

Notes du cours de “Structures de Calcul I” MMI 2008-2009.

G. H. E. DUCHAMP*

15-06-2009 04:00

Table des matières

0	Préambule	1
1	Introduction	1
2	Systèmes de Calcul Formel et structures de données	2
2.1	Révision numération	2
2.1.1	Quelques exercices	2
2.2	Les systèmes de Calcul Formel	3
2.2.1	Présentation	3
2.2.2	Exemples de sessions	4
2.2.3	Histoire succincte des Systèmes de Calcul Formel	9
2.3	Introduction à Maple : T.D.	10
2.4	Quelques structures de données	10
2.4.1	Arbres binaires complets	10
2.4.2	Arbres binaires incomplets	12
2.4.3	Arbre 1-2	12
2.4.4	Arité variable	12
2.4.5	Expressions arithmétiques	13
2.4.6	Mots noncommutatifs	13
2.4.7	Mots commutatifs	13
2.4.8	Autres séries	13
3	Les constructeurs set, multiset & list	14
3.1	Définitions générales	14
3.2	Set	14
3.3	List	14
3.4	Multiset	14
3.4.1	OGF	14
3.4.2	EGF	14
3.4.3	La formule exponentielle	14

* Prière de signaler les erreurs à LAURENT POINSOT ou à moi-même.

3.5	Séries rationnelles (représentations linéaires et aspect automatique)	14
3.5.1	Produit de Hadamard	16
4	Systèmes et Calcul	16
4.1	Introduction	16
4.2	Description de la structure d'automate	16
4.2.1	Graphe pondéré	16
4.2.2	Structure et comportement des automates	17
4.2.3	Premiers automates	18
4.2.4	Composition des automates	18
4.3	Séries	21
4.3.1	Exemple : Comportement d'un automate	21
4.3.2	Opérations sur les séries	21
4.3.3	Lien avec les grammaires et les structures de données	21
4.3.4	Énumération	21
4.3.5	Rudiments de Calcul Modulaire	21
5	Annexes	21
5.1	Fonctions génératrices	21
5.2	Une variable	21
5.3	Application au calcul de complexité	21
5.4	Plusieurs variables	21
6	TD et TP (E. Laugerotte & J-P. Dubernard)	22
7	TD	29

0 Préambule

Ce cours est commun aux étudiants originaires des filières Mathématique, Informatique et filières d'Ingénieurs . Il est conçu de façon à permettre aux étudiants des trois sensibilités de pouvoir s'entraîner tant au niveau de la programmation (TP - TD - libre service) qu'au niveau conceptuel : outils de calcul (analytiques et algébriques) de l'informatique théorique &/ou utilisés en informatique moderne (sécurité, simulation en finance, algorithmique rapide). Les T.D. se font en maple et Mupad, mais le contenu de l'enseignement est indépendant du langage.

Le poly correspond à un programme maximal. Seule une partie de celui-ci sera traitée cette année. Les exercices qui sont là pour aider à la compréhension du cours.

Il se peut que vous ne compreniez pas certains énoncés, dans ce cas choisissez que les questions que vous êtes capables d'aborder ... et contactez-nous rapidement pour le décodage (question par oral, email ou wiki).

Ceci vaudra aussi lorsque vous vous exercerez sur annales.

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

1 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 dérivation¹). 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 algorithmes de l'arithmétique élémentaire.

De même que la mathématique (quand elle n'est pas autogène) 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 main³. 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). Ainsi toute l'arithmétique et ses applications⁴ nécessite le calcul exact, en effet une erreur d'un digit même sur un nombre de 250 chiffres peut transformer un nombre pair en nombre impair ! Ceci est gênant 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 données 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..

2 Systèmes de Calcul Formel et structures de données

2.1 Révision numération

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

2.1.1 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 développements 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) $(\underbrace{101 \cdots 101}_{3n \text{ chiffres}})_2$ (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) $a_n = (0, \underbrace{5 \cdots 5}_n)_8$ b) $b_n = (12, \underbrace{1212 \cdots 12}_{2n})_8$ c) $c_n = (12, \underbrace{2112 \cdots 2112}_{n})_8$

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) $BA, (CA)^\infty$; $b = 16$

2.2 Les systèmes de Calcul Formel

2.2.1 Présentation

Un système de CF est composé de

- un noyau
- des bibliothèques

Les objets manipulés sont:

1. Toutes les structures de données classiques : arbres, listes, permutations, graphes, ensembles, intervalles, tables, tableaux, mots, partitions, compositions, partitions, endofonctions.
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 simplifier des mots (fusion, réunion, contraction etc...).
- (b) Pour les quantités du calcul scientifique : dérivation, intégration et sommation symboliques. équations différentielles, 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

$$\sqrt{2} + \sqrt{3} + \sqrt{5} = \sqrt{10 + 2\sqrt{10} + 2\sqrt{3}\sqrt{7 + 2\sqrt{10}}} \quad (1)$$

comment le système peut-il s'en apercevoir ? Plus difficile, comment le système peut-il donner la premier membre à partir du second ? (problèmes $A = B$, $A \rightarrow B$, voir les fonctions `convert(??,parfrac)`, `rationalize`, `normal`, `collect`) (\leftrightarrow 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

> $B := \sqrt{10 + 2\sqrt{10} + 2\sqrt{3}\sqrt{7 + 2\sqrt{10}}};$ `simplify(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 parce que périodique à partir d'un certain rang) ?

Comment passer d'une représentation à l'autre ?

Combinatoire. —

C'est l'art de manipuler les structures de données discrètes.

Exercice 2.1 i) Bijections $\mathbb{N}^2 \rightarrow \mathbb{N}$ et écriture explicite du couple de rang n .

ii) Arbre de Wilf

iii) Fractions continues.

