

Chemin algébrique et réseau systolique.

Ce devoir a été posé au troisième concours d'entrée 2003 de l'ENS Cachan. Les parties I et II sont indépendantes. La partie III peut être traitée en admettant les résultats de la partie I. Il est à rendre, en cours ou en TD, le 24 novembre.

Définitions et notations

- Un *graphe* $G = (S, A)$ est constitué d'un ensemble fini de sommets S et d'un ensemble d'arcs $A \subseteq S \times S$. On considère donc les graphes orientés sans arcs multiples mais pouvant contenir des arcs bouclant sur un sommet.
- Un *chemin* c dans un graphe est une suite de sommets

$$c = (s_0, s_1, \dots, s_n)$$

telle que $n \in \mathbb{N}$ et pour tout $i \in \mathbb{N}$ tel que $i \leq n - 1$, $(s_i, s_{i+1}) \in A$. Le chemin c est appelé un *chemin de s_0 à s_n* . Si $n = 0$ le chemin ne contient pas d'arc : on parle alors de *chemin vide* (réduit au sommet s_0).

- Un *graphe pondéré* (S, A, ω) est un graphe (S, A) muni d'une fonction de pondération $\omega : A \rightarrow H$ où H est un ensemble quelconque. $\omega(a)$ est appelé *poide* de l'arc a . Si (H, \otimes) est un monoïde (c'est-à-dire un semi-groupe muni d'un élément neutre), on peut étendre ω à l'ensemble des chemins :

$$\omega(s_0, s_1, \dots, s_n) = \omega((s_0, s_1)) \otimes \omega((s_1, s_2)) \otimes \dots \otimes \omega((s_{n-1}, s_n))$$

Le poids d'un chemin vide est évidemment $\mathbf{1}$, élément neutre du monoïde.

- Un *semi-anneau* (H, \oplus, \otimes) avec zéro $\mathbf{0}$ et unité $\mathbf{1}$ est une structure munie des deux opérations \oplus et \otimes et respectant les quatre axiomes suivants.

- (S₁) (H, \oplus) est un monoïde commutatif avec $\mathbf{0}$ pour élément neutre,
- (S₂) (H, \otimes) est un monoïde avec $\mathbf{1}$ pour élément neutre,
- (S₃) \otimes est distributif par rapport à \oplus ($\forall a, b, c \in H, a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$ et $(b \oplus c) \otimes a = b \otimes a \oplus c \otimes a$),
- (S₄) $\mathbf{0}$ est absorbant pour \otimes ($\forall a \in H, a \otimes \mathbf{0} = \mathbf{0} \otimes a = \mathbf{0}$).

- *Semi-anneaux de matrices* : si (H, \oplus, \otimes) est un semi-anneau, on peut munir l'ensemble $H^{n \times n}$ des matrices de taille $n \times n$ ($n \geq 1$) d'une structure de semi-anneau, comme en algèbre linéaire classique. Le zéro est la matrice qui contient uniquement la valeur $\mathbf{0}$, et l'unité est la matrice qui contient des $\mathbf{1}$ sur la diagonale et des $\mathbf{0}$ partout ailleurs.

I Problème du chemin algébrique

Le problème du chemin algébrique peut être formulé de la façon suivante : étant donné un graphe orienté pondéré $G = (S, A, \omega)$, avec $\omega : A \rightarrow H$, où H est un semi-anneau comme défini précédemment, trouver pour tout couple (i, j) de sommets la valeur $d_{i,j}$ définie par

$$d_{i,j} = \bigoplus_{\substack{c \text{ chemin} \\ \text{de } i \text{ à } j \\ \text{dans } G}} \omega(c)$$

L'ensemble des chemins de i à j peut être infini. Cependant, pour certains semi-anneaux, il est possible de donner un sens à cette somme infinie, qui aura une valeur définie dans le semi-anneau. Pour pouvoir parler de sommes infinies de façon consistante, on supposera que les deux propriétés suivantes sont vérifiées.

(S₅) Soient I et J deux ensembles dénombrables, alors l'égalité

$$\bigoplus_{(i,j) \in I \times J} a_i \otimes b_j = \bigoplus_{i \in I} a_i \otimes \bigoplus_{j \in J} b_j$$

est vérifiée dès que les deux sommes de la partie droite sont définies.

(S₆) Soient I, J et I_j des ensembles dénombrables tels que $\{I_j \mid j \in J\}$ est une partition de I , alors l'égalité

$$\bigoplus_{i \in I} a_i = \bigoplus_{j \in J} \bigoplus_{i \in I_j} a_i$$

est vérifiée dès que la somme en partie gauche est définie.

Nous aurons besoin de calculer des sommes infinies de la forme suivante, pour lesquelles nous introduisons la notation

$$c^* = \bigoplus_{i \geq 0} c^i = \mathbf{1} \oplus c \oplus (c \otimes c) \oplus (c \otimes c \otimes c) \oplus \dots$$

Applications du problème du chemin algébrique

▷ **Question 1** Donner une structure de semi-anneau pour laquelle le problème du chemin algébrique se ramène à calculer la relation d'accessibilité dans un graphe orienté.

Réponse □

▷ **Question 2** Donner une structure de semi-anneau pour laquelle le problème du chemin algébrique se ramène au problème du calcul des plus courts chemins dans un graphe orienté.

Réponse □

Représentation matricielle À un graphe pondéré (S, A, ω) on peut associer une matrice de taille $|S| \times |S|$

$$M = (m_{ij}), \text{ où } m_{ij} = \begin{cases} \omega((i, j)) & \text{si } (i, j) \in A \\ \mathbf{0} & \text{si } (i, j) \notin A \end{cases}$$

▷ **Question 3** Montrer que pour tout entier p ,

$$M^p = (m_{ij}^{(p)}), \text{ où } m_{ij}^{(p)} = \bigoplus_{\substack{c \text{ chemin de } i \text{ à } j \\ \text{contenant exactement } p \text{ arcs}}} \omega(c)$$

Réponse □

▷ **Question 4** Montrer que calculer l'inverse d'une matrice à coefficients dans \mathbb{R} peut se ramener au problème du chemin algébrique, pour un semi-anneau que l'on précisera. Les problèmes liés à l'existence d'une solution devront être soigneusement pris en compte.

Réponse □

Version algorithmique On se propose de trouver un algorithme qui résout le problème du chemin algébrique (sous l'hypothèse qu'une solution existe). Les sommets intermédiaires d'un chemin sont tous les sommets de ce chemin sauf le sommet initial et le sommet final. Soient i et j deux sommets, et k un entier tel que $0 \leq k \leq n$. On note $\mathcal{C}_{ij}^{(k)}$ l'ensemble de tous les chemins de i à j dont tous les sommets intermédiaires x vérifient $1 \leq x \leq k$. On note

$$p_{ij}^{(k)} = \bigoplus_{c \in \mathcal{C}_{ij}^{(k)}} \omega(c)$$

Le chemin constitué d'un seul arc de i à j est contenu dans $\mathcal{C}_{ij}^{(0)}$ (et donc dans tous les $\mathcal{C}_{ij}^{(k)}$). Par convention, le chemin vide qui commence et finit au nœud i n'est pas contenu dans $\mathcal{C}_{ii}^{(i-1)}$. En revanche, il est contenu dans $\mathcal{C}_{ii}^{(i)}$.

