Langages algébriques :
à la frontière entre la Combinatoire
et l'Informatique

M. DELEST

Résumé. Nous donnons un rappel sur la désormais classique méthode de Schützenberger ainsi que les deux extensions principales qui en ont été faites : les grammaires à opérateurs, les q-grammaires.

1 - INTRODUCTION

Dès les années 60, des travaux théoriques ont débutés sur l'étude syntaxique des langages en vue de produire des compilateurs performants. C'est en 64 [27] que Randell et Russel ont réalisé la première étude détaillée d'un compilateur pour ALGOL 60 dont les structures syntaxiques restent encore d'actualité pour les langages actuels. Le fait de pouvoir ou non reconnaître un langage passe par l'intermédiaire des générateurs des langages appelés grammaires. Les traductions en langages machines sont obtenues par des compilateurs construits à partir des grammaires et de règles de traduction. La méthode généralement utilisée est celle des attributs du à Irons [24][25][1]. Une automatisation de ces procédés est possible dans le cas de grammaires ayant des propriétés particulières par les programmes LEX et YACC très connus et dans le domaine public c'est-à-dire accessible via une connexion par FTP sur un site de l'INTERNET. Ces programmes sont fort utilisés par les personnes qui réalisent des compilateurs ou des interfaces homme-machine.

Au delà de ces travaux liés à l'Informatique très vite les langages sont apparus comme un moyen pour coder les objets. M.P. Schützenberger [28] [29] a montré une étroite relation entre les problèmes d'énumération en combinatoire et les classifications entre langages. L'utilisation de langages algébriques apparaît alors comme un lien élégant entre le niveau combinatoire et le niveau analytique. Soit Ω une classe d'objets combinatoires énumérés par une suite d'entiers a_n où n est la valeur d'un paramètre p sur ces objets. La méthodologie de M.P. Schützenberger consiste en trois étapes. Tout d'abord, construire une bijection entre les objets de Ω et les mots d'un *langage algébrique* de telle manière que pour un objet ω de Ω, le paramètre p sur ω devienne un nombre de lettres dans le mot codant ω. La notion de grammaire est alors essentielle puisqu'elle décrit le système non commutatif correspondant. L'utilisation de ce qui s'appelle le passage à l'image commutative permet alors de déduire une équation dont la fonction génératrice de la suite a_n est solution. Deux extensions de cette méthode ont été données. La première due à Cori et Richard [9] introduit des opérateurs afin de construire à partir d'un langage des langages plus complexes. Elle a permis notamment de prouver des formules de Tutte sur les cartes planaires [7]. La seconde méthode due à Delest et Fédou [13] consiste à introduire l'opération de traduction des langages. Cette technique est issue de la méthode des attributs utilisée pour décrire les mécanismes de traduction au sein d'un compilateur. Elle permet essentiellement d'obtenir des équations non algébriques (q-équations) à partir d'un codage algébrique des objets.

LaBRI - Université Bordeaux I, 351 Cours de la Libération, 33405 TALENCE CEDEX , FRANCE

Ce travail a reçu un soutien du PRC Mathématiques et Informatique et du Département de Mathématiques de l'Université de Melbourne.

Dans cet exposé, après quelques définitions et notations, nous exposerons d'abord la notion de grammaire telle qu'elle apparaît naturellement en Informatique. Nous montrerons ensuite la relation entre langages et séries formelles et essentiellement les problèmes liés à l'ambiguïté. Nous donnons ensuite les grandes lignes de la méthode de Schützenberger. Nous décrirons ensuite les notions de grammaires à opérateurs et de q-grammaires. Certaines des notions décritent ici interviennent dans l'implémentation du logiciel CalICo[1] [15][11] que nous réalisons.

2 - DEFINTIONS ET NOTATIONS

Cette section est un bref résumé des notions utiles pour comprendre la suite de ce texte. Des détails plus complets peuvent être trouvés dans Berstel [3], Ginsburg [22], Aho et Ullmann [1].

