MACHINES TURING et T-SCHÉMAS

PAR

C. CAZACU

Dans cette Note, nous introduisons la notion de *T-schéma* qui est équivalente à la notion de machine Turing. Les T-schémas représentent un appareil commode pour traiter certains problèmes de la théorie des algorithmes. Elles permettent aussi la mise en évidence de la structure interne des programmes des machines Turing. Selon notre opinion, les T-schémas sont des schémas d'algorithmes, dans le sens attribué à cette notion dans la programmation des calculateurs électroniques, dont les composantes sont des opérateurs très simples, qui réalisent les mêmes transformations que les instructions des programmes des machines Turing.

1. Soit $\mathfrak{S}$ une classe de machines Turing avec les alphabets des états externes égaux à l'ensemble $\mathfrak{A} = \{a_0, a_1, \ldots, a_p\}$, $(p \geqslant 2)$ et les alphabets des états internes égaux à des segments initiaux de $q_0, q_1, q_2, \ldots$

Comme d'habitude, a_0, q_1 et q_0 seront respectivement la celulle vide sur la bande d'une machine Turing, l'état interne initial et l'état interne final.

Le programme $\mathfrak{P}(M)$ d'une telle machine M contient quelques instructions de la forme $q_i\, a_j \rightarrow q_\alpha X$ où X est égal à S (déplacement à gauche), D (déplacement à droite) ou $a_\beta (a_\beta \in \mathfrak{A})$. Les instructions pour lesquelles $\alpha = 0$, sont appelées *instructions finales*. Un contenu $b_1 b_2 \ldots b_j \ldots b_t$ de la bande d'une machine Turing, un état interne q_i et une cellule visée b_j, forment une *configuration* désignée par $b_1 b_2 \ldots q_i b_j \ldots b_t$. Si la machine Turing M transforme la configuration $U_1 q_1 b_1 V_1$ en $U_0 q_0 b_0 V_0$, on représente cette situation par $M : U_1 \dot{b}_1 V_1 \Rightarrow U_0 \dot{b}_0 V_0$ ou par $M(U_0 \dot{b}_0 V_0) = U_1 \dot{b}_1 V_1$.

2. Considérons un graphe orienté $G = (\mathfrak{A}, \mathfrak{U})$ où $\mathfrak{A}$ est un ensemble de machines Turing (de la classe $\mathfrak{S}$) et $\mathfrak{U}$ un ensemble de paires de machines du $\mathfrak{A}$. Nous attachons à chaque paire $(M, N) \in \mathfrak{U}$ un ensemble f_0

de manière que $\subset_{\neq} f_0 \subset f(M)$, où $f(M)$ est l'ensemble des instructions finales de la machine M. Dans ce cas, nous disons que les machines M, N, sont *f_0-connectées* et nous notons cela par $M \xrightarrow{f_0} N$. Soit $M \in \mathfrak{A}$, $\{M_1, M_2, \ldots, \ldots, M_n\} = G(M)$ et $M \xrightarrow{f_i} M_i$ (pour $i = 1, 2, \ldots, n$). On dit que la machine M est *correctement connectée* aux machines $M_1, M_2, \ldots, M_n$, si les ensembles $f_1, f_2, \ldots, f_n$ sont disjoints deux par deux. On appelle la machine M de $\mathfrak{A}$ — *machine finale du graphe* G si $\bigcup_{i=1}^{n} f_i \subset_{\neq} f(M)$. Pour toutes les machines finales M, les instructions du $f(M) - \bigcup_{i=1}^{n} f_i$ seront nommées *instructions finales du graphe* G. Soit ensuite une machine fixée M_0 de $\mathfrak{A}$ qu'on appelle *machine initiale du graphe* G. Une suite de machines $M_0, M_1, \ldots, \ldots, M_k$ sera nommée *chaîne maximale du graphe* G si $(M_i, M_{i+1}) \in \mathfrak{U}$ (pour $i = 0, 1, \ldots, k-1$) et si M_k est une machine finale. On appelle le graphe G, *graphe correct*, si toute machine de $\mathfrak{A}$ appartient au moins à une chaîne maximale. Enfin, le graphe correct G, où toutes les machines sont correctement connectées, sera nommé *T-schéma* (schéma Turing). Les machines de $\mathfrak{A}$ seront nommées *composantes* du T-schéma G.