2.2.2 Exemples de sessions

Interpréteur (cf [3] p17). —

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

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

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

```
5
```

```
> op(4, c);
```

```
u
```

```
> c[4];
```

```
u
```

```
> d := [2, 5*u, 4*v];
```

```
d := [2, 5 u, 4 v]
```

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

```
e := [1, 2, 3, u, v, 2, 5 u, 4 v]
```

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

```
[1, 4, 9, u^2, v^2]
```

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:c:=[n, exp(-x^2)];
```

```
c := [0, e(-x2)]
```

```
> n:=n+1:c:=[n, factor(diff(c[2], x))];
```

```
c := [1, -2 x e(-x2)]
```

```
> n:=n+1:c:=[n, factor(diff(c[2], x))];
```

$$c := [2, 2e^{(-x^2)}(-1 + 2x^2)]$$

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

$$c := [3, -4xe^{(-x^2)}(-3 + 2x^2)]$$

> for i to 7 do n:=n+1:c:=[n,factor(diff(c[2],x))]: od : print(c);

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

> f:=1/(1+x^4);

$$f := \frac{1}{1+x^4}$$

> int(f,x);

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

> with(combinat);

Warning, the protected name Chi has been redefined and unprotected

[Chi, bell, binomial, cartprod, character, choose, composition, conjpart, decodepart, encodepart, fibonacci, firstpart, graycode, inttovec, lastpart, multinomial, nextpart, numbc comb, numbc omp, numbc part, numbc perm, partition, permute, powerset, prevpart, randcomb, randpart, randperm, stirling1, stirling2, subsets, vectoint]

> fibonacci(43);

> 433494437;

> # Les 20 premiers nombres de Fibonacci ...

> [fibonacci(i)\$ i=0..19];

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181]

> P:=fibonacci(43,x);

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

> subs(x=1, P);

(2)

433494437

Exercice 2.2 On considère la suite des nombres de Fibonacci $F_0 = 0, F_1 = 1, F_{n+2} = F_n + F_{n+1}$.

a) Montrer que

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n = \begin{pmatrix} F_n & F_{n+1} \\ F_{n+1} & F_{n+2} \end{pmatrix} \quad (3)$$

b) En déduire que, pour $n \geq 2$, on a

$$F_n = (0 \ 1) \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^{n-2} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (4)$$

c) En déduire une méthode rapide de calcul des F_n .

d) Évaluer la complexité, en nombre d'opérations arithmétiques, de la méthode naïve (fenêtre glissante) et de la méthode proposée en (b).

e) Indiquer comment on peut encore réduire la complexité en remarquant une propriété de symétrie sur les matrices $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n$.

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);

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

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

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