Soit X un ensemble non vide appelé alphabet. Les éléments de X sont appelés lettres. On appelle mot une suite de lettre de X. Le mot vide est noté usuellement par ε. Soit u et v deux mots sur X, $u=u_1...u_p$ et $v=v_1....v_q$. On définit la concaténation de deux mots comme le mot $uv=u_1...u_pv_1....v_q$. On note X^* le monoïde libre engendré par X c'est à dire l'ensemble des mots écrits sur X muni de l'opération de concaténation.

Le nombre d'occurrence d'une lettre x dans un mot u est noté $|u|_x$. Le nombre de lettres de w est appelé longueur de w et noté $|w|$.

Un langage est un sous ensemble de X^*. A tout langage L, on peut associer une fonction génératrice

$$L = \sum_{w \in \mathbf{L}} w,$$

qui est lément de Z<<X>> algèbre des séries formelles non-commutatives à variables dans X et coefficients dans Z.

3 - GRAMMAIRES ALGEBRIQUES

Dans ce paragraphe, nous définissons la notion de grammaire et décrivons deux exemples de base en Combinatoire et Informatique.

DEFINITION 1. *Une grammaire algébrique est un quadruplet* G=<N,X,P,s> *tel que* N *et* X *sont deux alphabets disjoints appelés respectivement alphabet non terminal et terminal,* s *est un élément de* N *appelé axiome et* P *un ensemble de paires* (α, β) *avec* $\alpha \in N$ *et* $\beta \in (N \times X)^*$ *appelées règles de production que l'on note aussi* $\alpha \to \beta$.

---

[1] CalICo est un logiciel développé au sein du Groupe de Combinatoire Enumérative de Bordeaux. Les personnes participantes en 1994 sont Y. Chiricota, M. Delest, J.M. Fédou, V. Gaudin, G. Mélançon, N. Rouillon.

Soit α dans N, et u dans $(N \times X)^*$ $u=u_1\alpha u_2$, on appelle dérivation dans G une réécriture du mot u en $v=u_1\beta u_2$ avec (α, β) dans P. On notera $u \rightarrow v$. On dit qu'un mot w dérive d'un symbole non terminal α dans G si il existe une suite de dérivations permettant de réécrire α en w. On notera $\alpha \overset{*}{\rightarrow} w$. L'ensemble des mots

engendrés par s est appelé langage algébrique engendré par G, on note L(G). En général, il existe plusieurs grammaires pour un langage algébrique donné.

EXEMPLE 1. L'exemple le plus classique de l'Informatique est le langage des expressions arithmétiques bien formées sur une seule lettre. Ces mots sont générés par la grammaire G_1 donnée par N={E,T,F}, X={a,**+**,*,(,)}, s=E, et les productions

E . E**+**T , E . T , T . T*F , T . F , F . (E), F . a .

Un exemple de mot engendré par cette grammaire est (a **+** a * a) **+** a * a. L'arbre de syntaxe abstraite c'est-à-dire celui déduit de la dérivation d'un mot dans G est un arbre binaire qui permet un calcul efficace de telles expressions en machine (voir figure 1). Dans la suite, puisque nous parlerons de séries formelles, pour plus de clarté, nous substituons à la lettre terminale **+** (resp. *, resp. (, resp.)) la lettre p (resp. m, resp. x, resp. —x).
Toutes les instructions et toutes les phrases "comprises" par un compilateur donné sont décrites par une grammaire algébrique. Le langage Pascal comme son prédécesseur ALGOL étaient décrits pour l'utilisateur par une grammaire.

EXEMPLE 2. L'exemple le plus classique en Combinatoire est le langage de Dyck qui code de nombreuses structures arbres, chemins, polyominos par exemple. Ces mots sont générés par la grammaire G_2 donnée par

N={D}, X={x, $\bar{x}$ }, s=D,

et les productions

D . x D $\bar{x}$ D , D . ε .

Un exemple de mot engendré est x x $\bar{x}$ x $\bar{x}$ $\bar{x}$ x $\bar{x}$. On trouvera figure 2 diverses structures combinatoires codées par ce mot.

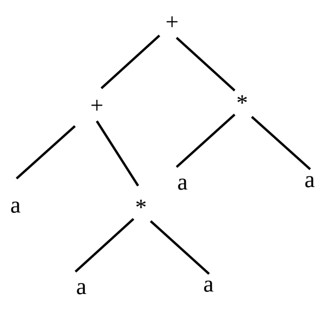

Figure 1. Arbre binaire associé à l'expression (a+a*a)+a*a.

