RAIRO. INFORMATIQUE THÉORIQUE

A. CARDON

M. CROCHEMORE

Détermination de la représentation standard d'une série reconnaissable

RAIRO. Informatique théorique, tome 14, nº 4 (1980), p. 371-379

<http://www.numdam.org/item?id=ITA_1980__14_4_371_0>

R A I R O Informatique théorique/Theoretical Informatics
(vol 14, n° 4, 1980, p 371 a 379)

DETERMINATION DE LA REPRÉSENTATION STANDARD D'UNE SÉRIE RECONNAISSABLE (*)

par A. CARDON et M. CROCHEMORE [1]

Communique par J BERSTEL

Resumé — *On étudie la minimisation des représentations linéaires d'une série reconnaissable en variables non commutatives à coefficients dans un corps On décrit un algorithme de minimisation reprenant les grandes lignes de celui déjà esquissé par M P Schutzenberger, et on montre qu'il peut être réalisé avec une complexité arithmétique en* $O(mn^3)$, *où n est la dimension de la représentation initiale et m le cardinal de l'alphabet*

Abstract — *We study the minimisation of linear representations of a recognizable formal power serie in noncommutative variables and with coefficients in a field We describe a minimisation algorithm developing the basic ideas of the one sketched by M P Schutzenberger, and we show that it can be realized within arithmetical complexity* $O(mn^3)$, *where n is the dimension of the original representation and m is the cardinal of the alphabet*

INTRODUCTION

Les séries formelles en variables non commutatives sont une généralisation naturelle des langages formels consistant essentiellement à attribuer à chaque mot une multiplicité. Cette multiplicité peut apparaître comme le degré d'ambiguité d'un mot dans un processus de génération (ou de reconnaissance) d'un langage : grammaire ou automate fini par exemple.

Les séries reconnaissables forment, dans ce cadre, la généralisation naturelle des langages rationnels (ou réguliers) et aussi celle des séries rationnelles en une variable (obtenues en développant en série le quotient de deux polynômes). Ces séries interviennent non seulement en théorie des langages [4, 7], mais également comme un important moyen de calcul symbolique pour les asservissements linéaires et non stationnaires [5].

On sait que, de même que tout langage rationnel peut être obtenu à partir d'un automate fini, toute série reconnaissable peut être obtenue à partir d'une représentation linéaire du monoïde libre. Dans le cas des séries en une variable,

(*) Reçu le 26 février 1979, et dans sa version finale le 18 décembre 1979
([1]) Laboratoire d'Informatique, Université de Rouen, Mont-Saint-Aignan

ces représentations correspondent aux relations de récurrence linéaire entre les coefficients. On démontre [8] qu'à chaque série correspond une unique représentation de dimension minimale, dite standard; ce résultat est à la fois une généralisation de l'existence d'une relation de récurrence linéaire de longueur minimale, et de l'existence d'un automate minimal.

Le but de cet article est de décrire la mise en œuvre d'un algorithme de calcul d'une représentation standard à partir d'une représentation linéaire, et d'en évaluer la complexité. On montre que celle-ci est bornée par $O(mn^3)$, ou m est le cardinal de l'alphabet et n la dimension de la représentation initiale.

L'algorithme originel de calcul d'une représentation standard (Schützenberger [8]) repose sur la construction de bases d'espaces vectoriels dont on connait un système générateur, et nécessite le calcul d'un ensemble préfixiel maximal associé à une base contenue dans le système générateur [2]. On simplifie cet algorithme en utilisant, par analogie avec la théorie des automates, la notion d'accessibilité, et en déterminant des bases qui ne sont plus nécessairement extraites du système générateur, et construites triangularisées. Grâce à cette modification, on peut mettre en œuvre l'algorithme avec une complexité arithmétique en $O(mn^3)$, alors qu'une méthode naïve conduirait à un algorithme en $O(mn^4)$. Toutefois, l'algorithme originel peut également être réalisé en temps $O(mn^3)$ en utilisant une procédure analogue à la décomposition L.U.P. d'une matrice (*voir* [3]), mais sa constante de temps est plus importante que celle de l'algorithme présenté ici.

