Agrégation externe de mathématiques, texte d'exercice diffusé en 2012
Épreuve de modélisation, option informatique

Résumé : La métamathématique étudie la correction des textes mathématiques à l'aide de raisonnements "finitistes". Au niveau le plus élémentaire, il s'agit d'algorithmique sur des *mots bien formés*. Le texte aborde également les propriétés énumératives des mots bien formés.

Mots clefs : formules, évaluation, pile, énumération, séries génératrices

---

► *Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Des suggestions de développement, largement indépendantes les unes des autres, vous sont proposées en fin de texte. Vous n'êtes pas tenu(e) de les suivre. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury appréciera que la discussion soit accompagnée d'exemples traités sur ordinateur.*

1. Introduction

Hilbert a proposé de considérer les théories mathématiques comme des textes que l'on peut examiner mécaniquement et auxquels on peut appliquer des raisonnements "finitistes" (c'est-à-dire, en substance, élémentaires et intuitivement évidents) afin de vérifier, par exemple, l'absence de contradictions. Pour un *métamathématicien* ces raisonnements relèvent de l'arithmétique élémentaire et de la combinatoire. Mais les ressources de l'algorithmique et de la théorie des langages sont également applicables.

Nous prenons un problème très basique, présenté à la fin du chapitre "Description de la mathématique formelle" du livre "Théorie des ensembles" de Bourbaki. Des règles de formation des mots significatifs ayant été données, Bourbaki formule et démontre un critère numérique pour reconnaitre les mots bien formés. Après avoir résumé ces définitions et ces résultats, nous en examinerons des approches basées sur la théorie des langages et sur un algorithme simple utilisant une pile. À la fin du texte, nous proposons une application à l'énumération.

Notations. Nous noterons $|w|$ la longueur du mot w. Un *facteur* (resp. un *préfixe*, resp. un *suffixe*) de w est un mot v tel que l'on ait une décomposition $w = uvu'$ (resp. $w = vu'$, resp. $w = uv$), où u, u' sont des mots. Le facteur (resp. le préfixe, resp. le suffixe) est dit *strict* si $v \neq w$. Le mot vide est noté ε.

() Option informatique

2. Mots significatifs

Un ensemble fini non vide S de *signes* est donné, avec une application $n : S \to \mathbf{N}$: on dit que $n(s)$ est le *poids* de s. Pour simplifier les notations, nous noterons S_k l'ensemble des signes de poids k. L'application n est étendue à S^* en posant que, pour tout mot w, son poids $n(w)$ est égal à la somme des poids de ses signes. Le poids du mot vide est donc $n(\varepsilon) = 0$.

Les mots *significatifs* sur l'alphabet S sont définis inductivement :

(1) Si $s \in S_0$, alors le mot d'une lettre s est significatif.

(2) Si $s \in S_k$ ($k \geq 1$) et si $w_1, \ldots, w_k \in S^*$ sont des mots significatifs, alors $sw_1 \cdots w_k$ est un mot significatif.

Puisqu'ils ont été définis inductivement, on peut raisonner sur les mots significatifs par induction structurelle : pour démontrer que tous les mots significatifs satisfont une propriété $\mathscr{P}$, on démontre que, si $s \in S_0$, alors le mot d'une lettre s satisfait $\mathscr{P}$; et que, si $s \in S_k$ ($k \geq 1$) et si $w_1, \ldots, w_k \in S^*$ sont des mots significatifs satisfaisant $\mathscr{P}$, alors $sw_1 \cdots w_k$ satisfait $\mathscr{P}$.

Remarque 1. *Dans tout ce texte, nous ne travaillerons qu'avec des expressions préfixes ou postfixes, et jamais infixes. Dans la pratique mathématique, presque tous les opérateurs sont binaires, les assemblages mathématiques sont rarement écrits en notation préfixe ou postfixe, et il faut parenthéser pour lever les ambiguïtés. Naturellement, les définitions et algorithmes présentés ici ne peuvent s'appliquer tels quels.*

Exemple. Prenons $S = S_0 \cup S_1 \cup S_2$ avec $S_0 = \{V, F\}$, $S_1 = \{\neg\}$ et $S_2 = \{\vee\}$. Alors V, $\neg V$, $\vee VF$ et $\vee \neg V \vee VF$ sont significatifs, mais pas $\neg V \vee F$.

2.1. *Mots équilibrés*

On dit qu'un mot $w \in S^*$ est *équilibré* si :

(1) On a : $|w| = n(w) + 1$.