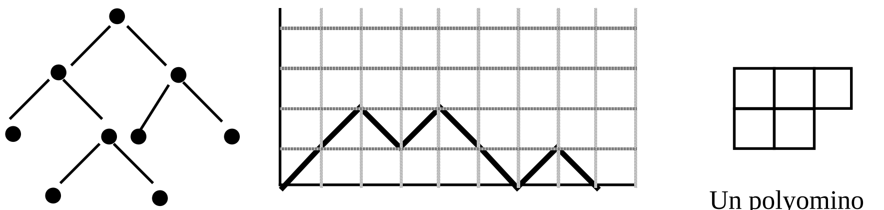

Figure 2. Objets combinatoires codés par le mot $x\, x\, \bar{x}\ \ x\, \bar{x}\ \ \bar{x}\ \ x\, \bar{x}$.

Les deux exemples ci-dessus sont ceux de grammaires algébriques non ambiguës c'est-à-dire telles que tout mot du langage engendré ne peut être obtenu qu'une seule fois par les règles de productions à partir de l'axiome dans une dérivation gauche-droite (on dérive d'abord le non terminal le plus à gauche). Dans ce cas les séries formelles associées aux langages vérifient des équations qui se déduisent directement des règles de production. On a ainsi pour G_2 le système

$$E = E\,\mathbf{p}T + T$$

$$T = T\mathbf{m}F + F$$

$$F = \mathbf{x}E\,\bar{\mathbf{x}} \ + a$$

et pour G_1, l'équation $D = x\, D\, \bar{x}\ D + \varepsilon$.

4 - MÉTHODOLOGIE DE SCHÜTZENBERGER

Cette méthodologie apparaît comme une idée de M.P. Schützenberger dès 1959 [28][29]. Par la suite, le premier a avoir développé cette idée fut Gross [23]. Des développements ont été donnés par Cori et Richard [9] conduisant à des énumérations sur les cartes planaires [7], d'autres par Fliess [3] avec des applications à la résolution d'équations différentielles, enfin Viennot avec des applications en combinatoire énumérative [33] et Flajolet au calcul assymptotique [21]. Signalons en 1975, en France, une école d'Informatique Théorique sur les séries formelles dans laquelle de nombreux articles sur ce sujet ont été édités par Berstel [3]. Cette méthode est connue maintenant sous le nom de DSV-méthodologie à la suite d'une demande de M.P. Schützenberger à Viennot [32].

Soit X un alphabet, X= $\{x_1, ..., x_k\}$. Le passage à l'image commutative permet d'obtenir à partir de la série non-commutative, une série commutative appelée série énumérative du langage et définie par

$$\chi_0(\mathrm{L}) = \sum_{(i_1, i_2, ..., i_k) \in \mathrm{N}^k} n_{i_1 i_2 \ldots i_k}\, x_1^{\,i_1} x_2^{\,i_2} \ldots x_k^{\,i_k}$$

telle que $n_{i_1 i_2 \ldots i_k}$ est le nombre de mots w de L tel que pour tout $j \in [1..k]$, $|w|_{x_j} = i_j$. On définitainsi une application de B<<X>> dans N[[X]] demi-anneau de séries formelles commutatives à variables dans X. On notera souvent L la série $\chi_0(L)$. L'application χ_0 n'est pas un morphisme mais on a les propriétés suivantes:

PROPRIETE 2. *Soient* A, B, C *trois langages sur* X. *Suposons que les opérations produit et somme sont non ambiguës*

(i) *si* A=BC *alors* χ_0(A) = χ_0(B) χ_0(C),

(ii) *si* A=B+C *et* B(C *est vide alors* χ_0(A) = χ_0(B) + χ_0(C),

(iii) *si* A *ne contient pas le mot vide alors* $\chi_0(A^*) = (1-\chi_0(A))^{-1}$.