▷ **Question 5** Donner les formules de récurrence permettant de calculer les $p_{ij}^{(k)}$, en considérant les valeurs respectives de i, j et k et en partitionnant $C_{ij}^{(k)}$ de façon adéquate. Réponse □

▷ **Question 6** Montrer précisément comment et pourquoi à partir de ces formules de récurrence on peut se ramener à l'algorithme donné en figure 1, constitué de trois phases, où N représente le nombre de sommets du graphe. Réponse □

```

PHASE 1 :
  pour i de 1 a N
  pour j de 1 a N
  faire
    pour k de 1 a min(i, j) - 1
       $p_{ij} := p_{ij} \oplus p_{ik} \otimes p_{kj}$  ;
    si  $i = j$  alors  $p_{ii} := p_{ii}^*$  ;
    si  $i > j$  alors  $p_{ij} := p_{ij} \otimes p_{jj}$  ;
  fait ;
PHASE 2 :
  pour i de 1 a N
  pour j de 1 a N
  faire
    si  $i < j$  alors  $p_{ij} := p_{ii} \otimes p_{ij}$  ;
    pour k de min(i, j) + 1 a max(i, j) - 1
       $p_{ij} := p_{ij} \oplus p_{ik} \otimes p_{kj}$  ;
    si  $i < j$  alors  $p_{ij} := p_{ij} \otimes p_{jj}$  ;
  fait ;
PHASE 3 :
  pour i de 1 a N
  pour j de 1 a N
  faire
    si  $i > j$  alors  $p_{ij} := p_{ii} \otimes p_{ij}$  ;
    pour k de max(i, j) + 1 a N
       $p_{ij} := p_{ij} \oplus p_{ik} \otimes p_{kj}$  ;
  fait ;

```

FIG. 1 – Algorithme en trois phases pour le chemin algébrique.

II Réseaux systoliques

Principe. Un réseau systolique est une architecture spécialisée qui permet d'implanter un algorithme effectuant un grand nombre de calculs simples de nature identique. Nous ne donnons pas ici de définition formelle d'un réseau systolique, mais les trois principales propriétés qui le caractérisent :

- **parallélisme massif** : un réseau systolique est constitué d'un grand nombre de cellules (processeurs) élémentaires ; une cellule peut effectuer quelques calculs simples et dispose de quelques registres ; les cellules peuvent être de plusieurs types, mais ce nombre de types différents reste très petit ;
- **localité** : les cellules sont reliées entre elles, mais uniquement de façon locale : chaque cellule ne peut être connectée qu'aux cellules voisines (au sens spatial du terme) ; seules les cellules situées à la frontière du réseau communiquent avec le monde extérieur (un processeur-hôte qui alimente le réseau en données d'entrée et récupère les résultats en sortie) ;

- **synchronisme** : les cellules évoluent en parallèle, sous le contrôle d'une horloge globale ; les calculs sont effectués simultanément dans tout le réseau ; à chaque cycle d'horloge, chaque cellule reçoit des données en provenance des cellules voisines (éventuellement complétées par des données de ses registres internes), effectue un calcul simple, puis transmet les résultats (toujours aux cellules voisines) un temps de cycle plus tard.

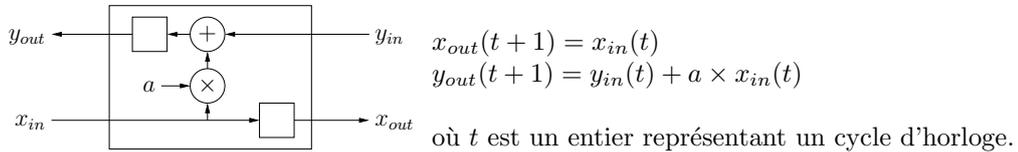
Fonctionnement général. Le fonctionnement d'un réseau systolique peut être vu comme celui d'un ensemble de données qui se déplacent dans le réseau, en étant modifiées par les cellules ou alors simplement transmises. Une même cellule peut présenter des comportements différents selon les temps d'horloge, grâce à des mécanismes de contrôle que nous n'aborderons pas ici.

Pour une cellule donnée, chaque cycle d'horloge ne correspond pas obligatoirement à une donnée valide : il peut y avoir un ou plusieurs cycles « à vide » entre le passage de deux données consécutives. Même si physiquement parlant il y a bien calcul ou transmission de données à chaque cycle, les données non significatives — car ne participant pas au calcul du résultat souhaité — ne sont pas prises en compte. Pour un cycle donné, une cellule sera dite *active* si elle participe au calcul ou à la transmission d'une donnée valide, et *passive* sinon.

Réseau pour la convolution non récursive. Notre but est d'implanter un algorithme permettant de résoudre le problème suivant : étant donnée une suite $(x_i)_{i \geq 1}$ de données réelles et une suite finie (a_1, \dots, a_k) de valeurs réelles fixées, calculer pour tout $i \geq k$ la valeur de

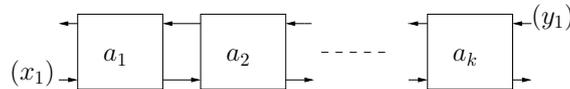
$$y_i = \sum_{j=1}^k a_j \times x_{i-j+1}$$

On propose un réseau systolique à une dimension, dont les cellules ont la forme suivante.



L'élément représenté par un rectangle est un registre : il permet de mémoriser une valeur et introduit un délai d'un cycle d'horloge entre son entrée et sa sortie. Le coefficient a correspond à l'un des a_j .

Le réseau proprement dit est constitué de la juxtaposition et de la connexion de k cellules de type identique (qui ne diffèrent que par la valeur de a). Les cellules sont numérotées de 1 à k (de la gauche vers la droite), et sont pourvues du coefficient a_j adéquat.



Les données $(x_i)_{i \geq 1}$ se déplacent de la gauche vers la droite, et les valeurs des $(y_i)_{i \geq k}$ en sens inverse, et avancent d'une cellule à chaque cycle d'horloge. Les données sont bien entendu introduites dans le réseau dans l'ordre croissant de leurs indices, et les y_i sont initialisés à 0 avant leur entrée.

▷ **Question 7** On suppose que l'instant $t = 1$ correspond au cycle d'horloge où la donnée x_1 entre dans la cellule 1. Quelle valeur de la suite (y) doit se trouver au même instant dans la même cellule ? Réponse □

▷ **Question 8** On se place à l'instant t où la donnée x_i arrive dans la cellule 1. Où doit alors se trouver la donnée x_{i-1} ? Quelle est la valeur de (y) correspondante ? Réponse □

▷ **Question 9** Pour p entier compris entre 1 et k , à quel instant la donnée x_i doit-elle se trouver dans la cellule p ? Même question pour y_i . Réponse □

▷ **Question 10** Sous les hypothèses induites par les réponses aux questions précédentes, quel est le débit (rapport entre le nombre de sorties valides et le nombre de cycles d'horloge) du réseau? Réponse □

▷ **Question 11** En supposant que les données sont introduites dans le réseau conformément aux hypothèses des questions précédentes, montrer que le réseau calcule effectivement le résultat souhaité. Réponse □

Réseau pour la convolution récursive. On veut maintenant calculer les valeurs $(y_i)_{i \geq k+1}$ définies par

$$y_i = \sum_{j=1}^k a_j \times y_{i-j}$$

où (y_1, \dots, y_k) sont des valeurs initiales données.

▷ **Question 12** En supposant que les cellules disposent d'un mécanisme qui leur permet à certains instants de transmettre des données de droite à gauche sans les modifier, proposer une modification du réseau précédent qui permet de résoudre ce problème, et prouver sa validité. Réponse □

Amélioration du débit

▷ **Question 13** Proposer un réseau pour la convolution non récursive où les (x_i) et les (y_i) circulent dans le même sens, et qui améliore le débit de la première version. Réponse □