(2) Pour tout préfixe strict v de w, on a : $|v| \leq n(v)$.

Théorème 1. *Pour qu'un mot soit significatif, il faut, et il suffit, qu'il soit équilibré.*

Démonstration. On prouve facilement que tout mot significatif est équilibré par induction structurelle. Pour la réciproque, on établit successivement les deux lemmes suivants :

Lemme 1. *En tout point d'un mot équilibré commence un et un seul facteur équilibré.*

Ce lemme signifie que tout suffixe non vide d'un mot équilibré admet un et un seul préfixe équilibré non vide.

Lemme 2. *Tout mot équilibré s'écrit d'une manière et d'une seule sous la forme $sw_1 \cdots w_k$, où $n(s) = k$ et où $w_1, \ldots, w_k$ sont équilibrés.*

Du théorème 1 et du lemme 2, on déduit le renforcement suivant du théorème :

Corollaire 1. *Tout mot significatif s'écrit d'une manière et d'une seule sous la forme $sw_1 \cdots w_k$, où $n(s) = k$ et où $w_1, \ldots, w_k$ sont significatifs.*

De plus, on dispose d'un algorithme pour réaliser cette décomposition.

2.2. *Le langage des mots significatifs*

La définition des mots significatifs se prête à une description algébrique.

Exemple. Avec le même exemple que précédemment, et en notant X l'axiome, on a la grammaire non contextuelle : $X \to V|F|\neg X|\vee XX$.

Il est clair que la donnée de S et de $n : S \to \mathbf{N}$ permet mécaniquement d'écrire une telle grammaire qui engendre le langage des mots significatifs. Cette grammaire est non ambiguë d'après le lemme 2.

Proposition 1. *(i) Le langage L des mots significatifs est non vide si, et seulement si, $S_0 \neq \emptyset$. Dorénavant, on supposera cette condition vérifiée.*
(ii) Le langage L est infini si, et seulement si, $S \neq S_0$.
(iii) Le langage L est rationnel si, et seulement si, $S = S_0 \cup S_1$.

Indication de démonstration. Seule l'implication directe de (iii) est non triviale. Si l'on a $s \in S_0$ et $t \in S_k$, avec $k \geq 2$, on prouve à l'aide du lemme d'itération que $L \cap \{s,t\}^*$ n'est pas rationnel, donc L non plus.

On supposera dorénavant que $S_0 \neq \emptyset$ et que $S \neq S_0 \cup S_1$. Le langage L n'est donc pas rationnel, et l'on devine que tout algorithme d'analyse devra mettre en jeu implicitement un automate à pile.

3. Évaluation par pile et reconnaissance

3.1. *Termes et évaluation*

Soient Ω un ensemble (fini non vide) de *symboles d'opérateurs* et $\alpha : \Omega \to \mathbf{N}$ l'application *arité*. On définit inductivement les *termes sur cette signature* comme suit :

– Les *constantes* (symboles d'arité 0) sont des termes
– Si $\alpha(\omega) = k \geq 1$, et si $t_1, \ldots, t_k$ sont des termes, alors il y a un terme noté $\omega(t_1, \ldots, t_k)$.

Pour *évaluer* les termes, c'est-à-dire définir une application de l'ensemble $\mathscr{T}$ des termes dans un domaine $\mathscr{D}$ de valeurs, il faut interpréter les symboles de Ω. On se donne donc, pour toute constante c de Ω une valeur $\overline{c} \in \mathscr{D}$; et, pour tout symbole ω d'arité $k \geq 1$, une application $\overline{\omega} : \mathscr{D}^k \to \mathscr{D}$. L'évaluation peut alors se faire récursivement :

```
evaluation(t) =
  si alpha(t) = 0
    alors valeur(t)
```

```
    sinon (filtrer t avec omega(t1,...,tk);
           omegabarre(evaluation(t1),...,evaluation(tk))) ;;
```

Il existe un algorithme itératif simple d'évaluation manipulant directement une pile. Pour l'utiliser, on part de l'écriture postfixe (polonaise) du terme t. L'évaluation est alors réalisable au fil de la lecture.

```
tant que possible
   (lire un symbole omega; soit k = alpha(omega) dans
    si k = 0
       alors empiler valeur(omega)
       sinon (pour i de 1 a k (depiler x ; v[i] := x);
              empiler omegabarre(v[k],...,v[1]))) ;
retourner le sommet de pile ;;
```

