combinaison linéaire d'un ensemble de séries* $S_1,S_2,\cdots S_n$ *qui est (linéairement) stable par décalages soit*

$$(\forall x \in A)(\forall i \in [1..n])(x^{-1}S_i = \sum_{0\le j\le n} \mu_{i,j}(x)S_j$$

iii) Il existe $\lambda \in K^{1\times n}$*,* $\mu : A \to K^{n\times n}$*,*$\gamma \in K^{1\times n}$ *tels que pour tout* $w \in A^*$*,* $(S,w) = \lambda\mu(w)\gamma$ *(où* $\mu()$ *dénote encore l'extention de* μ *à* A^**).*

Lorsque l'on a une partie $X \in A^*$, on peut se demander :
Quel est le langage $L(X)$ engendré par X ?
c'est à dire les suites finies d'instructions (i.e. le sous-monoïde engendré). On a $L(X) = \sum_{n\ge 0} X^n$ à coefficients dans $\mathbb{B}$. La même somme à coefficients dans $\mathbb{N}$ contient plus d'informations (soit le nombre de façons d'obtenir w comme produit de facteurs dans X).
Automates. Automates à multiplicité (notion de coût). Comportement d'un automate. Séries (exemples), rationnelles.
Passage SGO $\leftrightarrow$ Aut $\leftrightarrow$ Exp. rat.
Exemples de $\mathbb{N}$ et $\mathbb{Z}$ automates.
Séries génératrices (rationnelles -arbres de Fibonacci- et non rationnelles -arbres binaires, chemins de Dyck-). Résolution des premières récurrences, décalage et Δ. Complexité du comptage des boucles. Arbres $1-2$.

Références

[1] BOURBAKI N., *Intégration*, Hermann

[2] COHEN H., *A Course in Computational Algebraic Number Theory*. Springer (1993)

[3] CHAR B.W., GEDDES K.O., GONNET G.H., ALI.., *Maple V Library Reference Manual*, Springer (1992).

[4] CHAR B.W., GEDDES K.O., GONNET G.H., ALI.., *Maple V Language Reference Manual*, Springer (1992).

[5] DAVENPORT J., SIRET Y., TOURNIER E., *Calcul formel*, Masson (1986)

[6] DEMAZURE M., Cours d'algèbre : Divisiilité, Primalité, codes. Cassini (1997).

[7] VON ZUR GATHEN J. AND GERAHRD J. Modern Computer Algebra. Cambridge (1999).

[8] KNUTH D., *The art of computer programming* Tome I. Addison-Wesley (1981)

[9] KNUTH D., *The art of computer programming* Tome II. Addison-Wesley (1981)

[10] GRAHAM R., KNUTH D., PATASHNICK O., *Concrete Mathematics* Addison Wesley (1994).

[11] GRAHAM R., KNUTH D., PATASHNICK O., *Mathematiques Concretes* Trad. A. Denise, Vuibert (1999).

[12] NAUDIN P., QUITTÉ C., *Algorithmique Algébrique* Masson (1992)