Ces propriétés de l'application χ_0 sont les briques essentielles des travaux fondamentaux dûs à M.P. Schützenberger [28] dont une conséquence est

THEOREME 3. *L'image par* χ_0 *d'un langage algébrique est une série algébrique composante de la solution du système d'équations obtenu par* χ_0 *à partir d'une grammaire non ambiguë de ce langage.*

EXEMPLE 3. On peut calculer la série énumérative associée a G_2, on a le système commutatif suivant

$$E = \mathbf{p}ET + T\ ,\ T = \mathbf{m}TF + F\ ,\ F = \mathbf{x\bar{x}}\ E + a\ .$$

On a alors l'équation suivante

$$E = \frac{(1+pE)(x+Ez^2)}{1-mx-mx\bar{x}E},$$

et on obtient par exemple ainsi la fonction génératrice des nombres e_n d'expressions arithmétiques bien formées en identifiant toutes les variables à x

$$E(x) = \sum_{n\geq 0} e_n x^n = \frac{1-3x^2+\sqrt{1-6x^2+x^4}}{4x^3}.$$

Cet exemple est emprunté au livre d'exemples de calcul asymptotique en $\Lambda_{\Upsilon}\,\Omega$ [21]. Plus classique est le calcul sur le langage de dyck qui mène à l'équation

$$D = x\bar{x}D^2 + 1.$$

Un calcul analytique trivial montre que

$$D(x,\bar{x}) = \frac{1-\sqrt{1-4x\bar{x}}}{2x\bar{x}},$$

dont on déduit aisément que le nombre de mots de Dyck de longueur 2n est le nombre de Catalan

$$C_n = \frac{1}{n+1}\binom{2n}{n}.$$

De nombreux problèmes d'énumération dans divers domaines ont été résolus par cette méthode dont seules quelques références peuvent être listées ici : polyominos [10][33], structures secondaires de ^{t}RNA [30]. On trouvera une vue générale de ces résultats dans [31][34].

REMARQUE. Certains langages algébriques ne sont associés à aucune grammaire non ambiguë. Flajolet [20] a montré que leurs séries génératrices sont liées aux séries transcendantes.

4 - GRAMMAIRES A OPERATEURS

L'intérêt de considérer des grammaires est de pouvoir faire passer à travers la syntaxe une certaine sémantique de manière à obtenir des traductions des textes sources en des programmes exécutables performants. L'idée sous-jacente est d'obtenir à partir de langages simples des langages plus complexes. Cori et Richard [9] utilisèrent cette idée pour énumérer des objets que l'on ne peut pas coder par des langages algébriques mais dont la série énumératrice est algébrique. Pour cela ils ont introduits la notion d'équations à opérateur. Ils montrèrent ainsi que pour certaines de ces équations non algébriques la série génératrice des objets peut être algébrique. L'application est donnée dans le domaine des cartes par Cori [7]. C'est cette approche que nous voulons exposer ici partiellement. On définit l'image commutative d'un opérateur.

Définition 4. Si H est un opérateur de B<<X>> dans B<<X>>, H_{χ_o} est l'opérateur de N[[X]] dans N[[X]] qui s'il existe est tel que H_{χ_o} o $\chi_0 = \chi_0$ o H.

EXEMPLE 4. Les morphismes alphabétiques ont une image commutative. Soit H_1 l'opérateur qui conserve le mot vide et dans un mot non vide ne conserve que la première lettre. H_1 n'admet pas d'image commutative. Cori dans [7] définit l'opérateur Λ sur B<<{x,y}>> de la manière suivante:

si $|f|_x= 0$ alors $\Lambda(f)=0$ sinon $f=y^n xg$ et $\Lambda(f)= y^{n+1}g$.