On rappelle, dans la première partie de cet article quelques résultats classiques relatifs à la projection d'une représentation sur une représentation standard. Dans la seconde partie, on décrit l'algorithme de standardisation en « pidgin-algol » [1], puis on l'analyse et évalue sa complexité en nombre d'opérations arithmétiques.

I. REPRÉSENTATION STANDARD D'UNE SÉRIE RECONNAISSABLE

Nous présentons quelques résultats connus sur les séries reconnaissables et les notations qui sont nécessaires à l'exposé de l'algorithme de standardisation.

1. Séries reconnaissables

A désigne un alphabet de cardinal m et A^* le monoïde libre engendré par A, dont ε est l'élément neutre. K étant un corps, $K\langle\langle A\rangle\rangle$ désigne l'ensemble des séries formelles en les variables non commutatives a de A. La valeur d'une série S

en un mot f de A^* est notée (S, f), et ainsi nous écrivons S sous la forme :

$$S = \sum_{f \in A^*} (S, f) f.$$

Les séries reconnaissables sont celles pour lesquelles il existe :

– un entier $n > 0$,

– une forme linéaire λ définie sur K^n, une représentation linéaire μ de A^* dans le monoïde multiplicatif $K^{n \times n}$, et un vecteur ρ de K^n, avec la propriété :

$$\forall f \in A^* (S, f) = \lambda \cdot \mu f. \rho.$$

Nous dirons que (λ, μ, ρ) est une représentation de dimension n de la série S.

2. Matrice de Hankel

A une série S est habituellement associée l'application :

$$\mathcal{H} : \quad A^* \times A^* \to K,$$
$$(f, g) \to (S, fg),$$

dite matrice de Hankel de S et qui définit en fait un tableau infini dont lignes et colonnes sont indicées par les mots de A^*.

La reconnaissabilité d'une série est équivalente à la finitude du rang de sa matrice de Hankel. Ceci permet d'introduire la notion de représentation standard : une représentation d'une série reconnaissable est dite *standard* si sa dimension est égale au rang de sa matrice de Hankel.

Le théorème suivant [5] précise ces notions :

THÉORÈME 1 : *Soient K un corps et S une série de $K \langle\langle A \rangle\rangle$.*
S est reconnaissable ssi $\mathcal{H}$ est de rang fini; et si S est reconnaissable alors S admet une représentation standard.

Les représentations standards sont en fait les représentations de dimension minimale.

3. Projection d'une représentation

Par analogie avec la théorie des automates, nous introduisons la notion d'accessibilité d'une série reconnaissable. Notons que ceci correspond, dans le cadre plus général des K-modules sériels, à la notion de complète commandabilité [6].

Soit (λ, μ, ρ) une représentation de la série S. On note $\mathscr{E}$ l'espace vectoriel engendré par $\mu A^*.\rho$ et $\mathscr{F}$ celui engendré par $\lambda.\mu A^*$. On dit que la représentation est *accessible* (respectivement coaccessible) lorsque sa dimension est égale à celle de $\mathscr{F}$ (respectivement de $\mathscr{E}$).

Si (λ, μ, ρ) et (λ', μ', ρ') sont deux représentations de S, on définit un morphisme φ de (λ, μ, ρ) sur (λ', μ', ρ') par :

$$\varphi: \quad \mathscr{E} \to \mathscr{E}'$$

vérifiant :

$$\forall f \in A^*, \quad \varphi.\mu f = \mu' f.\varphi,$$

$$\rho' = \varphi\rho,$$

$$\lambda = \lambda'.\varphi.$$

Nous pouvons alors énoncer un théorème de projection [6, 3] :

THÉORÈME 2 : *Toute représentation d'une série non nulle se projette sur une représentation accessible (respectivement coaccessible). Pour toute représentation accessible, il existe un isomorphisme de représentation de celle-ci sur la représentation standard.*

On déduit notamment de ce théorème que la représentation standard est unique à un isomorphisme près, et qu'une représentation est standard si et seulement si elle est accessible et coaccessible.