III Réseau systolique pour le problème du chemin algébrique

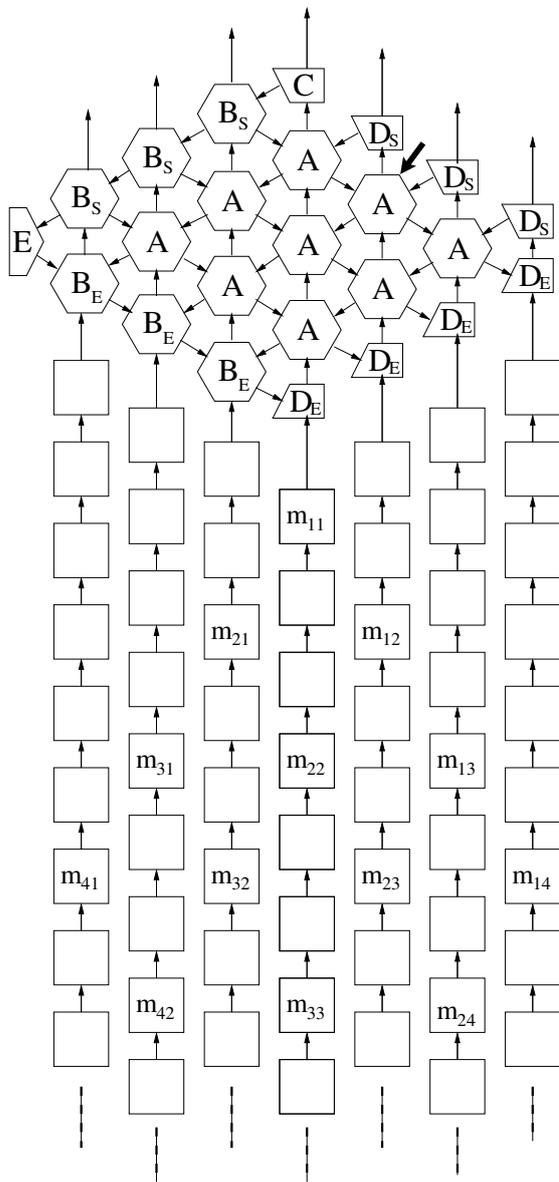
Notre but est d'implanter l'algorithme de la partie I à l'aide d'un réseau systolique. Nous conservons les définitions et conventions de la partie II pour tout ce qui concerne les caractéristiques générales d'un réseau systolique. On notera N le nombre de sommets du graphe.

Réseau proposé. Le réseau proposé est un réseau bidimensionnel de taille $(N+1) \times (N+1)$, composé de trois types de cellules. Les cellules sont disposées à l'intérieur d'un losange, et ont chacune au maximum six voisins. L'organisation complète du réseau est donnée en figure 2(a), pour la valeur exemple $N=4$. On pourra donc repérer une cellule du réseau par deux coordonnées : le numéro z de sa colonne ($-N \leq z \leq N-1$), et le numéro p de la cellule dans sa colonne, compté à partir du bas ($1 \leq p \leq N - |z| + 1$ pour une valeur de z donnée). Par exemple, la cellule signalée par une flèche sur la figure correspond aux coordonnées $(z=1, p=3)$.

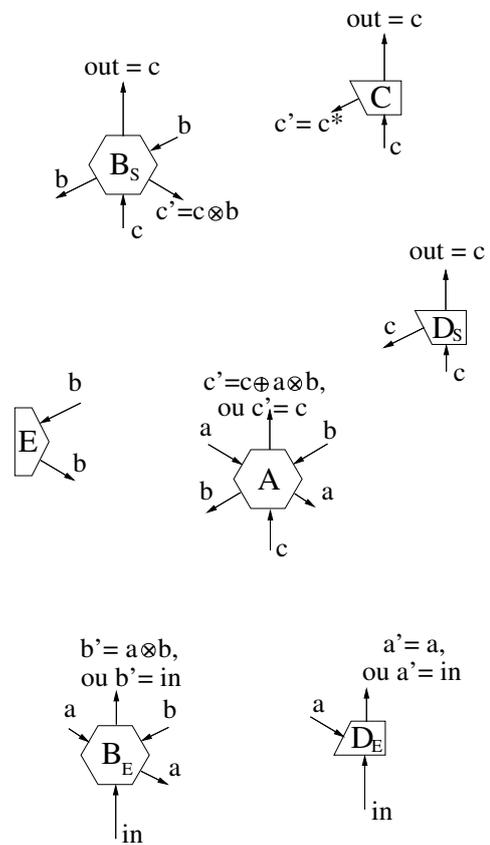
Types de cellules. Les trois types de cellules correspondent aux trois types de calculs suivants (cf. figure 2(b)) :

- type A : $c' := c \oplus (a \otimes b)$
- type B (avec les variantes B_E et B_S) : $b' := a \otimes b$ ou $c' := c \otimes b$
- type C : $c' = c^*$

À ceci s'ajoutent des cellules de type D_E, D_S et E , qui ne font que transmettre les données avec un délai d'un cycle d'horloge. Les entrées et sorties des cellules sont précisées figure 2(b). Chaque cellule dispose en outre d'un mécanisme de contrôle (non détaillé ici), qui permet de choisir entre calcul ou simple transmission des données (pour les cellules de type A et B_E), ou entre plusieurs directions possibles pour la transmission des données (pour les cellules de type B_E et D_E).



(a) Le réseau et les entrées-sorties, à $t = 0$. Les rectangles correspondent aux données en entrée. Les rectangles vides correspondent à des données non valides (donc à de simples délais).



(b) Les cellules. Chaque fil de sortie est muni d'un registre qui assure le décalage d'un cycle d'horloge entre entrées et sorties.

FIG. 2 – Le réseau et les différents types de cellules.

Initialisation, lecture et écriture des données. On suppose que tous les registres sont initialisés à la valeur zéro. Puis la matrice est introduite dans le réseau comme spécifié en figure 2(a).

Lorsque des cellules de type B_E ou D_E reçoivent des coefficients de la matrice, ceux-ci sont transmis vers le haut sans être modifiés. Ces coefficients sont ensuite acheminés et modifiés à travers le réseau. Lors du passage d'un coefficient dans une cellule, trois cas se présentent :

- la donnée est *passive* : le coefficient est simplement transmis d'une entrée vers une sortie ;
- la donnée est *écrite* : le coefficient est modifié par un calcul avant d'être transmis ;
- la donnée est *lue* : le coefficient n'est pas modifié avant transmission, mais sa valeur est utilisée pour modifier une autre donnée qui arrive au même instant dans la même cellule.

Les données peuvent donc être écrites dans deux types de situations : soit lors du passage par une cellule de type A , mais uniquement du bas vers le haut, soit lors du passage par une cellule de type B (accompagné par un changement de direction).

Circulation des données. La circulation d'une donnée dans le réseau peut être intuitivement expliquée de la façon suivante (cf. figure 3) :

- la donnée entre par un bord inférieur ; elle circule alors de bas en haut en restant dans la même colonne ; pendant ce trajet, elle est d'abord passive, puis écrite : ceci correspond à la première phase de l'algorithme ;
- arrivée au bord supérieur, elle est transmise par une cellule D sur une diagonale descendante jusqu'au bord inférieur opposé ; pendant cette phase elle est lue et sa valeur utilisée pour modifier d'autres coefficients ;
- puis intervient la phase 2 de l'algorithme, à nouveau du bord inférieur au bord supérieur sur une même colonne ;
- puis une nouvelle phase de transmission et lecture en diagonale ;
- et enfin une dernière phase quasi identique à la première, sur la même colonne que la colonne d'origine : lors de cette dernière phase, la donnée est passive dans les cellules où elle était écrite initialement, et réciproquement. Les écritures effectuées lors de cette dernière phase correspondent à la phase 3 de l'algorithme.

Les phases 1 et 3 de l'algorithme sont donc exécutées sur la même colonne, dans des ensembles de cellules complémentaires.

À la fin de cette dernière phase, les données sont sorties vers le haut du réseau en passant par les cellules B_S , D_S ou C .

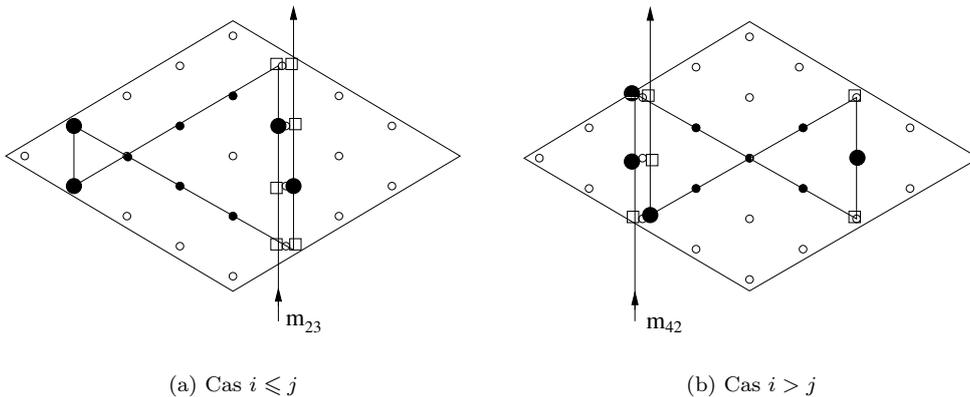


FIG. 3 – Chemin des données dans le réseau. Lors du passage dans une cellule, le coefficient $m_{i,j}$ peut être soit lu (symbole \circ), soit écrit (symbole \bullet), soit passif (symbole \square).