Théorème 2. *Cet algorithme réalise correctement l'évaluation de tout terme.*

Indication de démonstration. Elle se fait par induction structurelle avec comme hypothèse d'induction : *la concaténation de la pile et du suffixe restant à analyser est un terme sémantiquement équivalent au terme initial.*

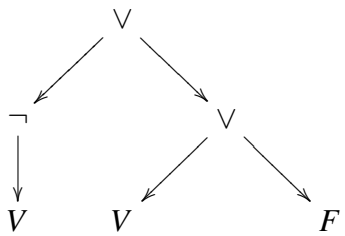
On en déduit que le codage postfixe est non ambigu : l'algorithme permet de reconstituer les termes à partir de leur écriture postfixe. On en déduit également un critère de bonne formation de l'écriture postfixe :

Corollaire 2. *Pour que la suite de symboles $s_1,\dots,s_n$ de Ω soit l'écriture postfixe d'un terme, il faut, et il suffit, que les $h_p = \sum_{i=1}^{p}\big(1-\alpha(s_i)\big)$ vérifient : $h_1,\dots,h_{n-1} > 0$ et $h_n = 1$.*

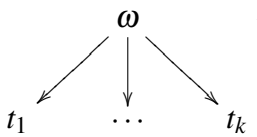
Démonstration. À la lecture d'un symbole d'arité k, la hauteur de pile augmente de $1-k$.

3.2. *Application aux mots significatifs*

La définition des mots significatifs s'apparente, de manière évidente, à celle de l'écriture préfixe des termes. On peut donc tirer de l'algorithme ci-dessus un algorithme d'analyse. Pour cela, on se fixe comme but d'associer à tout mot significatif (ou au terme dont il est l'écriture préfixe) sa représentation arborescente. Par exemple, au mot significatif $\vee\neg V\vee VF$, on voudrait associer l'arbre



L'algorithme ci-dessus permet cette reconstitution : il suffit d'interpréter une constante par un arbre réduit à une feuille (étiquetée) et un symbole ω d'arité $k \geq 1$ comme l'application qui, à

k arbres $t_1,\dots,t_k$, associe l'arbre  Il faut bien sûr parcourir à l'envers

le mot analysé, puisque l'algorithme d'évaluation par pile s'applique à des écritures *postfixes* alors que nos mots significatifs sont des notations *préfixes*. Ainsi, le critère numérique ci-dessus donne ici :

Théorème 3. *Pour que la suite de symboles $s_1,\dots,s_n$ de Ω soit l'écriture préfixe d'un terme, il faut, et il suffit, que les $h_p = \sum_{i=1}^{p}\big(1-\alpha(s_i)\big)$ vérifient : $h_1,\dots,h_{n-1} \le 0$ et $h_n = 1$.*

Indication de démonstration : On doit d'abord démontrer que la suite $s_1,\dots,s_n$ est une écriture préfixe correcte si, et seulement si, la suite miroir $s'_1 = s_n,\dots,s'_n = s_1$ est une écriture postfixe correcte ; noter que cela ne va pas de soi et qu'il ne s'agit pas là des deux écritures d'un même terme ! On voit ensuite que la suite $h'_1,\dots,h'_n$ attachée à la suite miroir est donnée par $h'_k = h_n - h_{n-k}$ (en convenant que $h_0 = 0$), et l'on applique le corollaire 2.

Notons qu'on peut également en déduire une nouvelle preuve du théorème 1 en comparant la condition "être équilibré" sur l'écriture préfixe avec le critère numérique sur l'écriture postfixe.

4. Énumération

La non-ambiguïté de la grammaire qui engendre les mots significatifs, ou, de manière équivalente, le corollaire du théorème 1, a pour conséquence des propriétés énumératives. Notons L le langage des mots significatifs, et, pour tout $s \in S$, notons L_s l'ensemble des mots de L qui commencesnt par s. Alors L est l'union disjointe des L_s. Si $n(s) = 0$, $L_s = \{s\}$. Si $n(s) = k \ge 1$, L_s est en bijection avec l'union disjointe : $\coprod_{s_1,\dots,s_k \in S} L_{s_1} \times \cdots \times L_{s_k}$. Notons a_n le nombre de mots significatifs de taille n et $a_n(s)$ le nombre de mots de L_s de taille n ; on a donc $a_n = \sum_{s\in S} a_n(s)$. Ainsi, $a_0 = 0$; $a_1(s) = 1$ si $s \in S_0$, et 0 sinon, donc $a_1 = \operatorname{Card} S_0$. Pour $n \ge 1$, on déduit des égalités précédentes : $a_n = \sum_{k\ge1}\sum_{s\in S_k} a_n(s)$ et $a_n(s) = \sum_{k\ge1}\sum_{\substack{s_1,\dots,s_k\in S\\ n_1+\cdots+n_k=n-1}} a_{n_1}(s_1)\cdots a_{n_k}(s_k)$, d'où la formule :

$$a_n = \sum_{k\ge1} \operatorname{Card} S_k \sum_{n_1+\cdots+n_k=n-1} a_{n_1}\cdots a_{n_k}.$$