Soit $U_1 q_1 b_1 V_1$ une configuration sur la bande de la machine initiale M_0 du T-schéma G. On dit que le T-schéma G transforme (finalement) la configuration $U_1 q_1 b_1 V_1$ dans une configuration $U_0 q_0 b_0 V_0$ s'il y a une chaîne maximale $M_0, M_1, \ldots, M_k$ de manière que

a) $M_i(U_{i+1} q_1 b_{i+1} V_{i+1}) = U_{i+2} q_0 b_{i+2} V_{i+2}$ (pour $i = 0, 1, 2, \ldots, k$).

b) $U_{k+2} = U_0$, $b_{k+2} = b_0$, $V_{k+2} = V_0$.

c) L'instruction finale utilisée dans la transformation $M : U_{k+1} q_1 b_{k+1} V_{k+1} \Rightarrow U_0 q_0 b_0 V_0$ est une instruction finale du graphe G.

Dans ce cas nous allons utiliser aussi les notations $G : U_1 \dot{b}_1 V_1 \Rightarrow U_0 \dot{b}_0 V_0$ ou $G(U_1 \dot{b}_1 V_1) = U_0 \dot{b}_0 V_0$. Si une chaîne maximale, dotée des propriétés énoncées plus haut, n'existe pas, alors on dit que $G(U_1 \dot{b}_1 V_1)$ n'est pas déterminée.

3. La machine Turing M est *équivalente* au T-schéma G, si pour tout $U_1 b_1 V_1$, on a $M(U_1 \dot{b}_1 V_1) = G(U_1 \dot{b}_1 V_1)$. L'équivalence entre les notions de machine et de schéma Turing résulte de deux lemmes suivants.

Lemme 1. *Toute machine Turing M est équivalente à un T-schéma.*

Démonstration. La machine M sera équivalente au schéma $S = (\mathfrak{A}, \mathfrak{U})$ où $\mathfrak{A} = \{M\}$, $\mathfrak{U} = \varnothing$, M étant en même temps machine initiale et machine finale du T-schéma S.

Lemme 2. *Tout T-schéma S est équivalent à une machine Turing.*

Démonstration. Soit le T-schéma $S = (\mathfrak{A}, \mathfrak{U})$ où $\mathfrak{A} = \{M_1, M_2, \ldots, \ldots, M_n\}$. On remplace dans les programmes $\mathfrak{P}(M_i)$ (pour $i = 1, 2, \ldots, n$) les lettres $q_0, q_1, \ldots$ par d'autres lettres dans l'ordre suivant :

a) On remplace les lettres q_j de $\mathfrak{P}(M_i)$ par $q_j^{(i)}$.

b) Si M_a est la machine initiale du schéma S, on remplace $q_1^{(a)}$ par q_1. On remplace ensuite les lettres $q_0^{(i)}$ par q là où $q_0^{(i)}$ fait partie des instructions finales du schéma S et par $q_1^{(j)}$ si $M_i \xrightarrow{f} M_j$ et $q_0^{(i)}$ entrent dans les instructions de f'.

c) Enfin, on remplace toute lettre $q_j^{(i)}$ par une lettre de la suite $q_2, q_3, \ldots$ de manière que la correspondance $q_j^{(i)} \to q_r$ soit une correspondance biunivoque sur un segment initial de la suite donnée.

La réunion des instructions ainsi obtenues formera le programme $\mathfrak{P}(M)$ d'une machine Turing M qui est équivalente au T-schéma donné S.

4. Si $S_1, S_2, \ldots, S_n$ sont quelques T-schémas, on peut former un nouveau T-schéma en connectant quelques machines finales des T-schémas donnés à d'autres composantes de ces schémas de manière que le nouveau graphe obtenu soit aussi un graphe qui respecte les conditions données dans la définition du T-schéma.

Un T-schéma de la forme $S_1 \to S_2$, où toutes les machines finales du T-schéma S_1, ont été connectées (par toutes les instructions finales du T-schéma S_1) à la machine initiale du T-schéma S_2, sera nommé *produit des T-schémas* S_1, S_2 et sera noté par $S_1 S_2$. Evidemment, le produit des T-schémas est associatif. Si $S_1 = S_2 = \ldots = S_n = S$, le produit $S_1 S_2 \ldots S_n$ sera noté par S^n.