II. L'ALGORITHME DE STANDARDISATION

Nous considérons une représentation de dimension n d'une série reconnaissable. L'algorithme de détermination de la représentation standard est constitué de deux étapes : dans la première nous projetons la représentation initiale sur une représentation accessible; dans la seconde étape nous projetons cette dernière représentation sur une représentation coaccessible. En vertu du théorème 2, cette dernière représentation est standard, et constitue la représentation cherchée. En fait les deux étapes procèdent de façon duale l'une par rapport à l'autre en échangeant lignes et colonnes, et nous n'expliciterons donc que la première étape.

Cette première étape repose sur la détermination d'une base de $\mathscr{F}$; nous la construisons triangularisée et ceci nous permet d'obtenir une complexité arithmétique en $O(mn^3)$, avec une faible constante de proportionnalité.

1. Algorithme déterminant une représentation accessible

Soit une représentation (λ, μ, ρ) de dimension n d'une série reconnaissable : on cherche une représentation accessible (λ', μ', ρ') de la même série.

L'algorithme détermine une suite maximale $\mathscr{V}=(v_1, v_2, \ldots, v_m)$ de vecteurs de l'espace vectoriel $\mathscr{F}$ engendré par $\lambda.\mu A^*$, telle que :

(1) l'ensemble $\{v_1, \ldots, v_m\}$ soit libre;

(2) pour tout a de A et pour tout v_r de $\mathscr{V}$, $v_r.\mu a$ soit linéairement dépendant de l'ensemble $\{v_1, \ldots, v_m\}$.

C'est relativement à l'ensemble ainsi déterminé qu'est exprimée la représentation accessible cherchée (λ', μ', ρ').

On construit la suite $\mathscr{V}$ et simultanément la représentation μ' par adjonctions successives de vecteurs à la suite $(v_1, \ldots, v_s)$ des s premiers vecteurs de $\mathscr{V}$ déjà calculés, d'une part, et de lignes à la suite $((\mu' a)_1, \ldots, (\mu' a)_{r-1})$ des lignes de $\mu' a$ déjà déterminées.

Le calcul des vecteurs de la suite $\mathscr{V}$ et de la représentation μ' se fait comme suit :

(*a*) on choisit un vecteur v_r dans $\mathscr{W}=(v_1, \ldots, v_s)$ et une lettre a de A;

(*b*) on détermine le vecteur w de l'espace vectoriel E engendré par $\{v_1, \ldots, v_s\} \cup \{v_r.\mu a\}$, de façon que :

– si $v_r.\mu a$ est indépendant de $\{v_1, \ldots, v_s\}$, $\{v_1, \ldots, v_s, w\}$ engendre E, et le système $(v_1, \ldots, v_s, w\}$ soit triangularisé à l'ordre des composantes près,

– si $v_r.\mu a$ est dépendant de $\{v_1, \ldots, v_s\}$, $w=0$;

(*c*) simultanément, on détermine le vecteur α des composantes de $v_r.\mu a$ sur l'ensemble $\{v_1, \ldots, v_s, w\}$ (ou $\{v_1, \ldots, v_s\}$ si $w=0$). C'est ce vecteur α qui sert à définir la r-ième ligne de $\mu' a$.

En cas d'indépendance de $v_r.\mu a$ par rapport à $\{v_1, \ldots, v_s\}$ on pose $v_{s+1}=w$, et on ajoute ce vecteur à $\mathscr{W}$.

On répète les étapes (*a*), (*b*), (*c*) pour les autres lettres de A d'une part, et jusqu'à ce que tous les vecteurs de $\mathscr{W}$ aient été examinés d'autre part.

L'algorithme se termine par le calcul de λ' et de ρ'.

Le calcul de w et de α définis dans les étapes (*b*) et (*c*) se fait dans une procédure auxiliaire DEP.

L'algorithme est donné en « pidgin-algol » dans la figure 1 sous forme d'une procédure que l'on note STAND. (λ, μ, ρ) est une représentation de dimension n d'une série et (λ', μ', ρ') est une représentation accessible de la même série.