```

(1 + 1/2 x^2 + 3/8 x^4 + 5/16 x^6 + 35/128 x^8 + 63/256 x^10 +
231/1024 x^12 + 429/2048 x^14 + 6435/32768 x^16 + 12155/65536 x^18 + O(x^20))
> f2 := (1 - x)^(-1/2);
(1 + 1/2 x + 3/8 x^2 + 5/16 x^3 + 35/128 x^4 + 63/256 x^5 + 231/1024 x^6 + 429/2048 x^7 +
6435/32768 x^8 + 12155/65536 x^9 + 46189/262144 x^10 + 88179/524288 x^11 + 676039/4194304 x^12 + 1300075/8388608 x^13 + 5014575/33554432 x^14 +
9694845/67108864 x^15 + 300540195/2147483648 x^16 + 583401555/4294967296 x^17 + 2268783825/17179869184 x^18 + 4418157975/34359738368 x^19 + O(x^20))
> 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(x^20))
> 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);
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);

```

Exercice 2.3 Les séries $\log(1+h)$ et $\exp(X)$ étant connues. On pose,

$$(1+h)^\alpha := \exp(\alpha \log(1+h))$$

pour $\alpha \in \mathbb{C}$. D'autre part, pour tout complexe α , on définit

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

i) En utilisant le développement de Taylor, montrer que

$$(1+h)^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} h^k$$

ii) En déduire la formule

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

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();

$$\begin{aligned} & (X + \sqrt{2} + \sqrt{3} + \sqrt{5}) (X + \sqrt{2} + \sqrt{3} - \sqrt{5}) (X + \sqrt{2} - \sqrt{3} + \sqrt{5}) (X + \sqrt{2} - \sqrt{3} - \sqrt{5}) \\ & (X - \sqrt{2} + \sqrt{3} + \sqrt{5}) (X - \sqrt{2} + \sqrt{3} - \sqrt{5}) (X - \sqrt{2} - \sqrt{3} + \sqrt{5}) (X - \sqrt{2} - \sqrt{3} - \sqrt{5}) \end{aligned}$$

> expand(");

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

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

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

> [solve(")];

$$\begin{aligned} & [10 + 2\sqrt{10} + 2\sqrt{3}\sqrt{7 + 2\sqrt{10}}, 10 + 2\sqrt{10} - 2\sqrt{3}\sqrt{7 + 2\sqrt{10}}, \\ & 10 - 2\sqrt{10} + 2\sqrt{3}\sqrt{7 - 2\sqrt{10}}, 10 - 2\sqrt{10} - 2\sqrt{3}\sqrt{7 - 2\sqrt{10}}] \end{aligned}$$

> op(1, ");

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

2.2.3 Histoire succincte des 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 bidimensionnel).

REDUCE (en Lisp) à Stanford.

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).

<http://mupad.sourceforge.net/>

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 polynômes symétriques, les algèbres de Hecke et leurs représentations.

MuPAD Système généraliste qui a des commandes similaires à Maple, mais la communication avec C et C++ est plus simple.

2.3 Introduction à Maple : T.D.

2.4 Quelques structures de données

2.4.1 Arbres binaires complets

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

$$\mathcal{A} = \bullet + \begin{array}{c} \swarrow \quad \searrow \\ \mathcal{A} \quad \mathcal{A} \end{array}$$

Il y a trois notions de *taille* classiques : le nombre de nœuds, le nombre de sommets, la profondeur.

Exercice 2.4 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.

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

$$\tau(\bullet) = 1; \tau((\mathcal{A}_g, \mathcal{A}_d)) = \tau(\mathcal{A}_g) + \tau(\mathcal{A}_d) + 1 \quad (5)$$

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).

$$T = x + xT^2 \quad (6)$$

soit $xT^2 - T + x = 0$ on résout (6) par la méthode habituelle. Le discriminant est $\Delta = 1 - 4x^2$ et les racines possibles sont

$$T_1 = \frac{1 - \sqrt{1 - 4x^2}}{2x}; T_2 = \frac{1 + \sqrt{1 - 4x^2}}{2x} \quad (7)$$

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

$$(1 + X)^\alpha = \sum_{k \geq 0} \binom{\alpha}{k} X^k \quad (8)$$

avec

$$\binom{\alpha}{k} := \frac{(\alpha)(\alpha-1)\cdots(\alpha-k+1)}{k!} \quad (9)$$

ceci qui donne

$$\sqrt{1-4x^2} = \sum_{k \geq 0} \binom{1/2}{k} (-4x^2)^k \quad (10)$$

à l'aide de (9), on a, pour $k \geq 1$

$$\binom{1/2}{k} = \frac{(1/2)(1/2-1)(1/2-2)\cdots(1/2-k+1)}{k!} = \frac{(-1)\cdot(-3)\cdots(3-2k)}{2^k k!} = \quad (11)$$

$$\frac{(-1)^{k-1} 1 \cdot 3 \cdots (2k-3)}{2^k k!} = \frac{(-1)^{k-1} (2k-2)!}{2^{2k-1} k! (k-1)!} \quad (12)$$

en remplaçant ce résultat dans (10) puis dans (7), il vient

$$T = \sum_{k \geq 1} \frac{\binom{2k-2}{k-1}}{k} x^{2k-1} = \quad (13)$$

$$x + x^3 + 2x^5 + 5x^7 + 14x^9 + 42x^{11} + 132x^{13} + 429x^{15} \dots \quad (14)$$

Exercice 2.5 Reprendre l'exercice précédent avec le nombre de sommets comme taille.

2.4.2 Arbres binaires incomplets

L'ensemble \mathcal{A} des arbres binaires est construit par la grammaire (G2)

$$\mathcal{A} = \triangle + \begin{array}{c} \bullet \\ \swarrow \\ \mathcal{A} \end{array} + \begin{array}{c} \bullet \\ \searrow \\ \mathcal{A} \end{array} + \begin{array}{c} \bullet \\ \swarrow \quad \searrow \\ \mathcal{A} \quad \mathcal{A} \end{array}$$

De même que précédemment, il y a trois notions de *taille* classiques : le nombre de nœuds, le nombre de sommets, la profondeur.

Exercice 2.6 Reprendre l'exercice précédent pour ce type d'arbres.

2.4.3 Arbre 1-2

Pour les arbres 1-2, on a la grammaire

$$\mathcal{A} = \bullet + \begin{array}{c} \bullet \\ | \\ \mathcal{A} \end{array} + \begin{array}{c} \bullet \\ \swarrow \\ \mathcal{A} \end{array} + \begin{array}{c} \bullet \\ \searrow \\ \mathcal{A} \end{array}$$

Exercice 2.7 1) Peut-on énumérer ces arbres par nombre de feuilles, par profondeur?
2) Reprendre les exercices précédents pour le nombre de nœuds.

Éléments pour la solution (s'entraîner à rédiger l'exercice précédent). —

Par nombre (total) de nœuds on a

$$T = x + xT + xT^2 \text{ soit } xT^2 + (x - 1)T + x = 0 \quad (15)$$

on obtient

$$T = \frac{1 - x - \sqrt{1 - 2x - 3x^2}}{2x} = x + x^2 + 2x^3 + 4x^4 + 9x^5 + 21x^6 + 51x^7 + 127x^8 + \dots \quad (16)$$

2.4.4 Arité variable

Cette fois, chaque nœud peut avoir un nombre arbitraire de fils. La grammaire en notation symbolique est

$$A = \bullet + \sum_{k \geq 1} \underbrace{(A, A, \dots, A)}_{k \text{ fois}} \quad (17)$$

Exercice 2.8 *Écrire et résoudre la série génératrice de ces arbres.*

2.4.5 Expressions arithmétiques

Une expression arithmétique, impliquant les quatre opérations (+, ×, −, /) est un arbre dont les nœuds sont marqués avec ces opérateurs. Les deux premières opérations (associatives) sont d'arité variable et les deux dernières (non associatives) d'arité deux. Le nombre de feuilles de ces arbres est le *nombre de places de l'expression arithmétique*.

