Analyse de dépendances

Résumé: Dans ce TD, nous allons nous familiariser avec l'analyse de dépendances et l'utilisation de l'algorithme d'Allen et Kennedy.

1 Analyse de dépendances

▷ **Question 1.** Effectuer l'analyse de dépendance sur le code suivant :

$S_1 : A = B + C$
$S_2 : D = E + F$
$S_3 : C = A + C$
$S_4 : E = C + D$
$S_5 : E = 1$

Réponse. Pour qu'il y ait dépendance sur une variable entre deux instructions il faut qu'au moins une des deux soit une écriture sur cette variable. Considérons donc la première instruction. Elle est effectue une écriture sur A. La seule autre instruction qui utilise A est S_1. Comme c'est une lecture et qu'elle intervient après S_1, c'est une dépendance de flôt.
On recommence avec S_2 qui modifie D. La seule autre instruction faisant intervenir D est S_4 et on a donc une dépendance de flôt pour les mêmes raisons que précédemment. En traitant ainsi toutes les instructions dans l'ordre, on obtient les dépendances suivantes :
- Flot $S_1 \to S_3$: variable A.
- Flot $S_2 \to S_4$: variable D.
- Anti $S_1 \to S_3$: variable C.
- Flot $S_3 \to S_4$: variable C.
- Anti $S_2 \to S_4$: variable E.
- Anti $S_2 \to S_5$: variable E (recouverte).
- Sortie $S_4 \to S_5$: variable E.

□

▷ **Question 2.** Effectuer l'analyse de dépendance du programme suivant :

Pour $i = 1$ **à** N
  Pour $j = 1$ **à** N
    $S_1 : a(i+1, j-1) = b(i, j+4) + c(i-2, j-3) + 1$
    $S_2 : b(i-1, j) = a(i, j) - 1$
    $S_3 : c(i, j) = a(i, j-2) + b(i, j)$

Réponse. On a les dépendances suivantes : Recommençons le même type d'analyse que précédemment. S_1 modifie la variable $a(i+1, j-1)$. S_2 lisant $a(i, j)$, il y une dépendance entre ces deux instructions. La première composante du vecteur d'itération devant être positive, on a une dépendance de S_1 vers S_3 et le vecteur de dépendance est $(1, -1)$. L'écriture se faisont donc avant la lecture, c'est une dépendance de flot. On obtient donc les dépendances suivantes :
- Flot $S_1 \to S_2$: variable a, distance $(1, -1)$
- Flot $S_1 \to S_3$: variable a, distance $(1, 1)$
- Flot $S_3 \to S_1$: variable c, distance $(2, 3)$
- Anti $S_1 \to S_2$: variable b, distance $(1, 4)$
- Anti $S_3 \to S_2$: variable b, distance $(1, 0)$

Observons le cas de S_2 qui modifie $b(i-1, j)$ et de S_1 qui utiliser $b(i, j+4)$. La première composante du vecteur d'itération devant être positive, on a une dépendance de S_1 vers S_2 et le vecteur de dépendance est $(1, 4)$. Comme l'écriture a lieu après la lecture, on a une dépendance anti. □

2 Parallélisation de boucles : Algorithme d'Allen et Kennedy

On rappelle que l'algorithme d'Allen et Kennedy permet de paralléliser un nid de boucles. On note n_S nombre de boucles englobant une instruction S. Dans le Graphe de Dépendances Réduit des Niveaux G, on note $l_{\min}(G)$ le niveau minimal des arêtes de G.

Pour paralléliser le nid de boucles, il suffit donc d'appeler la fonction suivante avec comme argument $k = 1$ et G le GDRN du code à paralléliser.