Soit s(x,y) une série de N[[{x,y}]]. L'image commutative de l'opérateur Λ est donnée par:

$$\Lambda_{\chi_0}(s(x,y)) = \frac{y}{x}\left(\chi_0(s(x,y)) - \chi_0(s(0,y))\right)$$

Il est clair que le fait qu'un opérateur commute permettra de prendre l'image commutative d'un système faisant intervenir cet opérateur. Le problème étudié concerne les conditions de résolution pour de telles équations. D'autres auteurs ont regardé par la suite les systèmes pour lesquels on pouvait déduire des propriétés d'algébricité de la série génératrice du langage solution. Ainsi Chottin [6] en a donné une formalisation puis Dulucq [18] a introduit des φ-dérivations pour lesquelles des résultats d'algébricité peuvent être prouvés. L'un des résultats est la preuve bijective par Cori, Dulucq, Viennot [8] du fait que $C_n C_{n+1}$ compte les produit de mélange de deux mots de Dyck et déduisent le nombre de permutations de Baxter alternantes. Par la suite cette technique fut utilisé à nouveau dans le calcul asymptotique de la complexité de l'algorithme de Naïmi-Trehel par Arnold, Delest et Dulucq [2].

5 - q-GRAMMAIRES

En Informatique, la compilation nous enseigne que la méthode des attributs sémantiques décrite par Irons [24][25] puis Knuth [26] permet d'associer une traduction à tout mot d'un langage algébrique. Certaines de ces traductions ne sont pas des langages algébriques. Ce point est d'un grand intérêt pour la combinatoire énumérative puisque certains problèmes connus comme algébriques pour certains paramètres sont aussi non algébriques pour d'autres. On prendra pour exemple le cas de la série énumérative des diagramme de Ferrers (représentation d'une partition, figure 3) qui est algébrique suivant le périmètre

$$f(x) = \frac{x^2}{1-2x},$$

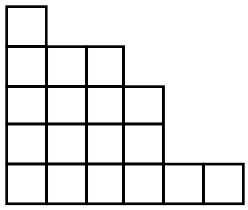

Figure 3. Représentation d'une partition d'un entier
par un diagramme de Ferrers.

et non algébrique suivant l'aire

$$f(q) = \prod_{i=1}^{\infty} \frac{1}{1-q^i} .$$

L'intérêt de la méthode des attributs réside dans le fait que chaque traduction est définie localement sur chaque règle de la grammaire. Ceci est un moyen très efficace de construire un traducteur fiable. En effet, les grammaires des compilateurs

comptent fréquemment une centaine de règles. Les problèmes de traduction en fonction du contexte d'utilisation d'une règle parmi les cents peut être ramené à un problème local. Le lecteur trouvera dans [26] une étude plus générale. Nous nous restreignons aux attributs "synthétisés". Pour introduire cette définition, nous décrivons tout d'abord un exemple.

EXEMPLE 5. Afin d'évaluer les expressions arithmétiques des compilateurs, des instructions machines de chargement dans des registres du processeur sont générées. Ces opérations de chargement sont très coûteuses en temps, il faut donc minimiser leurs nombres. On calcule le nombre minimum de registres nécessaires pour l'évaluation de l'expression avant génération du code par la grammaire attribuée suivante:

E . E+T, ν(E):=si ν(E)=ν(T) alors ν(T)+1 sinon max(ν(E),ν(T))
E . T , ν(E):=ν(T)
T . T*F, ν(T):=si ν(T)=ν(F) alors ν(T)+1 sinon max(ν(E),ν(T))
T . F , ν(T):=ν(F)
F . (E), ν(F):=ν(E)
F . a . ν(F):=1

On trouvera Figure 4 le calcul de cet attribut sur l'arbre de dérivation décrit figure1. Ce calcul n'est autre que celui du nombre de Strahler des arbres binaires [30] qui intervient également en Biologie dans le paramètre ordre des ^{t}RNA. On a montré figure 4 ce calcul sur l'arbre de syntaxe abstraite de l'expression. On trouvera un survol de ce sujet dans [34].

Nous donnons cidessous une définition très parcellaire des grammaires d'attributs.

DEFINITION 5. *Une grammaire d'attributs est une grammaire algébrique* G=<V,X,p,S> *ayant un ensemble fini d'attributs* A(Y) *associé à chaque non terminal* Y *de* V. *Chaque attribut* τ *est une application de* V *dans un domaine* D_τ. *Un attribut* τ *est dit synthétisé si pour chaque règle de* p *de la forme*

$$Y . u_1 Y_1 u_2 Y_2 ..... Y_n u_{n+1}$$

telle que u_i *est dans* X^*, $\tau(Y)$ *peut-être calculé récursivement comme suit*

$$\tau(Y) = f(\gamma_1(Y_1), ..., \gamma_n(Y_n))$$

avec pour tout k *dans* [1,n], γ_k *est dans* $A(Y_k)$ *et* f *est une application de* $D_{\gamma_1} \wedge ... \wedge D_{\gamma_n}$ *dans* D_τ. *L'image d'un mot* w *par* τ *est égal à la valeur de* $\tau(S)$ *dans la dérivation* S^*. w *de* w *dans* G.