▷ **Question 14** On suppose que l'instant $t = 1$ correspond au cycle d'horloge où le coefficient $m_{1,1}$ entre dans la cellule ($z = 0, p = 1$). En distinguant les cas $i < j$ et $i \geq j$, déterminer pour chaque coefficient m_{ij} la date où il parvient

1. à l'entrée du réseau
2. en haut de sa première colonne (fin de la phase 1 de l'algorithme)
3. en bas de sa première diagonale
4. en haut de sa deuxième colonne (fin de la phase 2 de l'algorithme)
5. en bas de sa deuxième diagonale
6. à nouveau en haut de sa première colonne (un cycle avant sa sortie du réseau)

Réponse

▷ **Question 15** À quel instant le coefficient $m_{1,1}$ sort-il du réseau ? Même question pour le coefficient $m_{N,N}$. Dans quel ordre de ses coefficients et sous quelle forme la matrice résultat est-elle fournie en sortie ?

Réponse

▷ **Question 16** Montrer qu'il ne peut pas y avoir de conflit transmission/écriture dans les cellules B_E , c'est-à-dire que les deux actions suivantes

- une donnée arrive par le bas (en entrée) et est transmise vers le haut,
- une donnée arrive par la diagonale, est écrite et transmise vers le haut,

ne peuvent avoir lieu au même instant.

Réponse

▷ **Question 17** Même question pour les cellules de type D_E .

Réponse

▷ **Question 18** Pour i et j donnés, et pour k compris entre 1 et N , montrer qu'il existe un instant et une cellule (et préciser lesquels) où le coefficient m_{ij} rencontre les coefficients m_{ik} et m_{kj} . On distinguera suivant les valeurs respectives de i et j et les phases de l'algorithme. On suppose que le mécanisme de contrôle des cellules est spécifié de telle façon que la donnée m_{ij} est mise à jour (écrite) à cet instant.

Réponse

▷ **Question 19** En déduire — sous l'hypothèse que les mécanismes de contrôle sont corrects — que le réseau implante effectivement l'algorithme de la partie I.

Réponse

▷ **Question 20** Si l'on considère qu'une cellule n'est active que lorsqu'elle écrit (modifie) une donnée, donner pour chaque instant entre le moment où la première donnée entre dans le réseau et celui où la dernière en sort, le nombre de cellules actives.

Réponse

▷ **Question 21** Quel est le rendement (nombre de cellules actives / nombre total de cellules) moyen du réseau sur cet intervalle de temps ?

Réponse

▷ **Question 22** Proposer un ou plusieurs moyens d'améliorer ce rendement.

Réponse

IV Réponses aux exercices

Pour information, cet énoncé est probablement tiré d'un article de Günter Rote [1].

▷ Question 1, page 2

On choisit pour H l'ensemble des booléens $\{\text{VRAI}, \text{FAUX}\}$ muni des opérations **Ou** et **Et** habituelles (addition et multiplication booléennes).

Soit $G = (S, A, \omega)$ un graphe pour le quel on souhaite calculer le problème du chemin algébrique. Définissons $G' = (S, A')$ le graphe orienté par $(u, v) \in A'$ si et seulement si $\omega(u, v) = \text{VRAI}$. On a alors $d_{ij} = \text{VRAI}$ si et seulement si il existe un chemin de i à j dans le graphe G' .

▷ Question 2, page 2

Les poids sont choisis dans $H = \mathbb{R} \cup \{-\infty, +\infty\}$. \oplus est alors l'opération de minimum, le **0** est $+\infty$, \otimes est l'addition sur les réels étendue à H (avec en particulier $-\infty \oplus +\infty = +\infty$. On pourra remarquer que dans ces conditions on a $c^* = 0$ pour $c \geq 0$ et $c^* = -\infty$ sinon.

Soit $G = (S, A, \omega)$ un graphe pour le quel on souhaite calculer le problème du chemin algébrique. Définissons $G' = (S, A')$ le graphe orienté par $(u, v) \in A'$ si et seulement si $\omega(u, v) \in \mathbb{R}$. d_{ij} est alors égal au poids du plus court chemin de i à j dans le graphe G' .

▷ Question 3, page 2

La démonstration se fait par récurrence sur p .

$p = 0$ Par convention, le poids du chemin vide qui commence et finit au nœud i est **1**. Il n'y a pas de chemin contenant à 0 arcs entre i et j si $i \neq j$. On a donc bien $(m_{ij}^{(0)}) = I = M^0$.

$p = 1$ Par définition de la matrice M , on a immédiatement $M = (m_{ij}^{(1)})$.

$p \rightsquigarrow p + 1$ Supposons l'égalité vraie pour un certain $p \in \mathbb{N}$. On a $M^{p+1} = M \otimes M^p$ et on sait par hypothèse de récurrence que

$$m_{kj}^{(p)} = \bigoplus_{\substack{c \text{ chemin de } k \text{ à } j \\ \text{contenant exactement } p \text{ arcs}}} \omega(c).$$

Tout chemin de i à j contenant exactement $p + 1$ arc se décomposant de façon unique en un arc de i à un certain k et un chemin à p arcs de k à j , on a

$$m_{ik} \otimes m_{kj}^{(p)} = \bigoplus_{\substack{c \text{ chemin de } i \text{ à } j \\ \text{de deuxième sommet } k \\ \text{contenant exactement } p + 1 \text{ arcs}}} \omega(c).$$

* On notera que le cas où il n'y a pas d'arc de k à j ne pose pas de problème puisque la somme sur un ensemble vide est égale par convention à **0** et que m_{kj} vaut également **0** dans un tel cas.

On a donc

$$(M \otimes M^p)_{ij} = \bigoplus_{k=1}^n m_{ik} \otimes m_{kj}^{(p)} = \bigoplus_{\substack{c \text{ chemin quelconque de } i \text{ à } j \\ \text{contenant exactement } p + 1 \text{ arcs}}} \omega(c) = m_{ij}^{(p+1)},$$

ce qui établit le résultat.

▷ **Question 4, page 2**

Remarque préliminaire : La multiplication de matrices peut se ramener au problème du chemin algébrique. Soient A, B deux matrices compatibles. Alors en utilisant la matrice

$$M = \begin{pmatrix} 0 & A & 0 \\ 0 & 0 & B \\ 0 & 0 & 0 \end{pmatrix}$$

on a

$$D = \sum_{i \geq 0} M^i = \begin{pmatrix} I & A & AB \\ 0 & I & B \\ 0 & 0 & I \end{pmatrix}$$

Soit A une matrice (inversible) que l'on souhaite inverser. Posons $A' = \frac{1}{n(\|A\|_\infty + 1)}A$, où $\|A\|_\infty = \max_{i,j} |a_{ij}|$.

La matrice $N = {}^t A' A'$ est alors symétrique définie positive et les valeurs propres N sont dans $]0, 1[$. En effet, soit λ une valeur propre de N . On a

$$\lambda \leq n\|N\|_\infty \leq n^2\|A'\|_\infty^2 \leq \frac{n^2\|A\|_\infty^2}{n^2(\|A\|_\infty + 1)^2} < 1$$

Les valeurs propres de $I - N$ sont donc également incluses dans $]0, 1[$ et la somme

$$\sum_{k=0}^{\infty} (I - N)^k$$

converge donc. Quand cette somme converge, il est aisé de vérifier que $\sum_{k=0}^{\infty} (I - N)^k = N^{-1}$.

Pour obtenir A^{-1} , il suffit de remarquer que $A'^{-1} = N^{-1} \cdot {}^t A'$ et donc que

$$A^{-1} = \frac{1}{n(\|A\|_\infty + 1)} \cdot A'^{-1} = \frac{1}{n^2(\|A\|_\infty + 1)^2} N^{-1} \cdot {}^t A$$

Pour calculer l'inverse d'une matrice A , il est donc possible de n'utiliser qu'un calcul de problème du chemin algébrique et des multiplications de matrices. Le calcul de l'inverse d'une matrice se ramène donc bien à un calcul de problème du chemin algébrique. Dans le cas où la matrice n'est pas inversible, le calcul du problème du chemin algébrique divergera puisque $I - N$ aura une valeur propre égale à 1.

