Combinatoire et Calcul Algébrique

Arithmétique des polynômes

Florent Hivert

Mél : Florent.Hivert@lri.fr
Adresse universelle : http://www.lri.fr/~hivert

Références

- J. von zur Gathen, J. Gerhard, *Modern Computer Algebra*, 3ème éd., Cambridge University Press, 2013.

- A. Casamayou, N. Cohen, G. Connan, T. Dumont, L. Fousse, F. Maltey, M. Meulien, M. Mezzarobba, C. Pernet, N. M. Thiéry, P. Zimmermann *Calcul mathématique avec Sage*, 468 pages, 2013, ISBN : 978-14-811-9104-3. `http://sagebook.gforge.inria.fr/`

- A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy et É. Schost, *Algorithmes Efficaces en Calcul Formel*, 2017, ISBN : 979-10-699-0947-2. `https://hal.archives-ouvertes.fr/AECF/`

Calcul symbolique : structure de donnée

Question

Quelles structures de données utiliser pour les calculs mathématiques ?

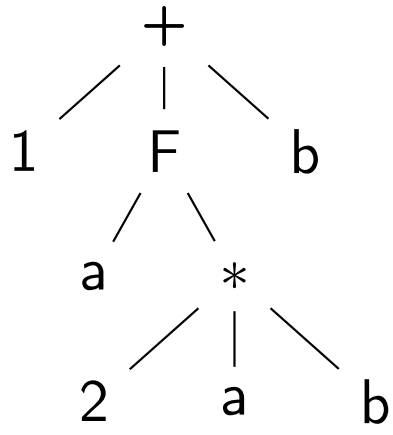
$$\frac{\partial}{\partial a}\left(1 + F(a, 2ab) + b\right) = 2b\,\mathrm{D}_1\left(F\right)\left(a, 2\,ab\right) + \mathrm{D}_0\left(F\right)\left(a, 2\,ab\right)$$

$$\sum_{i=0}^{i=n} i^2 = \frac{n(n+1)(2n+1)}{6} \qquad \sum_{n=1}^{\infty} 2^{-n} = 1 \qquad \sum_{n=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$$

À faire : effectuer ces calculs avec `SageMath`.

Arbres syntaxiques (1)

$$1 + F(a, 2ab) + b$$



Retenir

Codage par arbres syntaxiques

- *c.f. compilateur*
- *analyse syntaxique, arbres de syntaxe abstraits (AST)*

Arbres syntaxiques (2)

Codage par *arbres syntaxiques*

- **Avantages**
  - structure de données très versatile, facilement extensible
  - programmation par filtrage de motifs (pattern matching)
  - facile à implémenter dans un langage fonctionnel
  - très adapté aux calculs en analyses (dérivation, intégrations...)

- **Inconvénients**
  - données très peu structurées
  - structure de donnée récursive
  - beaucoup d'allocation dynamiques, peu de localité des données
  - peu de réutilisation de code, peu d'algorithmes génériques

Arbres syntaxiques : problèmes de sémantique

Codage par *arbres syntaxiques*

Problème

Problème de sémantique :

- si pas de «type» sur les variables, quelles sont les règles de calculs applicables ?

$$M * N = N * M \quad \text{si } M \text{ et } N \text{ sont des matrices...}$$

Arbres syntaxiques : problèmes de sémantique

Codage par *arbres syntaxiques*

Problème

Problème de sémantique :

- si pas de «type» sur les variables, quelles sont les règles de calculs applicables ?

$$M * N = N * M \quad \text{si } M \text{ et } N \text{ sont des matrices...}$$

Arbres syntaxiques : Problème de sémantique

Retenir

L'utilisation d'arbres syntaxiques non typés posent des problèmes de sémantique.

Voici un exemple de ce que ça peut donner :

Factorisation des polynômes

Arbres syntaxiques : Problème de sémantique

Problème

Comment tester l'égalité ?

Théorème (Richardson–Matiyasevich)

Dans la classe des expressions formées à partir d'une variable X et de la constante 1 par les opérations d'anneau $+$, $-$, $\times$ et la composition avec la fonction $\sin(\cdot)$ et la fonction valeur absolue $|\cdot|$, le test d'équivalence à 0 est indécidable.