Si $S : U_1 \dot{b}_1 V_1 \Rightarrow U_0 \dot{b}_0 V_0$ et le T-schéma S ne change pas le contenu des cellules à gauche de U_1 (à gauche de U_1 et à droite de V_1) nous disons que S transforme d'une manière correcte (strictement correcte) la configuration $U_1 q_1 b_1 V_1$ dans la configuration $U_0 q_0 b_0 V_0$. Pour ces notions, on utilise les notations

$$S : U_1 \dot{b}_1 V_1 \Longrightarrow U_0 \dot{b}_0 V_0 \qquad \text{et} \qquad S : U_0 \dot{b}_0 V_1 \Longrightarrow U_0 \dot{b}_0 V_0$$

respectivement.

Quelques composantes d'un T-schéma S peuvent être égales entre elles. Les machines $E_1, E_2, \ldots, E_s$ différentes entre elles deux à deux, seront nommées *éléments* du T-schéma S, si toute E_i entre, au moins une fois, en S comme composante et inversement, toute composante se trouve dans la suite $E_1, E_2, \ldots, E_s$.

Deux T-schémas S_1 et S_2 s'appellent *équivalentes* si pour toute configuration $U_1 q_1 b_1 V_1$, on a $S_1(U_1 \dot{b}_1 V_1) = S_2(U_2 \dot{b}_2 V_2)$. En ce cas on écrit $S_1 \cong S_2$.

Nous allons démontrer ensuite, que tout T-schéma est équivalent à un T-schéma construit de trois éléments.

5. Si M est une machine Turing, nous allons noter par M^{-1} la machine obtenue, en remplaçant partout dans le programme de la machine M les lettres D par S et inversement.

Soient les machines T, D, D^{-1}, où

$$\mathcal{S}(T) = \begin{cases} q_1 a_0 \to q_0 a_1, \\ q_1 a_1 \to q_0 a_2, \\ \cdots\cdots \\ q_1 a_{p-1} \to q_0 a_p, \\ q_1 a_p \to q_0 a_0, \end{cases} \qquad \mathcal{S}(D) = \begin{cases} q_1 a_0 \to q_0 D, \\ q_1 a_1 \to q_0 D, \\ \cdots\cdots \\ q_1 a_{p-1} \to q_0 D, \\ q_1 a_p \to q_0 D. \end{cases}$$

Evidemment, $T^{p+1} \simeq D^{-1} D \simeq D D^{-1} \simeq I$ où $I(U_1 \dot{b}_1 V_1) = U_1 \dot{b}_1 V_1$ pour toute configuration $U_1 q_1 b_1 V_1$.

Théorème 1. *Tout T-schéma est équivalent à un T-schéma formé par les éléments T, D, D^{-1}.*

Démonstration. Il suffit de démontrer, conformément au lemme 2, que toute machine Turing est équivalente à un T-schéma formé par les éléments T, D, D^{-1}. Soit donc M une machine Turing quelconque. Supposons que dans le programme de cette machine apparaissent seulement les états internes $q_0, q_1, \ldots, q_s$. Divisons les instructions de $\mathcal{S}(M)$ en s classes $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_s$, en introduisant dans la classe $\mathcal{E}_i$ toutes les instructions de la forme $q_i a_j \to \omega_{ij}$ $(j = 0, 1, \ldots, p)$. Pour chaque classe $\mathcal{E}_i$, nous construisons le T-schéma S_i de la figure 1, où $I_i = T^{p+1}$, la connexion $T \xrightarrow{i} Y$ se fait à l'aide de l'instruction $q_1 a_{i-1} \to q_0 a_i$ et

$$X_{ij} = \begin{cases} D & \text{si } \omega_{ij} = q_\alpha D, \\ D^{-1} & \text{si } \omega_{ij} = q_\alpha S, \\ T^{p+1-j+\beta} & \text{si } \omega_{ij} = q_\alpha a_\beta. \end{cases}$$

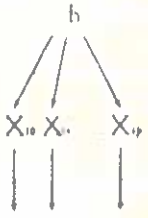

Fig. 1

A l'aide des T-schémas S_i ainsi construites, nous formerons ensuite un seul T-schéma S, en connectant toute machine X_{ij} à I_α si $\omega_{ij} = q_\alpha \delta$ $(\alpha \geqslant 1)$. Le T-schéma S où I_1 est la machine initiale et X_{ij} pour lesquels $\alpha = 0$, sont des machines finales, sera équivalent à la machine donnée M.