Allen-Kennedy(G, k)
Supprimer dans G toutes les arêtes de niveau $< k$
Calculer les composantes fortement connexes (CFC) G
Pour tout CFC C dans l'ordre topologique
    Si C est réduit à une seule instruction S, sans arête, **Alors**
        Générer des boucles **DOPAR** dans toutes les dimensions restantes, i.e. niveaux k à n_S, et générer le code pour S.
    Sinon
        $l \leftarrow l_{\min}(C)$
        Générer des boucles **DOPAR** boucles du niveau k au niveau $l-1$, et une boucle **DOSEQ** pour le niveau l.
        Allen-Kennedy($C, l+1$)

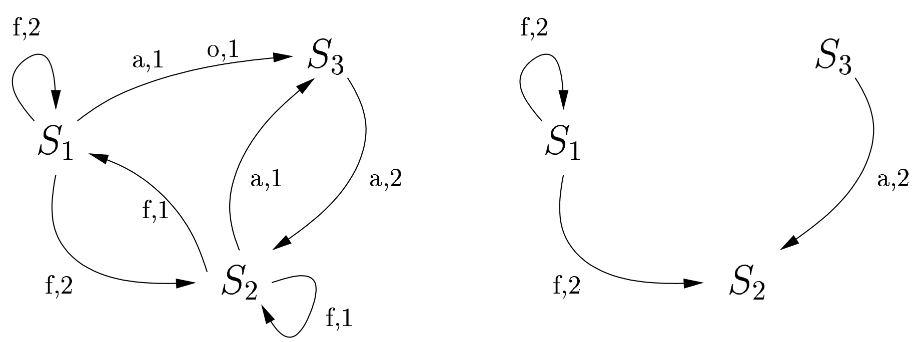
▷ **Question 3.** Donner le graphe de dépendances réduit avec le type de dépendances (flot, anti, sortie) de leur expression et leur niveau.

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

Appliquer l'algorithme de Kennedy et Allen pour restructurer le nid de boucles et vérifier la nature des boucles obtenues.

Réponse. En effectuant le même type d'analyse que précédemment, on obtient les dépendances suivantes :
- Flot $S_1 \to S_1$: variable a, distance $(0, 1)$
- Flot $S_1 \to S_2$: variable a, distance $(0, 2)$
- Flot $S_2 \to S_1$: variable b, distance $(1, -2)$
- Flot $S_2 \to S_2$: variable b, distance $(1, 1)$
- Anti $S_1 \to S_3$: variable a, distance $(1, -2)$
- Anti $S_2 \to S_3$: variable a, distance $(1, -3)$
- Anti $S_3 \to S_2$: variable b, distance $(0, 1)$
- Sortie $S_1 \to S_3$: variable a, distance $(1, -1)$

Le GDR avec les niveaux correspondant est dessiné sur la gauche :

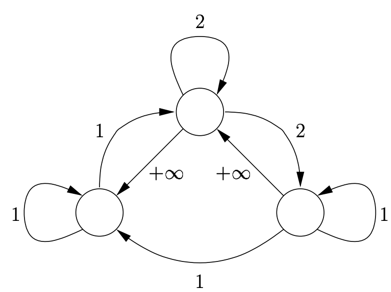


Le GDR est fortement connexe et il y a des dépendances de niveau 1. La boucle sur i sera donc séquentielle. En passant à la deuxième étape, on obtient le GDRN de droite sur la figure. L'algorithme de Kennedy et Allen conduit alors au nid restructuré suivant :

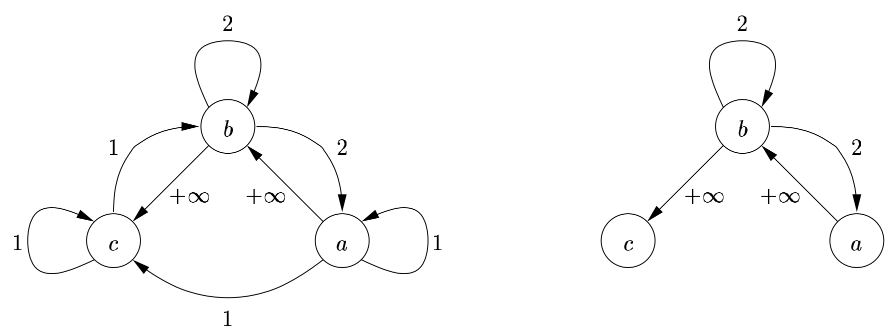
Pour $i = 1$ **à** N
  Pour $j = 1$ **à** N
    $S_1 : a(i+1, j+1) = a(i+1, j) + b(i, j+2)$
  Pour $j = 1$ **à** N **en parallèle**
    $S_3 : a(i, j+2) = b(i+1, j+1) - 1$
  Pour $j = 1$ **à** N **en parallèle**
    $S_2 : b(i+1, j) = a(i+1, j-1) + b(i, j-1)$