Exemple 2.9 *Par exemple ((○ + ○ + ○) * ○) / (○ - ○) est une expression arithmétique à 6 places.*

Exercice 2.10 1) *Donner la grammaire des expressions arithmétique et les premières d'entre elles.*

2) *Comment définir le domaine de définition d'une expression arithmétique ?*

2.4.6 Mots noncommutatifs

Cette structure est familière aux étudiants de nos formations, c'est celle des *mots* sur un alphabet donné A . La grammaire de ces mots est

$$M = \epsilon + \sum_{a \in A} aM \quad (18)$$

l'ensemble engendré est le *monoïde libre* $M = A^*$.

Exercice 2.11 *Donner la grammaire pour les mots de $\{a, b\}^*$ sans a^2 . Sans a^2 ni b^2 .*

2.4.7 Mots commutatifs

Ce sont les monômes du calcul algébrique. Par exemple pour deux lettres $\{a,b\}$, les monômes sont les expressions $a^i b^j$. Informatiquement, c'est la structure de liste à 2 places.

Remarque 2.12 Ces monômes sont munis d'une structure de monoïde par

$$(a^{i_1} b^{j_1})(a^{i_2} b^{j_2}) = a^{i_1+i_2} b^{j_1+j_2} \quad (19)$$

Plus généralement, soit un alphabet fini $A = \{a_1, a_2, \dots, a_n\}$, le monoïde commutatif libre est défini par

$$A^\oplus = \{a_1^{i_1} a_2^{i_2} \dots a_n^{i_n}\}_{i_k \in \mathbb{N}} \quad (20)$$

la loi étant donnée par

$$(a_1^{i_1} a_2^{i_2} \dots a_n^{i_n})(a_1^{j_1} a_2^{j_2} \dots a_n^{j_n}) = a_1^{i_1+j_1} a_2^{i_2+j_2} \dots a_n^{i_n+j_n} \quad (21)$$

ce qui permet les calculs comme $(x^3 y^5 z^2)(x^2 y^8 z^4) = x^5 y^{13} z^6$.

Exercice 2.13 *Paiement par pièces. (Donné en cours.)*

Codage avec les sous ensembles. Triangles.

2.4.8 Autres séries

Fontaines de pièces (Donné en cours).

3 Les constructeurs set, multiset & list

3.1 Définitions générales

3.2 Set

3.3 List

3.4 Multiset

3.4.1 OGF

3.4.2 EGF

3.4.3 La formule exponentielle

Soit G une classe de graphes étiquetés qui est

1. Stable par renommage. C'est à dire, si on réétiquette les sommets (bijectivement, c'est à dire sans répéter des étiquettes), on reste dans la classe.
2. Stable par composantes connexes c'est à dire, un graphe est dans G ssi toutes ses composantes connexes y sont.

on peut appliquer la formule exponentielle.

Théorème 3.1 Soit $M(n,k)$ le nombre de graphes :

i) Dont les étiquettes sont $[1..n]$

ii) Qui ont k composantes connexes.

On forme la série

$$SGE(G) = \sum_{n,k \geq 0} M(n,k) \frac{z^n}{n!} y^k \quad (22)$$

alors

$$SGE(G) = e^y \sum_{n \geq 1} M(n,1) \frac{z^n}{n!} \quad (23)$$

TODO exemples: classes de Burnside, Nombres Idempotents

3.5 Séries rationnelles (représentations linéaires et aspect automatique)

Exercices préliminaires. — Fontaine de pièces de Wilf, polyomino tas.

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 multiplisités). On a la proposition suivante (énoncée dans le cas général où l'ensemble des scalaires K est un corps commutatif quelconque)

Proposition 3.2 Soit $S = \sum_{n \in \mathbb{N}} a_n z^n \in K\langle\langle z \rangle\rangle$ 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

$$(\exists (\alpha_j)_{0 \leq j < k} \in K^k) (\forall n \in \mathbb{N}) (a_{n+k} = \sum_{j=0}^{k-1} \alpha_j a_{n+j}) \quad (24)$$

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

$$(\forall n \in \mathbb{N}) (a_n = \lambda T^n \gamma) \quad (25)$$

Preuve — ii) \implies iii). —

La relation de récurrence linéaire implique

$$(a_{n+1}, a_{n+2}, \dots, a_{n+k}) = (a_n, a_{n+1}, \dots, a_{n+k-1}) \begin{pmatrix} 0 & 0 & \dots & \dots & 0 & \alpha_0 \\ 1 & 0 & \ddots & & \vdots & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & \vdots \\ \vdots & \vdots & & \ddots & 0 & \alpha_{k-2} \\ 0 & 0 & \dots & \dots & 1 & \alpha_{k-1} \end{pmatrix} \quad (26)$$

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

$$a_n = v_n \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = (a_0, a_1, \dots, a_{k-1}) T^n \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (27)$$

iii) \implies 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

$$(I - zT)^{-1} = \frac{1}{\det(I - zT)} \text{comatrice}(I - zT) \quad (28)$$

et les $Q(z) = \det(I - zT)$ est un polynôme en z tel que $Q(0) = 1$ et $\text{comatrice}(I - zT)$ est une matrice de polynômes en z (exercice !!). D'où le résultat.

i) \implies 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 $n \in \mathbb{N}$,

$$a_{n+N} = -\sum_{j=1}^k \beta_j a_{n+N-j} \quad (29)$$

ce qui entraîne (ii). ♣

Définition 3.3 Un triplet $\mathcal{T} = (\lambda, T, \gamma)$ tel que $\langle S | z^n \rangle = \lambda T^n \gamma$ est appelé représentation linéaire de dimension n de S . De même S est appelée comportement de \mathcal{T} .

Note 3.4 Une série rationnelle admet en général plusieurs représentations linéaires. La dimension minimale de ces représentations est appelée rang de S . C'est aussi la dimension de l'espace vectoriel engendré par les décalées de S .

3.5.1 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

$$\sum_{n=0}^{\infty} a_n z^n \odot \sum_{n=0}^{\infty} b_n z^n = \sum_{n=0}^{\infty} a_n b_n z^n \quad (30)$$

Exercice 3.5 Effectuer les produits de Hadamard suivants

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

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

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 3.6 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.

4 Systèmes et Calcul

4.1 Introduction

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

4.2 Description de la structure d'automate

4.2.1 Graphe pondéré

L'élément de base de ces graphes est la flèche $A = q_1 \xrightarrow{a|\alpha} q_2$ 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-anneau⁵. 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 $c = A_1 A_2 \cdots A_n$ (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 ($k = \mathbb{N}$),

$$u = p \xrightarrow{a|2} q \xrightarrow{b|3} r \xrightarrow{c|5} s \quad (31)$$

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

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

$$\mathbf{q1} \begin{array}{c} \xrightarrow{u|\alpha} \\ \xrightarrow{u|\beta} \end{array} \mathbf{q2} \quad (32)$$

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

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.

semi-anneau.

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