Exemple. Soit $\mathcal{A} = \{0, 1, 2\}$ et la machine M avec le programme

$$\mathcal{S}(M) = \begin{cases} q_1 0 \to q_0 D, & q_2 0 \to q_1 S, & q_3 0 \to q_1 0, \\ q_1 1 \to q_2 2, & q_2 1 \to q_0 1, & q_3 1 \to q_0 S, \\ q_1 2 \to q_0 1, & q_2 2 \to q_3 D, & q_3 2 \to q_0 1. \end{cases}$$

Cette machine sera équivalente au T-schéma de la figure 2.

Un T-schéma S, construit par le procédé exposé plus haut, met en évidence la structure interne du programme $\mathcal{S}(M)$ et le moyen par lequel une machine Turing réalise ces transformations.

5. Nous allons montrer, comme une application directe, que toute fonction partiellement récursive peut être correctement calculée par un T-schéma.

Considérons dans ce but des machines Turing pour lesquelles $\mathcal{A} = \{0, 1\}$. Si M est une telle machine, nous allons noter par $\overline{M}$ la machine dont le programme s'obtient en remplaçant dans le programme de la machine, les états externes 0 par les états externes 1 et inversement.

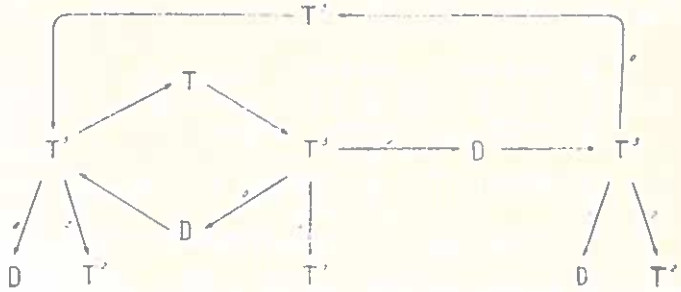

Fig. 2

Soit $f(x_1, x_2, \ldots, x_n)$ une fonction arithmétique partiellement définie et soit $\mathcal{D}$ le domaine de définition de cette fonction. Notons ensuite par 1^n un groupe de n unités. On dit que le T-schéma S *calcule correctement* la fonction arithmétique f si

a) $S : \dot{0} 1^{x_1+1} 0 1^{x_2+1} 0 \ldots 0 1^{x_n+1} 0 \Longrightarrow \dot{0} 1^{f(x_1, x_2, \ldots, x_n)+1} 0$

pour $(x_1, x_2, \ldots, x_n) \in \mathcal{D}$ et

b) le résultat de l'activité du T-schéma S n'est pas déterminé pour la configuration $q_1 0 1^{x_1+1} 0 1^{x_2+1} 0 \ldots 0 1^{x_n+1} 0$ si $(x_1, x_2, \ldots, x_n) \notin \mathcal{D}$.

Théorème 2. *Toute fonction partiellement récursive peut être correctement calculée par un T-schéma.*

Démonstration. Il résulte de la définition de la fonction partiellement récursive, qu'il suffit de démontrer que les fonctions élémentaires $o^{(n)}(x_1, x_2, \ldots, x_n) = 0$, $i_p^{(n)}(x_1, x_2, \ldots, x_n) = x_p$ $(1 \leqslant p \leqslant n)$ et $s(x) = x + 1$ sont correctement calculables à l'aide des T-schémas et les opérateurs de superposition, de récursion primitive et de minimisation conservent cette propriété.

Soient les T-schémas $p \simeq T^2$, $A \simeq \overset{1}{\overset{\downarrow}{D}}p$, $B \simeq \overset{\downarrow}{D}p\underset{\downarrow 0}{T}$. Evidemment,

$$A : \dot{\varepsilon} 1^n 0 \Longrightarrow \varepsilon 1^n \dot{0}, \qquad B : \dot{\varepsilon} 1^n 0 \Longrightarrow \varepsilon 0^n \dot{0} \qquad (\varepsilon = 0, 1 \,;\; n \geqslant 0).$$