Arbres syntaxiques : Problème de sémantique

Problème

Comment tester l'égalité ?

Théorème (Richardson–Matiyasevich)

Dans la classe des expressions formées à partir d'une variable X et de la constante 1 par les opérations d'anneau $+$, $-$, $\times$ et la composition avec la fonction $\sin(\cdot)$ et la fonction valeur absolue $|\cdot|$, le test d'équivalence à 0 est indécidable.

Bilan

Retenir

Pour contrôler la sémantique des calculs et les problèmes d'égalité, on se restreint à un domaine particulier d'expressions.

Il est souvent pratique que les objets considérés aient une forme normale (c'est-à-dire une écriture canonique unique).

Dans ce cours : **les polynômes**.

- simple, mais déjà très général
- proche des entiers en précision arbitraire.

Bilan

Retenir

Pour contrôler la sémantique des calculs et les problèmes d'égalité, on se restreint à un domaine particulier d'expressions.

Il est souvent pratique que les objets considérés aient une forme normale (c'est-à-dire une écriture canonique unique).

Dans ce cours : **les polynômes**.

- simple, mais déjà très général
- proche des entiers en précision arbitraire.

Bilan

Retenir

Pour contrôler la sémantique des calculs et les problèmes d'égalité, on se restreint à un domaine particulier d'expressions.

Il est souvent pratique que les objets considérés aient une forme normale (c'est-à-dire une écriture canonique unique).

Dans ce cours : **les polynômes**.

- simple, mais déjà très général
- proche des entiers en précision arbitraire.

Entiers multiprécisions

Retenir

Les entiers machines ont une précision limité.

Par exemple, en C++, sur ma machine :

type	bits	min	max
`int`	32	−2 147 483 648	2 147 483 647
`unsigned int`	32	0	4 294 967 295
`long`	64	−9 223 372 036 854 775 808	9 223 372 036 854 775 807
`unsigned long`	64	0	18 446 744 073 709 551 615

Question

Comment représenter des nombres de taille arbitraires ?

Entiers multiprécisions

Retenir

Les entiers machines ont une précision limité.

Par exemple, en C++, sur ma machine :

type	bits	min	max
`int`	32	−2 147 483 648	2 147 483 647
`unsigned int`	32	0	4 294 967 295
`long`	64	−9 223 372 036 854 775 808	9 223 372 036 854 775 807
`unsigned long`	64	0	18 446 744 073 709 551 615

Question

Comment représenter des nombres de taille arbitraires ?

Représentation des entiers

Retenir

On choisit une base B qui tient dans un mot machine, et on exprime le nombre dans cette base :

$$n = (a_k, \dots a_2 a_1 a_0)_B = \sum_{i=0}^{k} a_i B^i \, .$$

Exemple : `GMP` (pour GNU Multi Precision)
$B = 2^{64}$ sur architecture `x86_64`.

Note : l'instruction machine `Mul` multiplie deux nombres de 64 bits avec résultat 128 bits sur deux registres.

Représentation des entiers

Retenir

On choisit une base B qui tient dans un mot machine, et on exprime le nombre dans cette base :

$$n = (a_k, \dots a_2 a_1 a_0)_B = \sum_{i=0}^{k} a_i B^i \, .$$

Exemple : `GMP` (pour GNU Multi Precision)
$B = 2^{64}$ sur architecture `x86_64`.

Note : l'instruction machine `Mul` multiplie deux nombres de 64 bits avec résultat 128 bits sur deux registres.

Représentation des entiers

Retenir

On choisit une base B qui tient dans un mot machine, et on exprime le nombre dans cette base :

$$n = (a_k, \ldots a_2 a_1 a_0)_B = \sum_{i=0}^{k} a_i B^i \, .$$

Exemple : `GMP` (pour GNU Multi Precision)
$B = 2^{64}$ sur architecture `x86_64`.

Note : l'instruction machine `Mul` multiplie deux nombres de 64 bits avec résultat 128 bits sur deux registres.

Représentation des entiers

Retenir

L'écriture en base B est un polynôme :

$$n = P(B) \quad \text{où} \quad P = \sum_{i=0}^{k} a_i X^i \,.$$

Schématiquement :

$$\textit{arithmétique des entiers} = \textit{arith. des polynômes} \; + \; \textit{retenue}\,.$$

Dans ce cours, on va donc se concentrer sur les polynômes. . .