4.2.2 Structure et comportement des automates

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

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

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 flèches entrantes (resp. sortantes) marquées avec des poids. Dans tout ce processus, on peut ne pas indiquer les flèches 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 $\mathcal{A} : A^* \rightarrow k$ (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

$$M(\epsilon) = I_{Q \times Q}, M(w) = M(a_1 a_2 \cdots a_n) = M(a_1) M(a_2) \cdots M(a_n) \quad (33)$$

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,

$$\mathcal{A}(w) := IM(w)T \quad (34)$$

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

$$\mathcal{A}^{r,s}(w) := I_r \left(\sum_{\substack{c, \text{ chemin } l(c)=w \\ t(c)=r, h(c)=s}} \text{weight}(c) \right) T_s \quad (35)$$

alors

$$\mathcal{A}(w) = \sum_{r,s \in Q} \mathcal{A}^{r,s}(w) \quad (36)$$

Cette proposition a le sens intuitif suivant :

1. l'équation (35) 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

4.2.3 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 $\sum_{w \in A^*} F_{|w|} |w|w$, $\sum_{w \in A^*} F_{|w|_a} |w|_b w$

Fin du tronç commun

4.2.4 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 4.2 *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 :

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

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

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

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

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

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 4.3 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

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

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

Exemple 4.4 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 ([?] pp 321) comment coder les pavages, les énumérer, les générer.

e) À l'aide des décalages, former l'automate qui reconnaît la série S .

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

Exercice 4.5 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 "()" 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 $A^* \rightarrow k$ où k est un semi-anneau (i.e. suffisant pour faire le 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 $\lambda \in K^{1 \times n}$, $\mu : A \rightarrow K^{n \times n}$, $\gamma \in K^{1 \times n}$ tels que pour tout $w \in A^*$, $(S, w) = \lambda \mu(w) \gamma$ (où $\mu(\cdot)$ 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.

4.3 Séries

4.3.1 Exemple : Comportement d'un automate

4.3.2 Opérations sur les séries

4.3.3 Lien avec les grammaires et les structures de données

4.3.4 Énumération

4.3.5 Rudiments de Calcul Modulaire

5 Annexes

5.1 Fonctions génératrices

On a utilisé, tout au long de ce cours, la notion de *fonction génératrice*. Une telle fonction se présente sous la forme d'un "polynôme infini"

$$\sum_{n \in I} a_n z^n \quad (37)$$

où $I \subset \mathbb{N}$ est un ensemble d'indices entiers (souvent exprimé par des contraintes). On calcule avec de la même façon qu'avec les polynômes.

Exemple 5.1 *Voici plusieurs exemples*

- $\sum_{k \in \mathbb{N}} F_{2k+1} z^{2k+1} = \sum_{\substack{k \in \mathbb{N} \\ k \equiv 1[2]}} F_k z^k$ (SG des Fibonacci impairs)
- $\sum_{k \in \mathbb{N}} F_{2k} z^{2k} = \sum_{\substack{k \in \mathbb{N} \\ k \equiv 0[2]}} F_k z^k$ (SG des Fibonacci pairs)
- $T_n = \sum_{k \in \mathbb{N}} \binom{n}{k} z^k = \sum_{\substack{k \in \mathbb{N} \\ k \leq n}} \binom{n}{k} z^k$
- $S_k = \sum_{n \in \mathbb{N}} \binom{n}{k} z^k$
- etc ...

5.2 Une variable

5.3 Application au calcul de complexité

5.4 Plusieurs variables

6 TD et TP (E. Laugerotte & J-P. Dubernard)

MI-MIM

TD1 Calcul Formel

2002/2003

Exercice 1 On considère les expressions arithmétiques suivantes :

- $x * (y + z) * (-2)$
- $2 * x/y * z^4$
- $x \wedge (y \wedge z)$

1) Construire les arbres associés. CF

2) À l'aide des fonctions **whattype**, **nops**, **op** de Maple, confirmer et en déduire les types.

3) À l'aide de la fonction **subs** de Maple, remplacer x par $2 * x1 + x2$ dans les expressions précédentes.

Exercice 2 À l'aide des fonctions **whattype**, **nops**, **op** de Maple, donner l'arbre et le type associé à $22/7$ et à 32.6789 .

Exercice 3 (Suite de Fibonacci) Soit (F_n) la suite définie par

$$\begin{cases} F_0 = 0, \\ F_1 = 1, \\ F_{n+2} = F_{n+1} + F_n \quad \forall n \in \mathbb{N}. \end{cases}$$

1) Écrire une procédure récursive Maple efficace calculant F_n .

2) Calculer F_n en fonction de n . Pouvez-vous en déduire une nouvelle procédure ? Est-elle plus performante ?

3) Montrer que, pour tout entier n ,

$$F_{n+1}^2 - F_n F_{n+2} = (-1)^n.$$

4) Montrer que, pour tout entier n ,

$$F_{n+m} = F_m F_{n+1} + F_{m-1} F_n,$$

puis en déduire que

$$\text{pgcd}(F_n, F_m) = F_{\text{pgcd}(n,m)}.$$

Exercice 3 (Algorithme d'Euclide)

1) Écrire une procédure Maple rendant le PGCD de deux nombres entiers naturels a et b .

2) En supposant connu le théorème de Dirichlet qui dit que la probabilité pour que deux entiers positifs u et v soient premiers est de $\frac{6}{\pi^2}$, écrire une procédure qui calcule efficacement le PGCD de plusieurs entiers.

Exercice 4 (Théorème de Lamé) On suppose que a et b sont deux entiers naturels tels que $a > b > 0$.

1) Si l'algorithme d'Euclide demande n itérations pour le calcul du PGCD de a et b , montrer que $a \geq F_{n+2}$ et $b \geq F_{n+1}$. Combien y a-t-il d'itérations si $a = F_{n+2}$ et $b = F_{n+1}$?

2) Avec les mêmes hypothèses qu'en 1), montrer que

$$n + 2 < \frac{\log_{10} \sqrt{5}(a + 1)}{\log_{10} \phi}$$

et

$$n + 1 < \frac{\log_{10} \sqrt{5}(b + 1)}{\log_{10} \phi}$$

où ϕ est le nombre d'or i.e la racine positive de l'équation $X^2 - X - 1 = 0$.

3) Si n est le nombre de chiffres décimaux d'un entier b positif, montrer que

$$\left\lfloor \frac{\log_{10} \sqrt{5}(b + 1)}{\log_{10} \phi} \right\rfloor \leq 5p + 1.$$

4) Si n est le nombre d'itérations de l'algorithme d'Euclide pour $a > b > 0$, montrer que

$$n \leq 5([\log_{10} b] + 1).$$

En déduire la complexité de l'algorithme d'Euclide.

Travaux Pratiques 1 - Opérations sur les entiers

Manipulations 1 Taper, noter les résultats affichés et comprendre...