Pour résoudre cette récurrence, on introduit la série génératrice : $\mathscr{L}(z) = \sum_{n\ge0} a_n z^n$. En tant que série entière, son rayon de convergence est supérieur ou égal à $\dfrac{1}{\operatorname{Card} S}$, puisque, de manière évidente, $a_n \le (\operatorname{Card} S)^n$.

Théorème 4. *Notons : $P_S(X) = \sum_{k\ge0} (\operatorname{Card} S_k) X^k$. On a l'égalité : $\mathscr{L}(z) = zP_S(\mathscr{L}(z))$.*

Démonstration. On multiplie la relation de récurrence par z^n, et l'on somme pour $n \geq 1$.

Exemple. Dans l'exemple de la page 2, on a $P_S(X) = 2+X+X^2$, d'où : $\mathscr{L} = z(2+\mathscr{L}+\mathscr{L}^2)$. La seule solution série entière est : $\mathscr{L}(z) = \dfrac{1-z-\sqrt{1-2z-7z^2}}{2z}$. La formule du binôme (généralisée) de Newton donne $(1-u)^{1/2} = 1 - \dfrac{1}{2}u - \dfrac{1}{8}u^2 - \dfrac{1}{16}u^3 - \dfrac{5}{128}u^4 + O(u^5)$, d'où $\sqrt{1-2z-7z^2} = 1-z-4z^2-4z^3-12z^4+O(z^5)$, et : $\mathscr{L}(z) = 2z+2z^2+6z^3+O(z^4)$. On a donc $a_1 = a_2 = 2$ et $a_3 = 6$, ce qu'il est aisé de vérifier. Il n'y a pas de formule générale simple pour a_n, mais on peut contourner le problème de la manière suivante. La série entière $\sqrt{1-2z-7z^2}$ s'écrit $\sqrt{1-(1+2\sqrt{2})z}\sqrt{1-(1-2\sqrt{2})z}$.

Lemme 3. *Si* $0 < |a| < |b|$, *alors* $\sqrt{(1-az)(1-bz)} = \sum_{n\geq 0} \alpha_n z^n$ *avec* $\alpha_n \sim \sqrt{1-\dfrac{a}{b}}(-1)^n b^n \binom{1/2}{n}$.

Indication de démonstration. On sait que : $\sqrt{1-bz} = \sum_{n\geq 0} (-1)^n b^n \binom{1/2}{n} z^n$. On écrit :

$$\sqrt{(1-az)(1-bz)} - \sqrt{1-\frac{a}{b}}\sqrt{1-bz} = \frac{a}{b}\frac{(1-bz)^{3/2}}{\sqrt{1-az}+\sqrt{1-\dfrac{a}{b}}}.$$

Comme $\binom{3/2}{n}$ est négligeable devant $\binom{1/2}{n}$ et que le rayon de convergence de $\dfrac{1}{\sqrt{1-az}+\sqrt{1-\dfrac{a}{b}}}$ est $\dfrac{1}{|a|} > \dfrac{1}{|b|}$, la conclusion s'ensuit.

On complète cette formule par les relations :

$$\binom{1/2}{n+1} = (-1)^n 2^{-2n-1}\frac{1}{n+1}\binom{2n}{n} \sim \frac{1}{2\sqrt{\pi n^3}}.$$

(L'égalité est un calcul classique, l'équivalence résulte de la formule de Stirling). Pour en revenir à notre exemple, on a finalement un équivalent du nombre de termes de taille n :

$$a_n \sim -\sqrt{\frac{4-\sqrt{2}}{7}}(-1)^{n+1}(1+2\sqrt{2})^{n+1}\binom{1/2}{n+1} \sim \frac{1}{2}\sqrt{\frac{4-\sqrt{2}}{7\pi}}(1+2\sqrt{2})^{n+1}n^{-3/2}.$$

De manière générale, la seule connaissance du polynôme P_S permet de donner précisément le comportement asymptotique des coefficients de $\mathscr{L}$. Voici une information partielle mais facile à obtenir.