Suivant cette idée de traduction, en Combinatoire, on peut espérer ramener le problème de trouver des récurrences sur un objet suivant un paramètre non algébrique à un problème local de traduction sur des configurations particulières algébriques. Nous nous bornerons ici au cas simple où la traduction est équivalente à l'adjonction d'une lettre [13].

Definition 6. *Une* q-*grammaire* (G,τ) *est une grammaire d'attribut telle que :*
(i) G = <V,X,p,S> *une grammaire non ambiguë,*
(ii) τ e*st un attribut synthétisé à valeur dans* C[X"{q}],
(iii) *Pour chaque lettre* x *dans* X *et chaque mot* w *dans* l(G) *on a*

$$|\tau(w)|_x = |w|_x .$$

La traduction $\tau(w)$ d'un mot w est appelé q-analogue du mot w et est notée (w;q). Soit ${}^qG = (G,\tau)$ une q-grammaire.

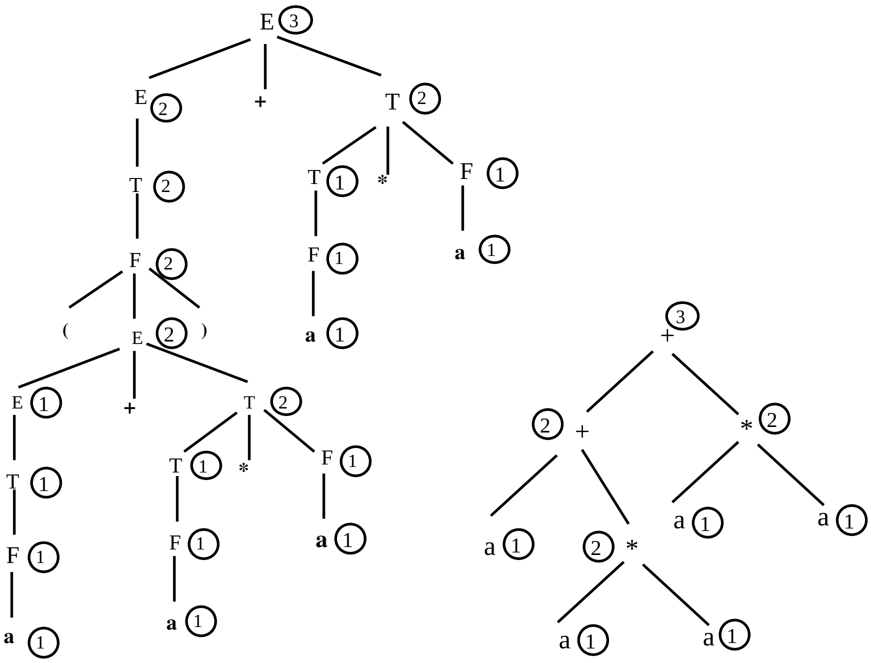

Figure 4. Arbre de dérivation attribué et nombre de strahler de l'arbre de syntaxe abstraite.

Définition 7. *Le* q-*analogue* ^{q}L(G) *du langage* L(G) *est défini par*

$$ {}^{q}L(G) = \sum_{w \in L(G)} (w;q). $$

Définition 8. *Soit* ^{q}G = (G,τ) *une* q-*grammaire*. *La fonction génératrice de* ^{q}L(G) *est définie par*

$$ {}^{q}l = \sum_{\text{w[l(G)}} \chi_1(\tau(w))\, \chi_0(w) $$

avec χ_1 *le morphisme qui supprime toutes les lettres de* X *dans un mot de* $(X \cup \{q\})^*$.