```
>22/7;  
>evalf(%);  
>a:=23/7;  
>evalf(a,500);  
>18/12;  
>?igcd  
>igcd(456,752);
```

Manipulations 2 Idem.

```
>sum(i,i=1..n);  
>factor(%);  
>sum(i^2,i=1..n);  
>factor(%);
```

Exercice 1

- 1) Calculer la forme factorisée de la somme $\sum_{i=1}^n i^3$.
- 2) Chercher le mode d'emploi de l'instruction `length`.
- 3) Donner le nombre de chiffres de $\sum_{i=1}^{100} i^7$.

Exercice 2

- 1) Calculer $100!$.
- 2) Donner le nombre de chiffres de $100!$.
- 3) Donner les 50 premières décimales de π .

Exercice 3

- 1) Calculer (ou programmer) la somme $S_n = \sum_{k=1}^n \frac{1}{k}$.
- 2) Exprimer de façon exacte puis de façon approchée S_7 , S_{20} , S_{40} , S_{1000} et $S_{1000000}$.

Exercice 4

- 1) Trouver la décomposition en facteurs premiers de 9876543210123456789.
- 2) Calculer le nombre de diviseurs.

Travaux Pratiques 2 - L'algorithme d'Euclide

Manipulations d'ensembles Tapez les ordres suivants et observez les réponses.

```
>A:={1,3,5,7,9};  
>B:={0,2,4,6,8};  
>C:={seq(2*i,i=0..9)};  
>DD:=A union B;  
>X:=A intersect C;  
>Y:=C minus B;
```

Exercice 1 Soit S l'ensemble des entiers strictement inférieurs à 100. Soient A , B et C les sous-ensembles de S respectivement multiples de 2, de 3 et de 5. On note \bar{X} le complémentaire de la partie X de S et on définit l'opération Δ par $X\Delta Y = (X \cap \bar{Y}) \cup (Y \cap \bar{X})$

- 1) Déterminer $(A\Delta B) \cap (A\Delta C)$.
- 2) Déterminer $A\Delta(B \cap C)$.
- 3) A-t-on $(A\Delta B) \cap (A\Delta C) \subset A\Delta(B \cap C)$?

Exercice 2

- 1) Écrire une procédure Maple rendant le premier (resp. dernier) élément d'une liste.
- 2) Écrire une procédure Maple rendant la liste privée de son premier (resp. dernier) élément.
- 3) Écrire une procédure Maple réalisant la fusion de deux listes.
- 4) Écrire une procédure Maple rendant une liste "miroir" de L .

Exercice 3 Les nombres de Mersenne sont de la forme $M_p = 2^p - 1$ avec p premier.

- 1) Écrire les nombres de Mersenne pour $p \leq 31$.
- 2) Parmi eux, quels sont ceux qui sont premiers?
- 3) Décomposer les autres en facteurs premiers et trouver leurs diviseurs.

Exercice 4 Soit p un nombre premier ($p \geq 5$). On définit la somme

$$S_p = \sum_{k=1}^{p-1} \frac{1}{k}.$$

- 1) Calculer S_5 , S_7 , S_{29} sous forme de fraction irréductible.
- 2) Donner le reste de la division du numérateur de S_p par p^2 pour les trois valeurs de p précédentes.
- 3) Calculer le reste de la division du numérateur de S_p par p^2 pour tout p premier dans l'ensemble $\{2, \dots, 100\}$.

Exercice 5

- 1) Écrire une procédure qui calcule le PGCD de deux entiers.
- 2) Écrire une procédure qui calcule le PGCD de plusieurs entiers sachant que la probabilité pour que deux entiers positifs u et v soient premiers est de $\frac{6}{\pi^2}$.
- 3) Écrire une procédure qui calcule le PPMC de plusieurs entiers.

Travaux Pratiques 3 - Représentation graphique

Il y a trois types d'objets Maple que l'on peut représenter graphiquement :

- **une expression**