Théorème 5. *Le rayon de convergence de* $\mathscr{L}$ *est égal à* $r = \dfrac{\alpha}{P_S(\alpha)}$, *où* α *est l'unique racine strictement positive de* $P_S - XP_S'$.

() Option informatique

Indication de démonstration. On se place dans le cas "non dégénéré" où S_0 n'est pas vide et où il existe un S_k non vide avec $k \geq 2$. Considérons la fonction de variable réelle $\frac{x}{P_S(x)}$ sur $\mathbf{R}_+$. Sa dérivée a le signe de $P_S(x) - xP'_S(x)$, qui est un polynôme dont le coefficient constant est positif, le coefficient de degré 1 nul, et les autres sont strictement négatifs. Sur $\mathbf{R}_+$, la fonction croît strictement jusqu'en $\alpha > 0$ puis décroît strictement. La fonction réciproque de la restriction de cette fonction à $[0, \alpha]$ est la restriction de $\mathscr{L}$ à $\mathbf{R}_+$, qui a une tangente verticale en r.

Appliqué à notre exemple, ce théorème donnerait l'information suivante : le nombre de mots significatifs de taille n est un $O(\rho^n)$ pour tout $\rho > 1 + 2\sqrt{2}$ et pour aucun $\rho < 1 + 2\sqrt{2}$. C'est plus faible que le calcul direct. En fait, on a le théorème (beaucoup plus difficile) :

Théorème 6 (Meir & Moon)**.** *On suppose que S est "apériodique", c'est-à-dire que les poids des éléments de S ont pour pgcd* 1. *Alors :* $s_n \sim cr^{-n}n^{-\frac{3}{2}}$, *où :* $c = \sqrt{\frac{P_S(\alpha)}{2\pi P''_S(\alpha)}} > 0$ *et* $r \in]0;1[$. *(La définition de* α *et r est la même que ci-dessus.)*

Exercice de programmation :

► *Il vous est demandé de rédiger un programme conforme aux spécifications ci-dessous dans l'un des langages C, Caml ou Java à votre choix. Ce programme devra être accompagné d'un exemple d'exécution permettant d'en vérifier le bon fonctionnement. La clarté et la concision du programme seront des éléments importants d'appréciation pour le jury.*

Programmer la reconnaissance des mots bien formés *en écriture préfixe* sur la signature de l'exemple du texte à l'aide du critère fourni par le théorème 3 page 5. Il est conseillé de représenter le mot à valider sous forme d'un tableau ou d'une liste de caractères.

Suggestions pour le développement

► *Soulignons qu'il s'agit d'un menu à la carte et que vous pouvez choisir d'étudier certains points, pas tous, pas nécessairement dans l'ordre, et de façon plus ou moins fouillée. Vous pouvez aussi vous poser d'autres questions que celles indiquées plus bas. Il est très vivement souhaité que vos investigations comportent une partie traitée sur ordinateur et, si possible, des représentations graphiques de vos résultats.*

- Démontrer le théorème 1 page 2 et son corollaire.
- Dire comment obtenir une description par grammaire non contextuelle (page 3) mécaniquement à partir de $n : S \to \mathbf{N}$, et prouver la proposition 1 page 3.
- Adapter le théorème 1 page 2 et la proposition 1 page 3 au cas d'expressions infixes parenthésées (remarque 1 page 2).
- Démontrer le théorème 2 page 4 et son corollaire.

() Option informatique

– Démontrer le théorème 3 page 5 et en déduire une nouvelle preuve du théorème 1, selon l'indication du texte. (Il faudra préxiser le lien entre écritures préfixe et postfixe d'un même terme.)
– Démontrer le théorème 4 page 5 et détailler l'exemple.
– Prouver le théorème 5 page 6 et dire quelle information asymptotique s'en déduit.
– Appliquer le théorème 6 page 7 à l'exemple.
– Étendre les définitions, résultats et algorithmes de ce texte au cas de signatures *hétérogènes*. On peut, par exemple, envisager (comme Bourbaki) l'existence de deux sortes d'assemblages : les termes, comme $\vee\neg VF$ et les *relations*. Ainsi, $=t_1t_2$ est une relation (où t_1,t_2 sont des termes) et $\Rightarrow r_1r_2$ est encore une relation (où r_1,r_2 sont des relations).