

## Nids de boucles - suite

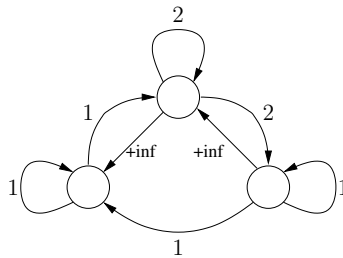
### 1 Algorithme d'Allen et Kennedy

▷ **Question 1** On considère le code suivant :

**Pour**  $i = 1$  à  $N$  **Pour**  $j = 1$  à  $N$   
 $S_1 : a(i + 1, j + 1) \leftarrow a(i + 1, j) + b(i, j + 2)$   
 $S_2 : b(i + 1, j) \leftarrow a(i + 1, j - 1) + b(i, j - 1)$   
 $S_3 : a(i, j + 2) \leftarrow b(i + 1, j + 1) - 1$

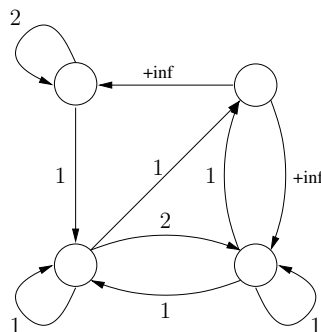
Donner le graphe de dépendances réduit, avec pour chaque dépendance son type (flot, anti, sortie) et son niveau. Appliquer l'algorithme d'Allen et Kennedy pour restructurer le nid de boucles et vérifier la nature des boucles obtenues.

▷ **Question 2** Donner un nid de boucles parfait où toutes les dépendances sont uniformes, et dont le GDRN est exactement le graphe suivant :



Exécuter l'algorithme d'Allen et Kennedy sur ce nid de boucles.

▷ **Question 3** Donner un nid de boucles parfait où toutes les dépendances sont uniformes, et dont le GDRN est exactement le graphe suivant :



Exécuter l'algorithme d'Allen et Kennedy sur ce nid de boucles.

### 2 Permutation de boucles

On considère un nid de boucle parfait de profondeur  $n$ , qui comprend  $m$  dépendances uniformes. Soit  $D$  la matrice de taille  $n \times m$ , obtenue à partir du signe des composantes des vecteurs de dépendance. Si  $d_i$  est le  $i$ -ème vecteur de dépendance, on stocke le signe  $+$ ,  $0$  ou  $-$  de ses composantes dans la  $i$ -ème colonne de  $D$ . Par exemple si  $n = 3$  et  $d_1 = (2, -2, 0)^t$ , on stockera  $(+, -, 0)^t$  dans la première colonne de  $D$ .

▷ **Question 4** Donner la matrice  $D_1$  pour le nid suivant :

```

for i = 1 to n do
  for j = 1 to n do
    for k = 1 to n do
      S1 : a(i + 1, j, k) = a(i, j, k) + 2
      S2 : b(i, j, k + 1) = b(i, j, k) - 1
      S3 : c(i + 1, j + 1, k + 1) = c(i, j, k) + 1

```

▷ **Question 5** Donner la matrice  $D_2$  pour le nid suivant :

```

for i = 1 to n do
  for j = 1 to n do
    for k = 1 to n do
      S1 : a(i, j, k + 1) = a(i, j - 1, k) + a(i - 1, j, k + 2) + 2

```

▷ **Question 6** Donner un nid dont la matrice  $D_3$  soit

$$D_3 = \begin{pmatrix} + & + & + & 0 & 0 & 0 \\ + & 0 & 0 & + & 0 & 0 \\ 0 & + & 0 & 0 & + & 0 \\ 0 & 0 & + & 0 & 0 & + \end{pmatrix}$$

L'algorithme de parallélisation fonctionne ainsi :

- Si une ligne de la matrice  $D$  (disons la  $i$ -ème) ne contient que des 0, alors on place cette boucle à l'extérieur du nid et on l'exécute en parallèle : on permute les boucles, la  $i$ -ème boucle devient la première boucle, qui est exécutée en parallèle.
- Si aucune ligne de la matrice  $D$  ne contient que des 0, on sélectionne la ligne qui contient le plus de +, on place la boucle correspondante à l'extérieur et on l'exécute en séquentiel.

On exécute alors récursivement l'algorithme sur les boucles et les dépendances restantes, i.e. qui n'ont pas été satisfaites par la séquentialisation de la boucle.

Par exemple pour  $D_1$ , on choisit de séquentialiser soit la première soit la troisième boucle. Si on choisit la première, pas besoin de permuter, les dépendances 1 et 3 sont satisfaites. Il reste deux boucles à l'intérieur, avec la matrice réduite  $\overline{D}_1 = (0, +)^t$ . On en déduit le code parallèle, où `forseq` représente une boucle exécutée en séquentielle, et `forpar` représente une boucle exécutée en parallèle :

```

forseq i
  forpar j
    forseq k
    ...

```

Bien sûr, on aurait pu choisir d'abord la boucle sur  $k$  et obtenir :

```

forseq k
  forpar j
    forseq i
    ...

```

▷ **Question 7** Faire tourner l'algorithme sur l'exemple avec  $D_2$ .

▷ **Question 8** Faire tourner l'algorithme sur l'exemple avec  $D_3$ .

▷ **Question 9** Prouver la correction de l'algorithme. Pour cela, prouver qu'on a le droit de déplacer à l'extérieur et paralléliser une boucle correspondant à une ligne nulle, et préciser quelles sont les dépendances qui sont satisfaites après le choix d'une boucle, son déplacement à l'extérieur et sa séquentialisation.

▷ **Question 10** Pensez-vous que cet algorithme conduise à un nombre maximal de boucles parallèles (utiliser l'exemple avec  $D_3$ ) ?

▷ **Question 11** Montrer que trouver le nombre maximal de boucles parallèles est NP-complet, par réduction à partir de SET-COVER : étant donné un ensemble  $X$  à  $n$  éléments,  $K$  sous-ensembles de  $X$  et une borne  $B$ , peut-on trouver  $B$  sous-ensembles parmi les  $K$  tels que tout élément de  $X$  appartienne à au moins l'un de ces  $B$  sous-ensembles ?