▷ **Question 5, page 2**

Cas $i \neq k$ et $j \neq k$: soit c un chemin dans $\mathcal{C}_{ij}^{(k)}$. Soit c ne passe pas par k auquel cas il est dans $\mathcal{C}_{ij}^{(k-1)}$, soit il est décomposable en un segment initial arrivant à la dernière occurrence de k (et donc appartenant à $\mathcal{C}_{ik}^{(k)}$), et un segment final qui ne passe pas par k (et qui appartient donc à $\mathcal{C}_{kj}^{(k-1)}$). On a donc

$$p_{ij}^{(k)} = p_{ij}^{(k-1)} \oplus p_{ik}^{(k)} \otimes p_{kj}^{(k-1)}$$

Cas $i \neq j$, $i = k$ ou $j = k$: $\mathcal{C}_{ii}^{(i)}$ contenant le chemin vide allant de i à i , on a :

$$p_{ij}^{(i)} = p_{ii}^{(i)} \otimes p_{ij}^{(i-1)}.$$

De même, on a

$$p_{ij}^{(j)} = p_{ij}^{(j-1)} \otimes p_{jj}^{(j)}.$$

Cas $i = j = k$: chaque chemin de $\mathcal{C}_{ii}^{(i)}$ est décomposable de façon unique en une suite de chemins partiels de $\mathcal{C}_{ii}^{(i-1)}$. On a donc

$$p_{ii}^{(i)} = \left(p_{ii}^{(i-1)} \right)^*.$$

▷ **Question 6, page 3**

En utilisant la question précédente, on en déduit directement l'algorithme suivant :

```

PHASE 1 :
1   pour i de 1 a N
2   pour j de 1 a N
3   faire
4       pour k de 1 a min(i, j) - 1
5            $p_{ij}^{(k)} := p_{ij}^{(k-1)} \oplus p_{ik}^{(k)} \otimes p_{kj}^{(k-1)}$  ;
6       si i = j alors  $p_{ii}^{(i)} := (p_{ii}^{(i-1)})^*$  ;
7       si i > j alors  $p_{ij}^{(j)} := p_{ij}^{(j-1)} \otimes p_{jj}^{(j)}$  ;
8   fait ;
PHASE 2 :
9   pour i de 1 a N
10  pour j de 1 a N
11  faire
12      si i < j alors  $p_{ij}^{(i)} := p_{ii}^{(i)} \otimes p_{ij}^{(i-1)}$  ;
13      pour k de min(i, j) + 1 a max(i, j) - 1
14           $p_{ij}^{(k)} := p_{ij}^{(k-1)} \oplus p_{ik}^{(k)} \otimes p_{kj}^{(k-1)}$  ;
15      si i < j alors  $p_{ij}^{(j)} := p_{ij}^{(j-1)} \otimes p_{jj}^{(j)}$  ;
16  fait ;
PHASE 3 :
17  pour i de 1 a N
18  pour j de 1 a N
19  faire
20      si i > j alors  $p_{ij}^{(i)} := p_{ii}^{(i)} \otimes p_{ij}^{(i-1)}$  ;
21      pour k de max(i, j) + 1 a N
22           $p_{ij}^{(k)} := p_{ij}^{(k-1)} \oplus p_{ik}^{(k)} \otimes p_{kj}^{(k-1)}$  ;
23  fait ;

```

L'ordre dans lequel les boucles sur i et j sont exécutées est tel que pour toute paire (i', j') vérifiant $i' \leq i$ et $j' \leq j$, la valeur en (i', j') est calculée avant celle en (i, j) .

- La phase 1 part de $p_{ij}^{(0)}$ et calcule $p_{ij}^{(j)}$ pour $i \geq j$ et $p_{ij}^{(i-1)}$ pour $i < j$.
- La phase 2 part de ces valeurs et calcule $p_{ij}^{(j)}$ si $i \leq j$ et $p_{ij}^{(i-1)}$ si $i > j$.
- La phase 3 calcule enfin les $p_{ij}^{(n)}$

Il nous faut donc montrer que l'on peut se passer des indices supérieurs. Allons-y gaiement.

Pour la phase 1 :

- Ligne 5 : comme $k < i$, on a $p_{ik} = p_{ik}^{(k)}$ (il a été calculé en ligne 7, à l'itération précédente). $p_{kj}^{(k)}$ n'a pas encore été calculé et on a donc $p_{kj} = p_{kj}^{(k-1)}$. En calculant $p_{ij} := p_{ij} \oplus p_{ik} \otimes p_{kj}$; on a donc bien au final $p_{ij} = p_{ij}^{(\min(i,j)-1)} = p_{ij}^{(i-1)}$.
- Ligne 6 : on utilise la valeur calculée à l'itération précédente donc tout va bien et $p_{ii}^{(i)} = p_{ii}$.
- Ligne 7 : comme $i > j$, $p_{jj}^{(j)}$ a bien été calculé (en ligne 6 à une itération précédente) et correspond à la valeur courante de p_{jj} . On a donc bien $p_{ij} = p_{ij}^{(j)}$.

Pour la phase 2 :

- Ligne 12 : comme $i < j$, on a $p_{ii} = p_{ii}^{(i)}$ (valeurs calculées en ligne 6). $p_{ij}^{(v)}$ aut donc initialement $p_{ij}^{(\min(i,j)-1)}$ et, après cette affectation, on a donc $p_{ij} = p_{ij}^{(\min(i,j))}$.
- Ligne 14 : si $i > j$, $p_{ik}^{(k)}$ a été calculé en ligne 7 et vaut p_{ik} . Si $i < j$, alors $k < j$ et $p_{ik}^{(k)}$ a été calculé en ligne 15 à une itération précédente et vaut p_{ik} . p_{kj} vaut toujours $p_{kj}^{(k-1)}$ car (k, j)

n'a pas encore été traité pour la ligne 12. On a donc $p_{ij} = p_{ij}^{(\max i, j-1)}$.

– Ligne 15 : si $i < j$ alors $p_{jj}^{(j)}$ a été calculé en ligne 6 et on a donc au final $p_{ij} = p_{ij}^{(j)}$.

Pour la phase 3 :

– Ligne 20 : $p_{ii}^{(i)}$ a été calculé en ligne 6 et n'a pas été modifié depuis et donc au final $p_{ij} = p_{ij}^{(i)}$.

– Ligne 22 : $p_{ik}^{(k)}$ a été calculé en ligne 7 ou en ligne 15. $p_{kj}^{(k-1)}$ a été calculé en ligne 7 ou en ligne 14.

On a donc $p_{ij} = p_{ij}^{(N)}$ pour tout i, j .

Pour résumer, voici les différentes lignes où sont calculées les $p_{ij}^{(k)}$:

$i < j$					$i = j$			$i > j$				
$k < i$	$k = i$	$i < k < j$	$k = j$	$k > j$	$k < i$	$k = i$	$k > i$	$k < j$	$k = j$	$j < k < i$	$k = i$	$k > i$
5	12	14	15	22	5	6	22	5	7	14	20	22

▷ Question 7, page 4

Le terme $a_1 x_i$ n'apparaissant que dans le calcul de y_i , x_i et y_i doivent entrer en même temps dans la première cellule. À l'instant $t = 1$, c'est donc la valeur y_1 qui doit se trouver à l'entrée de la cellule 1.

▷ Question 8, page 4

Si au temps t , la donnée x_i arrive dans la cellule 1, cela signifie qu'au temps $t + 1$, on aura ajouté $a_1 \times x_i$ à ce qui était entré en y dans la cellule 1 au temps t . Ce terme n'apparaissant que dans y_i , cela signifie qu'au temps t , y_i est à l'entrée de la cellule 1 et que l'on vient de rajouter (dans la seconde cellule) à y_i la valeur $a_2 \times x_{i-1}$. x_{i-1} était donc à l'entrée de la cellule 2 au temps $t - 1$ et est donc à l'entrée de la cellule 3 au temps t . De même, y_{i+1} est à l'entrée de la cellule 3 au temps t .