A l'aide de ces T-schémas on construit les T-schémas

$$R \simeq D p D^{-1} B^{-1} \overset{\downarrow}{D} T A T D p A^{-1}, \qquad Q \simeq D p A^2 T \overset{\downarrow}{A^{-1}} T D^{-1} p A A T A^{-1},$$

$$\underset{\downarrow 0}{B^{-1} A^{-1} D T} \qquad \underset{\downarrow 0}{B^{-1} D T} \qquad \underset{\downarrow 0}{B D^{-1} T}$$

$$Q_n \simeq ((A R)^{n-1} Q (A^{-1} R)^{n-1})^n$$

On peut voir que

$$R : 01^x \dot{0} 1^y 0 \Longrightarrow 01^x \dot{0} 1^y 0 \qquad (x, y \geq 0),$$

$$Q : \dot{0} 1^x 00^y \Longrightarrow 01^x \dot{0} 1^y \qquad (x \geq 0),$$

$$Q_n : \dot{0} 1^{x_1} 01^{x_2} 0 \ldots 01^{x_n} 00^{y_1} 00^{y_2} 0 \ldots 00^{y_n} \Longrightarrow$$

$$\Longrightarrow 01^{y_1} 01^{y_2} 0 \ldots 01^{y_n} \dot{0} 1^{x_1} 01^{x_2} 0 \ldots 01^{x_n}.$$

Il résulte, en notant par $O^{(n)}$, $I_p^{(n)}$ et S les T-schémas qui calculent les fonctions élémentaires, que

$$O^{(n)} \simeq A^n B^{-n} D T D^{-1}, \quad I_p^{(n)} \simeq A^n B^{n-p} (A^{-1} R A B^{-1})^{p-1} A^{-1}, \quad S \simeq A T A^{-1}.$$

Si $H, G, G_1, G_2, \ldots, G_s$ sont les T-schémas qui calculent correctement les fonctions $h, g, g_1, g_2, \ldots, g_s$, alors la fonction f obtenue par superposition, récursion primitive ou minimisation conformément aux formules

1) $$f(x_1, x_2, \ldots, x_n) = \\ = h(g_1(x_1, x_2, \ldots, x_n), g_2(x_1, x_2, \ldots, x_n), \ldots, g_s(x_1, x_2, \ldots, x_n)),$$

2a) $$\begin{cases} f(0) = a, \\ f(y+1) = h(y, f(y)), \end{cases}$$

2b) $$\begin{cases} f(x_1, x_2, \ldots, x_n, 0) = g(x_1, x_2, \ldots, x_n), \\ f(x_1, x_2, \ldots, x_n, y+1) = h(x_1, x_2, \ldots, x_n, y, f(x_1, x_2, \ldots, x_n)), \end{cases}$$

3) $$f(x_1, x_2, \ldots, x_n) = \mu_y (g(x_1, x_2, \ldots, x_{n-1}, y) = x_n),$$

peut être correctement calculée respectivement par les T-schémas suivants :

(1) $$F \simeq (\prod_{i=1}^{s} (Q_n G_i (R A^{-1})^n A)) A^n B^{-n} A^{-n} H.$$

(2a) $$F \simeq A T D^{-1} T A^2 (T D)^{a+1} A^{-2} \overset{\downarrow}{D^{-1}} p D Q_2 H R A B^{-1} A^{-2} T D^{-1} T.$$

$$\underset{\downarrow 0}{D T A R A B^{-1} A^{-1}}$$

(2b) $$F \simeq A^{n-1} D^{-1} T (A^{-1} R)^n \times$$

$$\times Q_n G A^2 T D^{-1} R A^{-2} \overset{\downarrow}{D^{-1}} p D Q_{n+2} H R A B^{-1} A^{-2} (T D^{-1} T A^{-1})^{n+1} A,$$

$$\underset{\downarrow 0}{(A T)^{n+1} A R A B^{-1} A^{-1}}$$

(3) $$F \simeq A^{n-1} \overset{\downarrow}{T} D A^{-(n+1)} Q_{(n+1)} A^n (A^{-1} R)^{n-1} G q \rightarrow (A^{-1} R)^n A^n B^{-n} A^{-1},$$

où

$$q \simeq D^2 p D^{-4} p (D T)^2 A (D^{-1} T)^2 A^{-1}$$

$$\underset{\downarrow 0}{D^{-4} p} \xrightarrow{1} \underset{\downarrow 0}{A^2 B^{-2}} \rightarrow$$

$$\underset{\downarrow}{A^2 B^{-2}} \xrightarrow{+} \rightarrow$$

et $q : 01^x \dot{0} 1^x 0 \Longrightarrow \dot{0} 00 \ldots 00$.