```
>P:=x^3-3*x^2+x+1;  
>plot(P,x=-5..5,-3..3);
```

- **une fonction**

```
>f:=x->if x=0 then 1 else sin(x)/x fi;  
>plot(f,-3*Pi..3*Pi);
```

- **une ligne polygonale**

```
>L:=[[1,1],[3,4],[1,3],[-1,5]];  
>plot(L);
```

On peut représenter également une famille (g_n) de fonctions (respectivement d'expressions) :

```
>g:=n->n*exp(x)+x;  
>famille:={seq(g(n),n=0..10)};  
>plot(famille,x=-5..5,y=-5..5);
```

Exercice 1 Représenter la fonction f définie par $f(x) = x \sin x$ dans les fenêtres suivantes $[-10,10,-10,10]$ et $[-500,500,-500,500]$.

Exercice 2 Représenter la suite de fonctions (f_n) définie par $f_n(x) = nx/(1 + n^2x^4)$ pour n variant de 0 à 10 dans la fenêtre $[0,2,0,10]$.

Exercice 3 Représenter sur un même graphique la fonction $f(x) = e^x$ et la suite de fonctions (f_n) pour $n \in \{0, \dots, 10\}$ définie par $f_n(x) = \sum_{k=0}^n x^k/k!$. Reprendre le même travail avec la fonction $f(x) = -\ln(1-x)$ et la suite de fonctions (f_n) définie par $f_n(x) = \sum_{k=1}^n x^k/k$.

Exercice 4 Étant donnée une fonction numérique f , un réel a et un entier n , on veut obtenir la liste

$$L_n = [[u_0,0],[u_0,u_1],[u_1,u_1],[u_1,u_2], \dots, [u_{n-1},u_n],[u_n,u_n]]$$

avec

$$\begin{cases} u_0 = a, \\ u_{n+1} = f(u_n), \quad \forall n \in \mathbb{N}. \end{cases}$$

1) Déterminer cette liste quand $f(x) = 1/2(x^2 + 1)$, $a = 1/2$ et $n = 4$. Représenter, si possible sur un même graphique, la courbe de f , la droite $y = x$ et la liste L_4 .

2) Écrire une procédure Maple qui admet une fonction f , un réel a et un indice n , et qui construit la liste L_n . Représenter sur un même graphique, la courbe de f , la droite $y = x$ et la liste L_n avec $a = 1.2$ et $n = 15$.

3) Écrire une procédure Maple qui "automatise" cette étude graphique.

4) Étudier graphiquement les suites définies par les récurrences

$$\begin{cases} u_0 = 1 \\ u_{n+1} = 2u_n/(1 + u_n) - \ln(1 + u_n) \quad \forall n \in \mathbb{N} \end{cases}$$
$$\begin{cases} u_0 = 1 \\ u_{n+1} = 2(1 - e^{-u_n}) \quad \forall n \in \mathbb{N} \end{cases}$$
$$\begin{cases} u_0 = 3.9 \\ u_{n+1} = 2\sqrt{4 - u_n} \quad \forall n \in \mathbb{N} \end{cases}$$

Travaux Pratiques 4 - Nombres aléatoires

Exercice 1 Écrire une procédure qui calcule les n premiers termes de la suite obtenue par la méthode MSM connaissant la graine et le nombre maximum de chiffres dans l'écriture décimale. Quel nombre suit 1010101010 et 3792 dans la méthode MSM? Donner les premiers termes si la graine est 2345678 et le nombre maximum de chiffres est 16.

Exercice 2 On considère un générateur donné par :

$$\begin{aligned} u_0 &= g \\ u_i &= f(u_{i-1}) [m] \quad (i > 0) \end{aligned}$$

- 1) Écrire une procédure calculant le n ème terme de la suite connaissant g , f et m .
- 2) Montrer que la suite est ultimement périodique.
- 3) Soit $\ell(n)$ la plus grande puissance de 2 qui est plus petite ou égale à n . Montrer qu'il existe un entier $n > 0$ tel que $u_n = u_{\ell(n)-1}$. Écrire un algorithme (l'algorithme de Brent) qui calcule les différents paramètres de la suite ultimement périodique. Évaluer l'efficacité du générateur de nombres aléatoires de Maple puis celui du langage C qui est donné par :

$$u_n = 1103515245 \times u_{n-1} + 12345 [2^{31}] \quad (n > 0).$$

- 4) Soit le générateur à deux pas défini par :

$$\begin{aligned} u_0 &= g_0 \\ u_1 &= g_1 \\ u_i &= f(u_{i-1}, u_{i-2}) [m] \quad (i > 1) \end{aligned}$$

Quelle transformation faut-il faire pour appliquer l'algorithme de Brent? Généraliser aux générateurs à n pas.

Exercice 1 Donner des formes closes de fonction génératrice ordinaire et exponentielle de la suite $2, 5, 13, 35, \dots, 2^n + 3^n, \dots$

Exercice 2 Pavage d'un rectangle $2 \times n$

1. Quel est le nombre T_n de façons de recouvrir entièrement un rectangle $2 \times n$ avec des dominos 2×1 ? Nous supposons que les dominos sont tous identiques et que seule compte leur orientation (verticale ou horizontale).

2. Un collectionneur excentrique achète des pavages $2 \times n$ par des dominos au prix de 4 euros le domino vertical et 1 euro le domino horizontal; Combien de pavages valent exactement m euros?

Exercice 3 Résoudre les relations de récurrence suivantes par la méthode des fonctions génératrices :

1. $\begin{cases} F_0 = 0, F_1 = 1, \\ F_n = F_{n-1} + F_{n-2} \text{ sinon.} \end{cases}$