On note les indices de composantes des vecteurs entre crochets.

```
  Procédure STAND (λ, μ, ρ, λ', μ', ρ')
  début
      DEP (0, λ, α);
      si λ=0 alors écrire « série nulle »;
      sinon
          début
              r ←
              s ← 1;
              v_s ← λ;
1             tant que r<s faire
                  début
                      r ← r+1;
2                     pour tout a dans A faire
                          début
3                             w ← v_r.μa;
4                             DEP(s, w, α);
5                             (μ'a)_r ← α;
                              si w≠0 alors
                                  début
                                      s ← s+1;
                                      v_s ← w;
                                  fin
                          fin
                  fin
6             pour i ← 1 jusqu'à s faire
                  début
                      λ'[i] ← 0;
                      ρ'[i] ← v_i.ρ;
                  fin
7             λ'[1] ← 1;
          fin
  fin
```

Figure 1

2. Procédure de dépendance DEP

A partir de la suite $(v_1, \ldots, v_s)$ de vecteurs indépendants triangularisés, et d'un vecteur $\bar{w}$, la procédure DEP détermine un vecteur w de sorte que la suite $(v_1, \ldots, v_s, w)$ soit triangularisée et engendre le même espace vectoriel que $(v_1, \ldots, v_s, \bar{w})$. Dans le cas où $\bar{w}$ est dépendant de $(v_1, \ldots, v_s)$, w est nul, et la procédure donne les composantes de $\bar{w}$ sur ce système, dans un vecteur noté α. Si $\bar{w}$ est indépendant de $(v_1, \ldots, v_s)$, alors $(v_1, \ldots, v_s, w)$ est un système libre et la procédure donne, dans α, les composantes de $\bar{w}$ sur ce système.

On calcule w en déterminant le vecteur α vérifiant :

$$w=\bar{w}-\sum_{i=1}^{s} \alpha[i]\, v_i,$$

de façon que le système $(v_1, \ldots, v_s, w)$ soit triangularisé.

En fait, la suite $(v_1, \ldots, v_s)$ est triangularisée à une permutation des composantes près; cette permutation est représentée par un vecteur p :

– à chaque vecteur v_i, $1 \leqq i \leqq s$, est associé un indice de composante $p[i]$ tel que :

$$v_i[p[i]] \neq 0,$$

$$v_j[p[i]] = 0, \qquad i < j \leqq s.$$

Ainsi, pour un vecteur $\bar{w}$ de l'entrée de la procédure, on calcule $\alpha[i]$ de sorte que la p [i]-ième composante de w soit nulle et ceci pour tout i inférieure à s.

La procédure est décrite en pidgin-algol dans la figure 2; dans la liste des paramètres, la variable w décrit à la fois le vecteur d'entrée $\bar{w}$ et le vecteur de sortie w.

```
procédure DEP (s, w, α)
début
   pour i ← 1 à s faire
      début
         α[i] ← w[p[i]] / v_i[p[i]];
         w ← w − α[i] v_i;
      fin
   si (w≠0) alors
      début
         α[s+1] ← 1;
         p[s+1] ← 1re composante non nulle de w;
      fin
fin
```

Figure 2

3. Analyse et complexité

LEMME 1 : *La procédure STAND détermine une base* $\mathscr{V} = \{v_1, \ldots, v_m\}$ *de* $\mathscr{F}$.

Preuve : Par construction, les vecteurs v_i de $\mathscr{V}$ sont indépendants et il reste à montrer que $\lambda . \mu A^*$, système générateur de $\mathscr{F}$, est engendré par $\mathscr{V}$.

Pour cela, utilisant le fait que λ appartient à $\mathscr{V}$, nous montrons, par induction sur la longueur des mots de A^*, que si $v \in \mathscr{V}$ et $f \in A^*$, $v . \mu f$ est combinaison linéaire des vecteurs de $\mathscr{V}$:

– ceci est vrai pour ε;

– supposons que $f = ag$, $a \in A$.

Alors $v . \mu f = (v . \mu a) . \mu g$. Par construction $v . \mu a$ est combinaison linéaire des vecteurs de $\mathscr{V}$, et l'hypothèse d'induction appliquée à g donne le résultat. ∎

LEMME 2 : *La procédure STAND détermine une représentation linéaire accessible* (λ', μ', ρ').