Conséquences. 1. Toute fonction partiellement récursive est correctement calculable par une machine Turing avec l'alphabet des états externes égal à $\{0, 1\}$. (Théorème 2, lemme 2).

2. Toute fonction arithmétique obtenue par un nombre fini de superpositions à l'aide des fonctions d'un ensemble donné M, peut être correctement calculée par un T-schéma, obtenu sous la forme d'un produit où entrent les T-schémas qui calculent correctement les fonctions de M et les T-schémas A, A^{-1}, B, B^{-1}, R et Q.

3. Il y a les T-schémas X_i, Y_i, Z_i, p_i, de manière que la fonction f obtenue à l'aide d'un des opérateurs 1), 2a), 2b, 3), peut être correctement calculée respectivement par un T-schéma de la forme

1) $$Q_n G_1 X_1 G_2 X_1 \ldots X_1 G_s Y_1 H,$$

2a) $$X_2 \overset{\downarrow}{p_2} H Y_2,$$

2b) $$X_3 G Y_3 \overset{\downarrow}{p_3} H Z_3,$$

3) $$X_4 \overset{\downarrow}{Y_4} G p_4.$$

BIBLIOGRAPHIE

1. Kleene S. C. — *Introduction to metamathematics*. D. Van Nostrand Company, Inc. New York, Toronto, 1952.
2. Мальцев А. И. — *Алгоритмы и рекурсивные функции*. Изд. „Наука", Москва 1965.
3. Maruşciac I. — *Teoria algoritmilor*. Ed. did. şi ped. 1966.
4. Заславский И. Д. — *Граф-схемы с памятью*. Труды мат. инст. им. В. А. Стеклова, том LXXII, 1964.

MAŞINI TURING ŞI T-SCHEME

Rezumat

Se introduce noţiunea de T-schemă, care este echivalentă cu noţiunea de maşină Turing. T-schemele reprezintă un aparat mai comod pentru tratarea unor chestiuni din teoria algoritmilor. În acelaşi timp, T-schemele pun în evidenţă structura internă a programelor maşinilor Turing. Se arată că orice funcţie parţial recursivă poate fi calculată corect (în sens A. I. Malţev) de o T-schemă pentru care alfabetul stărilor interne este format din două simboluri.

SUR LES ALGÈBRES MATRICIELLES-POLYNÔMIALES À SCALAIRES RÉELS

PAR

I. CREANGĂ et I. ENESCU

Communication présentée à la session scientifique de l'Université „Al. I. Cuza" du 25–27 octobre 1968

1. Le but de cette Note est de démontrer que l'ensemble de n-uples ordonnées de matrices réelles du type (n, n) peut être organisé comme une algèbre d'ordre n^3 sur le corps réel, dont le produit est défini d'une manière similaire à celle spécifique aux algèbres polynômiales. Nous allons aussi construire effectivement ce type d'algèbres dont les propriétés de structure seront également étudiées. Nous utiliserons les notations suivantes:

$\mathfrak{M}$ — l'algèbre complète de matrices d'ordre n^2 à scalaires réels; A — l'ensemble de n-uples ordonnées d'éléments de $\mathfrak{M}$; I — la matrice unité de $\mathfrak{M}$; R et K, respectivement le corps réel et le corps complexe; $P(\lambda)$ — un polynôme de degré n à coefficients réels; H — l'algèbre engendrée par $P(\lambda)$; e_{ij} une base pour $\mathfrak{M}$, dont la loi de multiplication est

$$e_{ij}\, e_{kl} = \begin{cases} e_{il} & \text{pour} \quad j = k, \\ 0 & \text{pour} \quad j \neq k. \end{cases} \tag{1}$$

2. Soit $x = (A_1, A_2, \ldots, A_n)$ et $y = (B_1, B_2, \ldots, B_n)$ deux éléments de A. Nous définissons la somme des éléments x et y et le produit d'un élément de A par les scalaires de R, respectivement par

$$x + y = (A_1 + B_1,\ A_2 + B_2,\ \ldots,\ A_n + B_n), \tag{2}$$

$$\lambda x = (\lambda A_1,\ \lambda A_2,\ \ldots,\ \lambda A_n). \tag{3}$$

On constate facilement que, doté de ces opérations, l'ensemble A est un espace linéaire de dimension n^3 sur R, dont une base est donnée par les éléments