Représentation des entiers

Retenir

L'écriture en base B est un polynôme :

$$n = P(B) \quad \text{où} \quad P = \sum_{i=0}^{k} a_i X^i \,.$$

Schématiquement :

$$\textit{arithmétique des entiers} = \textit{arith. des polynômes} \; + \; \textit{retenue}\,.$$

Dans ce cours, on va donc se concentrer sur les polynômes. . .

Représentation des polynômes

Retenir (Représentations creuses et denses)

- *Représentation **dense** : on stocke dans un tableau tous les coefficients jusqu'au degré*
- *Représentation **creuse** : on ne stocke que les coefficients non nul avec leur degré (table d'associations, listes triées)*

Note : similaire matrices, vecteurs

Opération de base polynôme dense

Retenir

Hypothèse : polynôme de degré d

- *Évaluation : $O(d)$, optimal*
- *Addition : $O(d)$, optimal*
- *Multiplication naïve : $O(d^2)$ non optimal*

Pour la multiplication, il existe de meilleur algorithmes :

- Karatsuba $O(n^{\log_2(3)}) = O(n^{1,58})$
- Toom-Cook $O(n^{\log_3(5)}) = O(n^{1,46})$
- Schönhage–Strassen (Transformée de Fourier) $O(n \log(n))$

Évaluation/Interpolation

Retenir (Idée)

Pour avoir une multiplication plus rapide de P_1 et P_2 :

- *On évalue $P_1(a)$ et $P_2(a)$ pour un grand nombre de a*
- *On calcule $(P_1 \times P_2)(a) = P_1(a)P_2(a)$*
- *On interpole pour retrouver P à partir des $P(a)$.*

Rappel : Il faut au moins $d+1$-valeurs pour connaître un polynôme de degré d.

L'algorithme de Schönhage–Strassen par Transformée de Fourier, applique cet idée en prenant $a_k = \exp\left(\frac{2i\pi k}{2^n}\right)$ les racines 2^n-ième de l'unité.

Évaluation/Interpolation

Retenir (Idée)

Pour avoir une multiplication plus rapide de P_1 et P_2 :

- *On évalue $P_1(a)$ et $P_2(a)$ pour un grand nombre de a*
- *On calcule $(P_1 \times P_2)(a) = P_1(a)P_2(a)$*
- *On interpole pour retrouver P à partir des $P(a)$.*

Rappel : Il faut au moins $d+1$-valeurs pour connaître un polynôme de degré d.

L'algorithme de Schönhage–Strassen par Transformée de Fourier, applique cet idée en prenant $a_k = \exp\left(\frac{2i\pi k}{2^n}\right)$ les racines 2^n-ième de l'unité.

Évaluation : Méthode de Hörner

Retenir

Le calcul de la valeur $P(a)$ d'un polynôme dense $P(X)$ au point a se fait en utilisant la méthode de Hörner qui consiste à coder l'écriture suivante du polynôme :

$$P(x) = (\cdots((a_d x + a_{d-1})x + a_{n-2})x + \cdots + a_1)x + a_0$$

Coût : d-additions et d-multiplications.

C'est optimal si l'on doit calculer les valeurs en 1 point, mais pas si l'on veux calculer les valeurs en plus de point.

- entiers $0 \ldots n$: opérateur Δ de newton
- Sur les racines de l'unité : transformé de Fourier rapide
- Sur un grand nombre de points : division euclidienne + diviser pour régnier.

Évaluation : Méthode de Hörner

Retenir

Le calcul de la valeur $P(a)$ d'un polynôme dense $P(X)$ au point a se fait en utilisant la méthode de Hörner qui consiste à coder l'écriture suivante du polynôme :

$$P(x) = (\cdots((a_d x + a_{d-1})x + a_{n-2})x + \cdots + a_1)x + a_0$$

Coût : d-additions et d-multiplications.

C'est optimal si l'on doit calculer les valeurs en 1 point, mais pas si l'on veux calculer les valeurs en plus de point.

- entiers $0 \ldots n$: opérateur Δ de newton
- Sur les racines de l'unité : transformé de Fourier rapide
- Sur un grand nombre de points : division euclidienne + diviser pour régnier.