On notera cependant qu'il est également parfaitement possible d'intervertir la boucle portant S_3 avec celle portant S_1. □

▹ **Question 4.** Donner un nid de boucles parfait où toutes les dépendances sont uniformes, et dont le graphe de dépendances avec niveaux de dépendances est exactement le graphe suivant :



Exécuter l'algorithme d'Allen et Kennedy sur ce nid de boucles.
Réponse.
Pour $i = 1$ **à** N
  Pour $j = 1$ **à** N
    $S_1 : a(i, j) = a(i-1, j) + b(i, j-1)$
    $S_2 : b(i, j) = a(i, j) + b(i, j-1) + c(i-1, j)$
    $S_3 : c(i, j) = a(i-1, j) + b(i, j) + c(i-1, j)$

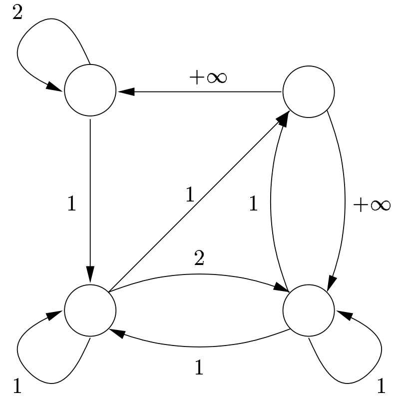


L'algorithme de Kennedy et Allen conduit au nid restructuré suivant :

Pour $i = 1$ **à** N
  Pour $j = 1$ **à** N
    $S_1 : a(i, j) = a(i-1, j) + b(i, j-1)$
    $S_2 : b(i, j) = a(i, j) + b(i, j-1) + c(i-1, j)$
  Pour $j = 1$ **à** N **en parallèle**
    $S_3 : c(i, j) = a(i-1, j) + b(i, j) + c(i-1, j)$

□

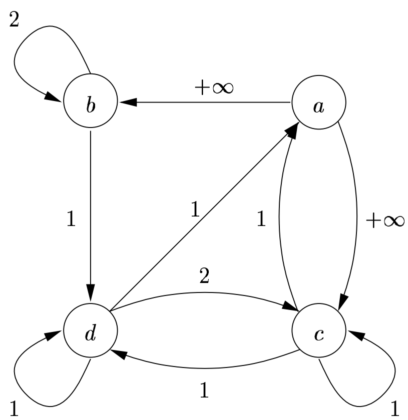
▹ **Question 5.** Donner un nid de boucles parfait où toutes les dépendances sont uniformes, et dont le graphe de dépendances avec niveaux de dépendances est exactement le graphe suivant :

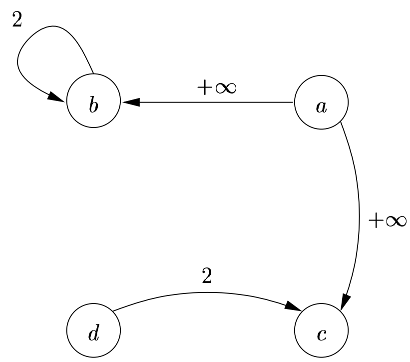


Exécuter l'algorithme d'Allen et Kennedy sur ce nid de boucles.