▷ Question 9, page 4

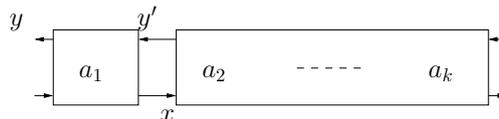
Les données sont entrées dans le réseau un cycle sur 2. x_1 étant dans la cellule p au temps $t = p$, x_i est donc dans la cellule p au temps $t = 2i + p - 2$. Au même moment, dans cette cellule, on calcule y_{i+p-1} . y_i est donc dans la cellule p au temps $t = 2(i - p + 1) + p - 2 = 2i - p$.

▷ Question 10, page 4

Les données entrent et sortent à raison d'un cycle sur deux. Le débit en régime permanent est donc de $1/2$.

▷ Question 11, page 5

On montre le résultat par récurrence sur la taille du réseau.

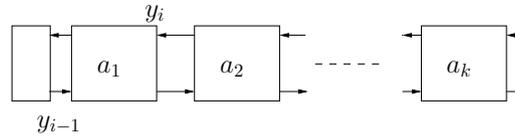


Au temps $t = 1$, x_1 rentre en a_1 par la gauche et que y'_1 rentre en a_1 par la droite. On suppose par hypothèse de récurrence que $y'_i = \sum_{j=2}^k a_j x_{i-j+1}$ et on va montrer que $y_i = \sum_{j=1}^k a_j x_{i-j+1}$. Au temps $t = 2i$, x_i rentre donc en a_1 par la gauche et y'_i rentre par la droite. Au temps $t = 2i + 1$, $y_i = a_1 x_i + y'_i = \sum_{j=1}^k a_j x_{i-j+1}$ sort donc par la gauche.

On peut également montrer que les synchro

▷ **Question 12, page 5**

Il suffit d'ajouter une cellule de retard à gauche du réseau. Quand un variable y_i circule de droite à gauche, elle est en train d'accumuler sa valeur définitive, puis elle «rebondit» dans la cellule de retard pour circuler inchangée en sens inverse, jouant le rôle des x_i dans le réseau précédent.



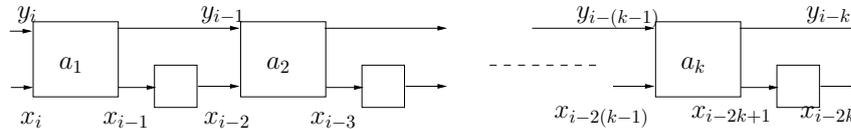
Pour $i \leq k$, les y_i circulent de droite à gauche : ce sont les coefficients d'initialisation.

La validité du réseau découle directement de celle du réseau précédent : lorsque y_i est calculé en a_1 , y_{i-1} est utilisé pour ce calcul (comme x_i précédemment). Il suffit donc d'effectuer une translation sur les indices : x_l devient y_{l-1} et on obtient

$$y_i = \sum_{j=1}^k a_j x_{i-j+1} = \sum_{j=1}^k a_j y_{i-j}$$

▷ **Question 13, page 5**

Si l'on souhaite améliorer notablement le débit, il va falloir qu'un nouvel y_i sorte à chaque top du réseau. Il va donc falloir que les y_i se déplacent deux fois plus vite que les x_i , ce qui peut s'obtenir en insérant des délais supplémentaires pour les x_i .



▷ **Question 14, page 8**

Au temps $t = 1$, m_{11} entre dans la cellule $(z, p) = (0, 1)$. La colonne centrale contient $n + 1$ éléments et pour les autres colonnes ($z \in [-n, n - 1]$), la colonne z contient $n - |z| + 1$ éléments. L'élément m_{ij} se trouve donc initialement dans la colonne $z = j - i$.

On commence par décrire le mouvement de m_{ij} pour $i \leq j$:

- **Entrée** : l'entrée se fait en $(j - i, 1)$ au temps $t_0 = 2j + i - 2$.
- **Montée** : au temps $2j + i - 2 + l$ (pour $l \in [0, N - (j - i)]$) m_{ij} se trouve en position $(j - i, l + 1)$.
- **Changement de direction** : la descente le long de la première diagonale commence en $(j - i, N + 1 - j + i)$ au temps $t_1 = N + j + 2i - 2$.
- **Descente** : au temps $N + j + 2i - 2 + l$ (pour $l \in [0, j - i]$) m_{ij} se trouve en $(j - i - l, N + 1 - (j - i))$. Au temps $N + 2j + i - 2 + l$ (pour $l \in [0, N - (j - i)]$), m_{ij} se trouve en $(-l, N + 1 - (j - i) - l)$.
- **Changement de direction** : la remontée commence en $(j - i - N, 1)$ au temps $t_2 = 2N + 2i + j - 2$.
- **Montée** : au temps $2N + 2i + j - 2 + l$ (avec $l \in [0, j - i]$), m_{ij} se trouve en $((j - i) - N, l + 1)$.
- **Changement de direction** : on s'oriente sur la seconde diagonale en $((j - i) - N, j - i + 1)$ au temps $t_3 = 2N + 2j + i - 2$.
- **Descente** : au temps $2N + 2j + i - 2 + l$ (pour $l \in [0, N - (j - i)]$) m_{ij} est en position $((j - i) - N + l, j - i + 1)$. Au temps $3N + 2i + j - 2 + l$ (pour $l \in [0, j - i]$), m_{ij} est en position $(l, j - i - l + 1)$.

- **Changement de direction** : on s'oriente sur la colonne initiale en $(j - i, 1)$ au temps $t_4 = 3N + 2j + i - 2$.
- **Montée** : au temps $3N + 2j + i - 2 + l + 1$ (avec $l \in [0, N - (j - i)]$), m_{ij} se trouve en $(j - i, l + 1)$.
- **Sortie** : m_{ij} sort donc du réseau en $(j - i, N + 1 - (j - i))$ au temps $4N + 2i + j - 2$.

Pour $i > j$, le mouvement de m_{ij} est le suivant :

- **Entrée** : l'entrée se fait en $(j - i, 1)$ au temps $t_0 = 2i + j - 2$.
- **Montée** : au temps $2i + j - 2 + l$ (pour $l \in [0, N - (i - j)]$) m_{ij} se trouve en position $(j - i, l + 1)$.
- **Changement de direction** : la descente le long de la première diagonale commence en $(j - i, N + 1 - (i - j))$ au temps $t_1 = N + 2j + i - 2$.
- **Descente** : au temps $N + 2j + i - 2 + l$ (pour $l \in [0, i - j]$) m_{ij} se trouve en $(j - i + l, N + 1 - (i - j))$. Au temps $N + j + 2i - 2 + l$ (pour $l \in [0, N - (i - j)]$), m_{ij} se trouve en $(l, N + 1 - (i - j) - l)$.
- **Changement de direction** : la remontée commence en $(N - (i - j), 1)$ au temps $t_2 = 2N + i + 2j - 2$.
- **Montée** : au temps $2N + i + 2j - 2 + l$ (avec $l \in [0, (i - j)]$), m_{ij} se trouve en $(N - (i - j), l + 1)$.
- **Changement de direction** : on s'oriente sur la seconde diagonale en $(N - (i - j), i - j + 1)$ au temps $t_3 = 2N + 2i + j - 2$.
- **Descente** : au temps $2N + 2i + j - 2 + l$ (pour $l \in [0, N - (i - j)]$) m_{ij} est en position $(N - (i - j) - l, i - j + 1)$. Au temps $3N + 2j + i - 2 + l$ (pour $l \in [0, i - j]$), m_{ij} est en position $(-l, i - j - l + 1)$.
- **Changement de direction** : on s'oriente sur la colonne initiale en $(j - i, 1)$ au temps $t_4 = 3N + 2i + j - 2$.
- **Montée** : au temps $3N + 2i + j - 2 + l + 1$ (avec $l \in [0, N - (i - j)]$), m_{ij} se trouve en $(j - i, l + 1)$.
- **Sortie** : m_{ij} sort donc du réseau en $(j - i, N + 1 - (i - j))$ au temps $4N + i + 2j - 2$.