2. $\begin{cases} U_0 = U_1 = 1, \\ U_n = 4U_{n-1} - 4U_{n-2} + n - 1 \text{ sinon.} \end{cases}$
3. $\forall n \geq 2, 2U_n = 3U_{n-1} - U_{n-2}$
4. $\forall n \geq 2, U_n = 4U_{n-1} - 4U_{n-2}$
5. $\begin{cases} U_0 = 1, U_1 = 0, \\ U_n = 3U_{n-1} - 2U_{n-2} + n/2^n \text{ sinon.} \end{cases}$

Exercice 4 Résoudre les relations de récurrence simultanées

$$\begin{cases} U_0 = 1, U_1 = 0 \text{ et } \forall n \geq 2, U_n = U_n = 2V_{n-1} + U_{n-2} \\ V_0 = 0, V_1 = D10 \text{ et } \forall n \geq 2, V_n = U_n = U_{n-1} + U_{n-2} \end{cases}$$

Exercice 5 On suppose que la durée d'un point dans le code morse est de 2 unités de temps et celle d'un trait de 3 unités. Déterminer la fonction génératrice des mots du code morse selon leur durée

Exercice 6 Nombres de Stirling de deuxième espèce

Soit $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$ le nombre de façons de partitionner $[n] = \{1, 2, \dots, n\}$ en k classes (ou sous-ensembles). Par exemple, $\left\{ \begin{matrix} 4 \\ 2 \end{matrix} \right\} = 7$.

1. Déterminer une récurrence sur les $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$.
2. En déduire une expression de $B_k(x) = \sum_n \left\{ \begin{matrix} n \\ k \end{matrix} \right\} x^n$, puis de $\left\{ \begin{matrix} n \\ k \end{matrix} \right\}$.

Exercice 7 Dans le "problème du collectionneur de coupons", imaginons que nous voudrions avoir la collection complète des d photos des stars de cinéma. A chaque achat d'une boîte de céréales, on obtient une de ces photos qui peut bien sûr être une que l'on a déjà.

Soit p_n la probabilité d'obtenir en n achats exactement la collection complète.

1. Montrer que

$$p_n = \frac{d! \binom{n}{k}}{d^n}.$$

2. Soit $p(x) = \sum_n d_n x^n$. Montrer que

$$p(x) = \frac{(d-1)!x^d}{(d-x)(d-2x)\cdots(d-(d-1)x)}.$$

Exercice 8 Soit T un arbre binaire complet à $2n + 1$ sommets. On considère

$$E = \sum_{x \text{ feuille}} \text{prof}(x),$$

$$I = \sum_{x \text{ sommet interne}} \text{prof}(x)$$

Trouver une relation entre E , I et n et la démontrer.

Exercice 9 Soient C_n le nombre d'arbres binaires complets ayant $2n + 1$ sommets et $C(x)$ la fonction génératrice de ces arbres selon le nombre de sommets.

$$C(x) = \sum_{n \geq 0} C_n x^n.$$

1. Déterminer une récurrence sur les C_n .
2. En utilisant le résultat précédent, trouver une équation fonctionnelle pour $C(x)$ et la résoudre.
3. En déduire la valeur de C_n .

Exercice 10 Le but de cet exercice est d'énumérer les objets étudiés en construisant une bijection avec des mots d'un langage.

1. Construire tous les arbres binaires complets ayant au plus 5 sommets.
2. Déterminer un codage de ces arbres par des mots construits sur l'alphabet $\{x, \bar{x}\}$.
3. Calculer le codage des arbres construits en 1.
4. Caractériser le langage engendré par ces mots et déterminer la grammaire algébrique correspondante.
5. En déduire une équation fonctionnelle et la résoudre.
6. En utilisant l'équation fonctionnelle déterminée en 4., écrire un programme Maple calculant les p premiers termes de $C(x)$.

Exercice 11 Après avoir fait une dictée aux élèves de sa classe, un instituteur redistribue les copies. On note b_n le nombre de façons de distribuer les copies de façon à ce qu'aucun élève ne reçoive sa propre copie.

1. Calculer b_1, b_2 et b_3 .
2. Montrer que l'on a la relation $b_n = (n-1)(b_{n-1} + b_{n-2})$, pour $n > 2$.
3. Etablir la relation $b_n - b_{n-1} = (-1)^n$, pour $n \geq 2$.
4. On pose $b_0 = 1$ et on définit la série génératrice exponentielle par

$$b(z) = \sum_{n \geq 0} \frac{b_n z^n}{n!}.$$

Montrer que $b(z) = \frac{e^{-z}}{1-z}$.

Références

- [1] COHEN H., *A Course in Computational Algebraic Number Theory*. Springer (1993)
- [2] CHAR B.W., GEDDES K.O., GONNET G.H., ALI., *Maple V Library Reference Manual*, Springer (1992).
- [3] CHAR B.W., GEDDES K.O., GONNET G.H., ALI., *Maple V Language Reference Manual*, Springer (1992).
- [4] DAVENPORT J., SIRET Y., TOURNIER E., *Calcul formel*, Masson (1986)
- [5] DEMAZURE M., Cours d'algèbre : Divisibilité, Primalité, codes. Cassini (1997).
- [6] VON ZUR GATHEN J. AND GERAHRD J. *Modern Computer Algebra*. Cambridge (1999).
- [7] KNUTH D., *The art of computer programming* Tome I. Addison-Wesley (1981)
- [8] KNUTH D., *The art of computer programming* Tome II. Addison-Wesley (1981)
- [9] NAUDIN P., QUITTÉ C., *Algorithmique Algébrique* Masson (1992)