Preuve : Considérons la matrice χ dont la r-ième ligne est le vecteur v_r (r-ième vecteur de $\mathscr{V}$). Par construction de $\mu' a$ (étape 5 dans la figure 1), $(\mu' a)_r$ est le vecteur des composantes de $v_r . \mu a$ sur $\mathscr{V}$; on a alors en notation matricielle :

$$v_r . \mu a = (\mu' a)_r . \chi, \qquad 1 \leqq r \leqq s,$$

ou encore :

$$\chi . \mu a = \mu' a . \chi.$$

Les vecteurs λ' et ρ' calculés dans l'algorithme (étapes 6 et 7 dans la figure 1) vérifient :

$$\lambda = \lambda' . \chi,$$
$$\rho' = \chi . \rho$$

et on en déduit pour a appartenant à A :

$$\lambda . \mu a . \rho = \lambda' . \chi . \mu a . \rho = \lambda' . \mu' a . \chi . \rho = \lambda' . \mu' a . \rho',$$

ce qui montre que (λ', μ', ρ') est une représentation linéaire. D'autre part, elle est accessible puisque sa dimension est celle de $\mathscr{V}$, base de $\mathscr{F}$ d'après le lemme 1, et que $\lambda' . \mu' A^*$ engendre $\mathscr{F}$. ■

D'après le théorème 2, la représentation (λ', μ', ρ') est isomorphe à une représentation standard, c'est-à-dire que la dimension de l'espace $\mathscr{E}'$ (engendré par $\mu' A^* . \rho'$) est égale au rang de la matrice de Hankel de la série. Pour obtenir une représentation standard il suffit alors, à partir de (λ', μ', ρ'), de déterminer une base de $\mathscr{E}'$ et d'exprimer cette représentation relativement à la base trouvée; ceci revient à calculer une représentation coaccessible. Cette seconde partie de l'algorithme est symétrique de la première par échange des lignes et colonnes. Nous appellerons STANDMIR la procédure qui la réalise.

THÉORÈME 3 : *L'appel successif des procédures STAND et STANDMIR détermine une représentation standard d'une série rationnelle sur un alphabet à m lettres à partir d'une représentation linéaire de dimension n avec une complexité en temps en $O(mn^3)$.*

Preuve : L'obtention d'une représentation standard à la fin de l'algorithme découle de la remarque précédente et du lemme 2.

Nous évaluons la complexité de l'algorithme représenté à la figure 1 en comptant le nombre d'opérations arithmétiques.

Chaque exécution de la procédure DEP appelée à l'étape 4 nécessite $(2n+1)s$ opérations arithmétiques, ce qui est majoré par $(2n+1)n$. La détermination de w

à l'étape 3 coûte $n(2n-1)$ opérations arithmétiques, et la complexité de la boucle 2 est alors majorée par 4 mn^2. Cette boucle est exécutée une fois pour chaque vecteur de la base cherchée ce qui donne une majoration de 4 mn^3 pour la boucle 1. L'étape 6 nécessite au plus $n(2n-1)$ opérations. Ainsi la complexité de l'algorithme de standardisation, en prenant les opérations arithmétiques comme unité de temps, est donc en $O(mn^3)$. ■

BILIOGRAPHIE

1. A. Aho, J. Hopcroft et J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974, p. 33-39.
2. J. Berstel, *Séries rationnelles*, in Séries formelles, J. Berstel, éd., E.N.S.T.A., 1978, p. 5-22.
3. A. Cardon et M. Crochemore, *Thèse de 3e cycle*, Laboratoire d'Informatique, Université de Rouen, 1978.
4. S. Eilenberg, *Automata, Languages and Machines*, vol. A, New York Academic Press, 1974.
5. M. Fliess, *Un outil algébrique : Les séries formelles non commutatives*, Rapport n° 139, I.R.I.A., 1975.
6. G. Jacob, *Langages, automates, séries formelles*, Publication 107, Université des Sciences et Techniques de Lille, 1978.
7. A. Salomaa et M. Soittola, *Automata Theoretic Aspects of Formal Power Series*, Springer-Verlag, 1978.
8. M. P. Schützenberger, *On the Definition of a Family of Automata*, Information and Control, vol. 4, 1961, p. 245-270.