▷ Question 15, page 8

Le coefficient m_{11} sort du réseau au temps $t_5 + 1 = 4N + 2$ et le coefficient m_{NN} sort du réseau au temps $t_5 + 1 = 7N - 1$. La matrice sors du réseau sous la même forme que celle sous laquelle elle est rentrée.

▷ Question 16, page 8

Considérons une cellule B_E . Une donnée qui arrive par le bas correspond à un m_{ij} avec $i > j$, initialement dans la colonne $j - i$. Elle arrive dans la cellule B_E au temps $t_0 = 2i + j - 2$.

Les données qui arrivent par la diagonale sont initialement sur la colonne $j - i + N$. Elles correspondent à des coefficient $m_{i'j'}$ avec $j' - i' = j - i + n$. Ces données entrent dans la cellule B_E au temps $t'_2 = j' + 2i' + 2N - 2$.

On a alors $t'_2 - t_0 = 2(i' - i) + (j' - j + 2N) = 3(i' - i) + 3N > 0$.

▷ Question 17, page 8

On raisonne de la même façon avec $j \geq i$. Le coefficient m_{ij} arrive dans la cellule D_E au temps $t_0 = i + 2j - 2$. Les données qui arrivent par la diagonale sont initialement sur la colonne $j - i - N$. Ce sont des $m_{i'j'}$ avec $j' - i' = j - i - N$. Ces données sont dans la cellule D_E au temps $t'_2 = 2j' + i' + 2N - 2$.

On a alors $t'_2 - t_0 = 2(j' - j) + (i' - i) + 2N = 3(j' - j) + 3N > 0$.

▷ Question 18, page 8

La solution la plus simple consiste à dessiner les trajectoires des éléments impliqués comme dans la figure 4.

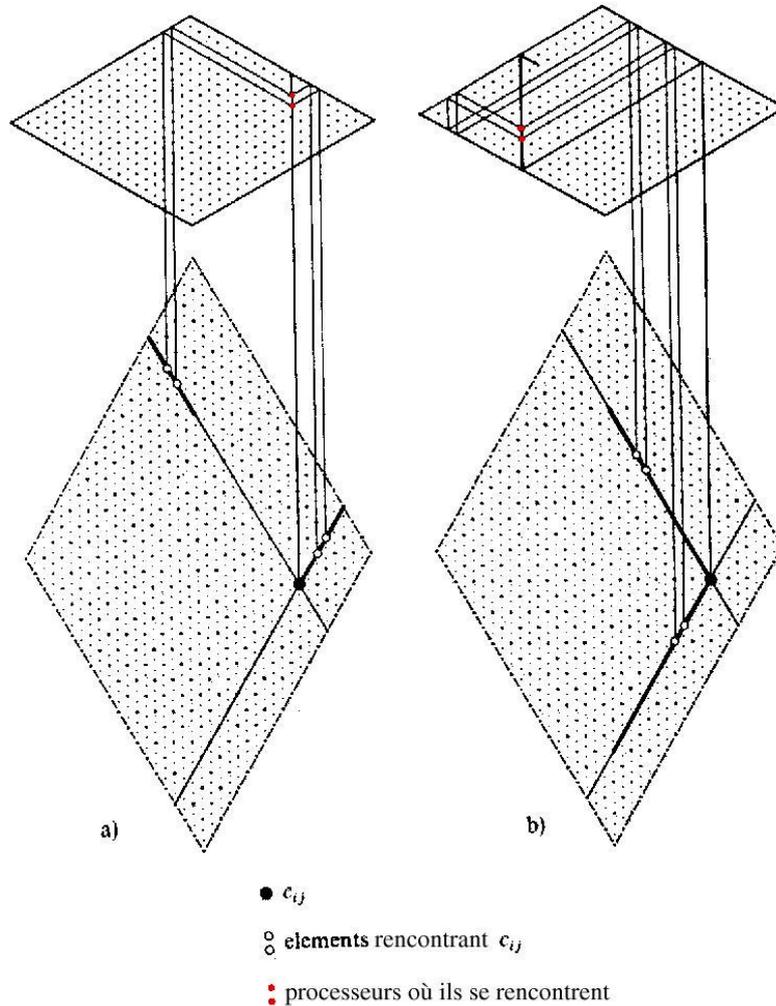


FIG. 4 – Trajectoire et collisions

Le dessin de gauche représente la première phase puisque m_{ij} rencontre les m_{ik} et m_{kj} pour k compris entre 1 et $\min(i, j)$. Le dessin de droite représente la seconde phase puisque m_{ij} rencontre les m_{ik} et m_{kj} pour k compris entre $\min(i, j) + 1$ et $\max(i, j) - 1$.

On notera qu'il est possible de montrer tout cela formellement en vérifiant les différents cas mis en valeur lors de la réponse à la question 6 et en utilisant les trajectoires calculées pour la questions 14. À titre d'exemple, je traite quelques cas mais c'est quand même bien besogneux...

$i < j$ et $k < i$ Montrons que m_{ik} et m_{kj} croisent m_{ij} alors que m_{ij} est en train d'effectuer sa première montée le long de la colonne $j - i$, que m_{ik} est en train d'effectuer sa descente le long de sa première diagonale et que m_{kj} est en train d'effectuer sa descente le long de sa première diagonale.

D'après les calculs effectués pour la question 14, m_{ik} se trouve en $(l, N + 1 - (i - k) - l)$ au temps $(N + k + 2i - 2 + l)$. Il croise donc la colonne $j - i$ au temps $N + k + i + j - 2$ et en position $(j - i, N + 1 + k - j)$.

D'après les calculs effectués pour la question 14, m_{ij} se trouve en $(j - i, l' + 1)$ au temps $2j + i - 2 + l'$. Il est donc également en position $(j - i, N + 1 + k - j)$ au temps $N + k + i + j - 2$. D'après les calculs effectués pour la question 14, m_{kj} se trouve en $(j - k - l'', N + 1 + k - j)$ au temps $N + j + 2k - 2 + l''$. Il est donc également en position $(j - i, N + 1 + k - j)$ au temps $N + k + i + j - 2$. Cela correspond bien à la première phase de l'algorithme.

En regardant le reste des déplacements, on peut vérifier qu'à aucun autre instant m_{ij} ne croise ce m_{ik} ou ce m_{kj} .

$i < j$ et $k = i$ Montrons que m_{ii} croise m_{ij} alors que m_{ij} est en train de commencer sa seconde montée (le long de la colonne $(j - i) - N$) et que m_{ii} est en train d'effectuer sa descente le long de sa seconde diagonale.

D'après les calculs effectués pour la question 14, m_{ii} se trouve en $(l - N, 1)$ au temps $2N + i - 2 + l$. Il croise donc la colonne $j - i - N$ au temps $2N + j - 2$ et en position $(j - i, 1)$.

D'après les calculs effectués pour la question 14, m_{ij} se trouve en position $(j - i, 1)$ au temps $2N + j - 2$.

Cette cellule étant de type B_E , on a bien $m_{ij} \leftarrow m_{ii} \otimes m_{ij}$, ce qui correspond à la ligne 12.

▷ Question 19, page 8

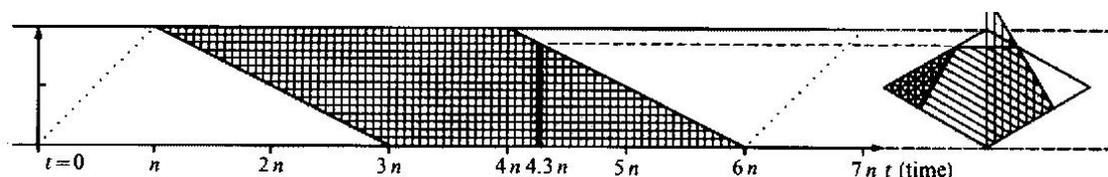
On déduit, on déduit...

▷ Question 20, page 8

Les schémas de la figure 5 résument le mouvement des données dans le réseau. Le petit losange représente le réseau et les lignes parallèles internes à la matrice représente la direction courante des éléments.