On montre [16] que si l'attribut est linéaire (en un sens que nous ne préciserons pas ici), on obtient des systèmes de q-équations directement à partir de la grammaire attribuée, l'intervention de l'attribut se manifeste dans le passage à l'image commutative par des substitutions non algébriques dans l'équation. Le cas général de plusieurs attributs synthétisés a été depuis traité par P. Duchon. Dans ce cadre, des résultats similaires sont décrits.

EXEMPLE 6. Le langage codant les diagrammes de Ferrers est le langage codant le bord supérieur du diagramme par des mots w=aub écrits sur l'alphabet {a,b}. Une grammaire est G = <{S,L},{a,b},{S$\rightarrow$ aLb, L $\rightarrow$ aL; L $\rightarrow$ bL; L $\rightarrow$ ε}, S}. L'attribut τ définit ci-dessous permet de calculer l'aire à partir de ce codage

$S \rightarrow a\,L\,b$,	$\tau(S) = q^{\text{À}\tau(L)\text{À}_a+\text{À}\tau(L)\text{À}_b+1}\ a\,\tau(L)\,b$
$L \rightarrow a\,L$,	$\tau(L) = q^{\text{À}\tau(S)\text{À}_b}\ a\,\tau(L)$,
$L \rightarrow b\,L$,	$\tau(L) = b\,\tau(L)$,
$L \rightarrow \varepsilon$,	$\tau(L) = \varepsilon$.

On montre que le système de q-équations dans ce cas est

$$ {}^{q}S(a,b) = q\ a\ {}^{q}L(aq,bq)\ b, $$

$$ {}^{q}L(a,b) = a\ {}^{q}L(a,bq) + b\ {}^{q}L(a,b) + \varepsilon\ , $$

dont on peut déduire la fonction génératrice bien connue

$$ {}^{q}l(a,b) = \sum_{n=0}^{\infty} \frac{a^n q^{n+1}}{(1-qb)(1-q^2 b)\ldots(1-q^{n+1}b)}. $$

Un point non résolu pour l'instant est celui de l'utilisation possible d'attributs hérités dans des problèmes d'énumération.

CONCLUSION

Si dans le cas de la DSV-méthodologie d'origine les solutions sont à coup sûr obtenues car algébriques il n'en va pas de même dans les deux extensions signalées. Une autre extension est en cours de développement par Dutour et Fédou [19], il s'agit de grammaires d'objets. Il n'existe pas de méthodes systématiques de résolution de telles équations. M. Bousquet-Mélou dans [5] donne ainsi la solution de q-équations apparaissant souvent dans l'énumération des polyominos. Ces résultats réunis ont permis de produire la formule donnée par Dubernard et Dutour [17].

NOTE

Cet article aurait pu être celui de l'équipe de Combinatoire Bordelaise qui depuis 1975 sous la direction de Robert Cori et Xavier Viennot a considérablement développé cette démarche.

REMERCIEMENTS

L'auteur remercie Tony Guttmann pour son accueil chaleureux et efficace en Australie.