Au temps $t = -0.1n$, la matrice est située à l'extérieur du réseau. Elle se déplace alors vers le haut ($t = 0.5n$) et au temps $t = n$, les premiers éléments commencent à rebondir ($t = 1.3n$). Pour plus de clarté, on ne représente qu'une moitié de la matrice dans les schémas suivants, le schéma complet étant obtenu en superposant le schéma dont on a pris le symétrique par rapport à l'axe horizontale passant par le centre du réseau. Après le temps $2n$ (au temps $t = 2.3n$ par exemple), la matrice est repliée trois fois et entre les temps $3n$ et $4n$, elle est repliée quatre fois et est complètement comprise dans le réseau. Au temps $t = 4n$, la matrice commence à sortir du réseau.

Si on regarde quels sont les processeurs actifs, on peut remarquer que les processeurs d'un même niveau horizontal sont soit tous actifs, soit tous passifs. On en déduit donc le diagramme d'utilisation des différents niveaux suivants :



À la louche, on obtient donc les activités suivantes

- pour $t \in [0, n]$, aucun processeur n'est actif;
- pour $t \in [n, 2n]$, $\frac{(t-n)}{2} \times (t - n)$ processeurs sont actifs;
- pour $t \in [2n, 3n]$, $n^2/2 + \frac{(t-2n)}{2} \times (4n - t)$ processeurs sont actifs;
- pour $t \in [3n, 4n]$, n^2 processeurs sont actifs;
- pour $t \in [4n, 5n]$, $n^2/2 + \frac{(5n-t)}{2} \times (t - 3n)$ processeurs sont actifs;
- pour $t \in [5n, 6n]$, $\frac{(6n-t)}{2} \times (t - 4n)$ processeurs sont actifs;
- pour $t \in [0, 7n]$, aucun processeur n'est actif;

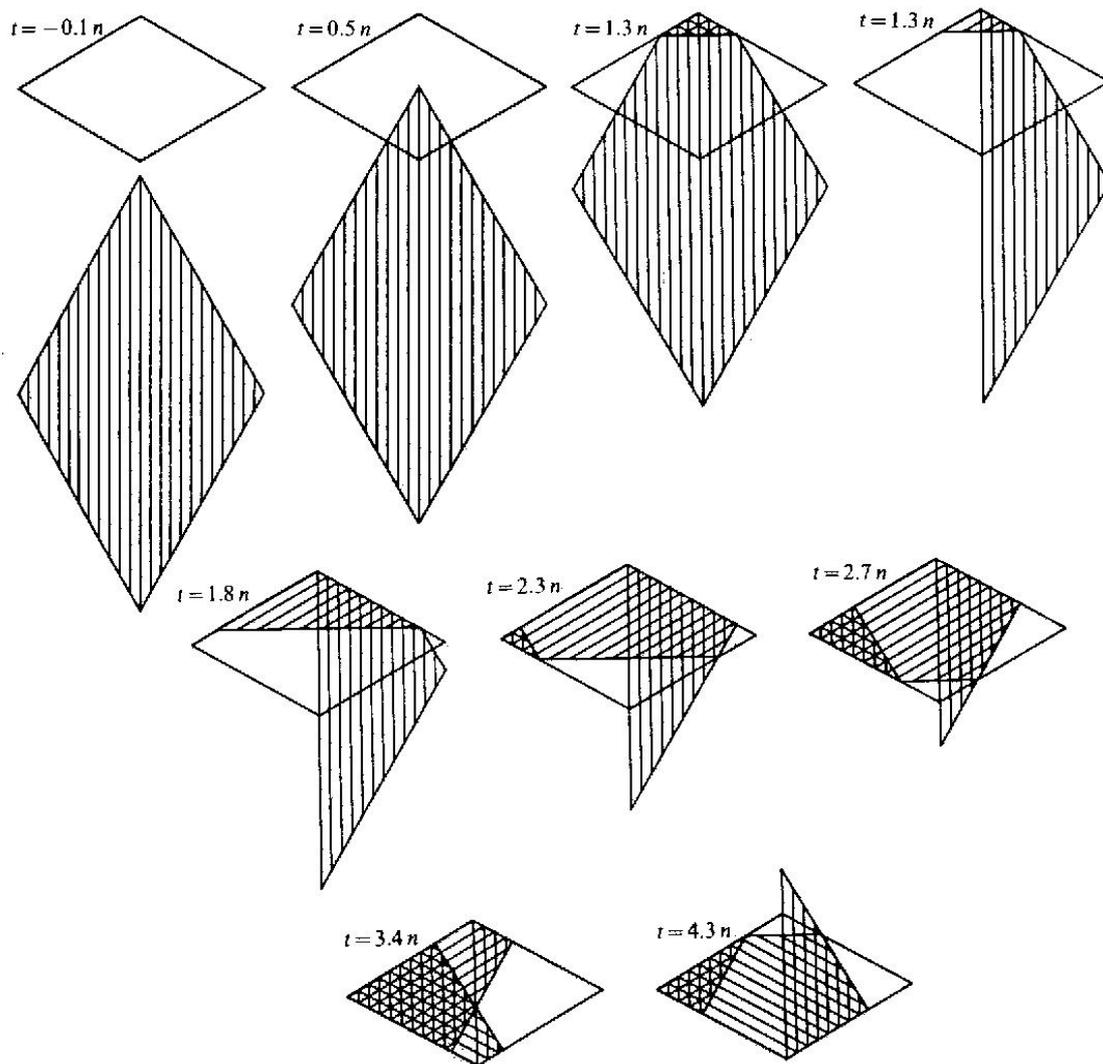


FIG. 5 – Mouvement pendant les différentes phases de calcul

▷ Question 21, page 8

Il y a deux façons de répondre à cette question. Soit on utilise la question d'avant, soit on calcule la complexité de l'algorithme.

La première phase a une complexité de $\sum_i \sum_j (\min(i, j) - 1) + n + n(n - 1)/2 = \frac{n^3 + 2}{3}$. En effectuant un calcul similaire, on trouve que la seconde phase a une complexité de $\frac{n^3 + 4n^2 - 5n + 5}{3}$. La troisième et dernière phase a une complexité de $\frac{n^3 + 3n^2 - 1}{3}$. Il y a donc $\frac{3n^3 + 7n^2 - 14n + 4}{3}$ cellules actives et ce pendant une durée de $7n - 1$ temps de cycle. Le débit du réseau est donc de $\frac{1}{7}$ et son efficacité est de $\frac{3}{7}$.

Mais cette valeur n'est pas très pertinente puisqu'on peut commencer à faire rentrer une seconde matrice avant que la première ne soit complètement sortie. En effet la dernière donnée m_{nn} amorce sa remontée au temps $t = 6n$ et il est possible de faire rentrer la matrice suivante juste après. Le véritable débit de ce réseau est donc de $\frac{1}{6}$ et son efficacité de $\frac{3}{6} = \frac{1}{2}$.

▷ Question 22, page 8

La forme du diagramme d'utilisation des différents niveaux suggère la possibilité qu'une seconde matrice puisse être entrée $3n$ étapes (et non $6n$) après l'entrée de la première. Les éléments de la seconde matrice utilisent alors les processeurs inutilisés (pour un calcul) par la première. Cependant, pour $t \in [3n, 4n]$ l'intégralité de la matrice est dans le réseau et il n'est donc pas possible de stocker ni de faire circuler une seconde matrice... à moins d'augmenter la mémoire des cellules et de doubler les liens. Le mécanisme de contrôle est alors un peu plus délicat mais toutes les cellules sont effectivement actives et les deux calculs s'entrelacent effectivement pour atteindre un débit de $\frac{1}{3}$ et une efficacité de 1.

Références

- [1] Günter Rote. A systolic array algorithm for the algebraic path problem (shortest paths; matrix inversion). *Computing*, 34 :191–219, 1985. Available at [http://page.mi.fu-berlin.de/~rote/Papers/pdf/A+systolic+array+algorithm+for+the+algebraic+path+problem+\(shortest+paths;+matrix+inversion\).pdf](http://page.mi.fu-berlin.de/~rote/Papers/pdf/A+systolic+array+algorithm+for+the+algebraic+path+problem+(shortest+paths;+matrix+inversion).pdf).