REFERENCES

[1] A. AHO, J. ULLMAN, "The theory of parsing, translation and compiling", Prentice Hall, Englewood Cliffs, NJ (1973).
[2] A. ARNOLD, M. DELEST, S. DULUCQ, Complexité de l'algorithme d'exclusion mutuelle de Naïmi-Trehel, Actes du 37ème Congrès de l'ACFAS, UQUAM, Montréal, Canada, Mai 1989.
[3] J. BERSTEL et alt, "Séries formelles en variables non commutatives et applications", Actes de la cinquième Ecole de Primptemps d'Informatique Théorique, Vieux-Boucau les Bains (1977).
[4] J. BETREMA, J.G. PENAUD, Animaux et arbres guingois, TCS 117 (1993), 67-89.
[5] M. BOUSQUET-MELOU, A method for the enumeration of various classes of column-convex polygons, Rapport LaBRI No. 578-93, soumis a publication.
[6] L. CHOTTIN, Etude syntaxique de certains langages solutions d'équations avec opérateurs, T.C.S. 5 (1977), 51-84.
[7] R. CORI, Un code pour les graphes planaires et ses applications, Astérisque 27 (1975).
[8] R. CORI, S. DULUCQ, X. VIENNOT, Shuffle of parenthesis systems and Baxter permutations, J. of Comb. Th. A, 43 (1986), 1-22.
[9] R. CORI, J. RICHARD, Enumération des graphes planaires à l'aide des séries formelles en variables non commutatives, Discrete Maths., 2 (1972), 115-162.
[10] M. DELEST, Polyominoes and animals: some recent results, J. of Maths Chemistry, 8 (1992), 3-18.
[11] M. DELEST, J.M. FEDOU, G. MELANCON, N. ROUILLON, Faire des mathématiques avec CalICo, à paraître dans le Bulletin des Annales du Québec.
[12] M. DELEST, J.M. FEDOU, Enumeration of skew Ferrers diagrams, Discrete Maths, 112 (1993), 65-79.
[13] M. DELEST, J.M. FEDOU, Attribute grammars are usefull for combinatorics, Actes des Journées Traitement des séries en variables non commutatives de Paris, Rapport Université de Rouen, Eds. G. Duchamps, G. Jacob et D. Krob, Avril 1990, TCS 98 (1992) 65-76.
[14] M. DELEST, X. VIENNOT, Algebraic languages and polyominoes enumeration, Theor. Comp. Sci., 34(1984), 169-206.
[15] M. DELEST, N. ROUILLON, CalICo a software for combinatorics, à paraître dans AMS series.
[16] J.P. DUBERNARD, q-grammaires et polyominos parallélogrammes, Thèse de l'Université Bordeaux 1, 1993.
[17] J.P. DUBERNARD, I. DUTOUR, Enumération de polyominos convexes dirigés, à paraitre dans les actes de SFCA'94.
[18] S. DULUCQ, Equations avec opérateurs : un outil combinatoire, Thèse de 3ième cycle, Bordeaux, 1981.
[19] I. DUTOUR, J.M. FEDOU, Grammaires d'objets, pré-publication, LaBRI 1994.
[20] P. FLAJOLET, Ambiguity and transcendence, rapport INRIA, 397 (1985).

[21] P. FLAJOLET, B. SALVY, P. ZIMMERMAN, Automatic average-case analysis of algorithms, TCS 79, 1 (1991), 37-109.
[22] S. GINSBURG, "The mathematical theory of context free languages", McGraw-Hill, New-York, 1966.
[23] M. GROSS, Applications géométriques des langages formels, I.C.C. Bul., 5 (1961), 141-168.
[24] E.T. IRONS, A syntax directed compiler for Algol 60, Comm. ACM 4 (1961) 51-55.
[25] E.T. IRONS, Towards more versatile mechanical translations, Proc. Symp. Appl. Math., 15 (1963) 41-50.
[26] D.E. KNUTH, Semantics of context-free languages, Maths. Sys. 2 (1968), 127-145.
[27] B. RANDELL, L.J. RUSSEL, *Algol 60 implementation*, Academic Press, New-York, 1964.
[28] M.P. SCHÜTZENBERGER, Certain elementary families of automata, in Proc. Symp. on Mathematical Theory of Automata, Polytechnic Institute of Brooklyn, 1962, 139-153.
[29] M.P.SCHÜTZENBERGER, Context-free langages and pushdown automata, Information and Control 6 (1963), 246-264.
[30] M. VAUCHAUSSADE, X. VIENNOT, Enumeration of RNA secondary structures by complexity, Actes Mathematics in Biology and Medecine, Bari (1985), Lecture Notes in Biomathematics, 1985, 360-365.
[31] X. VIENNOT, Enumerative combinatorics and algebraic languages, Actes FCT'85, ed. L. Budach, Lecture Notes in Computer Science, 199, 450-464.
[32] X. VIENNOT, entretien privé avec M.P. Schützenberger, Mai 1991.
[33] X. VIENNOT, A survey of polyominoes enumeration, Actes SFCA'92, Ed. P. Leroux et C. Reutenauer, Montréal, 1992, 399-420.
[34] X. VIENNOT, Trees dans "Mots", mélange offert à M.P. Schützenberger, M. Lothaire, Ed. hermes, Paris, 1990, 265-291.