Réponse. On peut construire le code suivant :

Pour $i = 1$ **à** N
    Pour $j = 1$ **à** N
        $S_1 : a(i,j) = d(i-1,j) + c(i-1,j)$
        $S_2 : b(i,j) = a(i,j) + b(i,j-1)$
        $S_3 : c(i,j) = a(i,j) + c(i-1,j) + d(i,j-1)$
        $S_4 : d(i,j) = c(i-1,j) + d(i-1,j) + b(i-1,j)$





L'algorithme de Kennedy et Allen conduit au nid restructuré suivant :

Pour $i = 1$ **à** N
    Pour $j = 1$ **à** N **en parallèle**
        $S_4 : d(i,j) = c(i-1,j) + d(i-1,j) + b(i-1,j)$
    Pour $j = 1$ **à** N **en parallèle**
        $S_1 : a(i,j) = d(i-1,j) + c(i-1,j)$
    Pour $j = 1$ **à** N **en parallèle**
        $S_3 : c(i,j) = a(i,j) + c(i-1,j) + d(i,j-1)$
    Pour $j = 1$ **à** N
        $S_4 : d(i,j) = c(i-1,j) + d(i-1,j) + b(i-1,j)$

$\square$

3 Factorisation de Cholesky

La factorisation de Cholesky permet de décomposer une matrice symétrique

Pour $k = 1$ **à** N
$S_1 : a(k,k) := \sqrt{a(k,k)}$
Pour $i = k+1$ **à** N
$S_2 : a(i,k) := a(i,k)/a(k,k)$
Pour $j = k+1$ **à** N
$S_3 : a(i,j) := a(i,j) - a(i,k) \times a(j,k)$

▷ **Question 6.** Construire à la main, en étudiant l'algorithme, le graphe de tâches étendu (dont les sommets sont les instances des instructions).

Réponse. □

▷ **Question 7.** Montrer qu'il n'y a que des dépendances de flot (i.e. toutes les autres correspondent à des arcs de transitivité).

▷ **Question 8.** Calculer combien de paires de références devrait considérer un analyseur de dépendances.

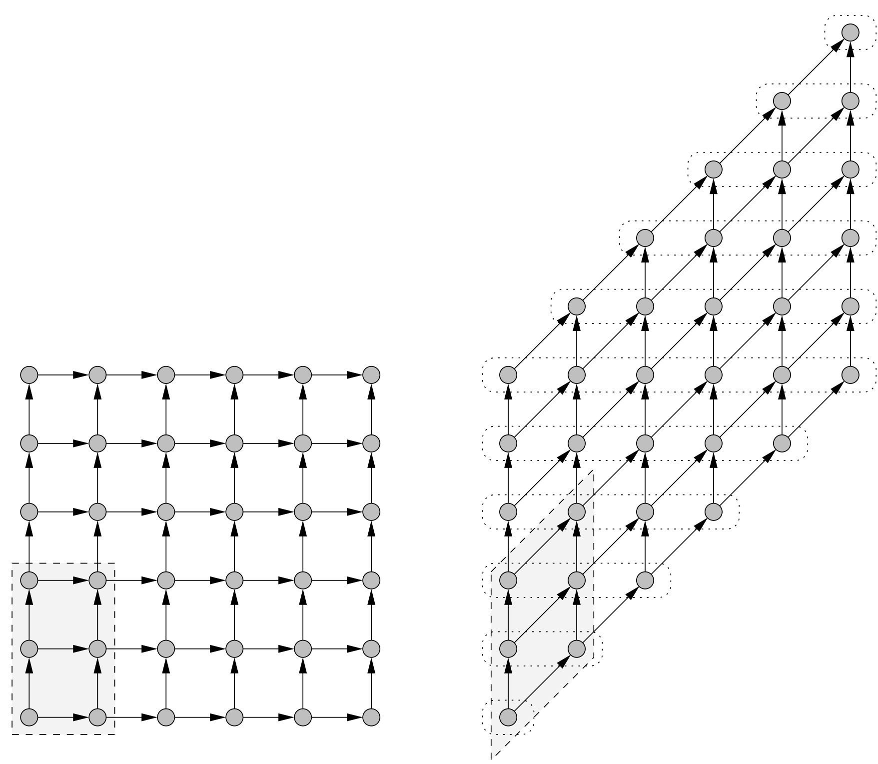
▷ **Question 9.** Faire l'analyse de dépendance sur une dépendance de sortie et montrer comment elle est recouverte.

4 Méthode aux différences finies

Pour $i = 1$ **à** N
Pour $j = 1$ **à** N
$a(i,j) = \frac{1}{4}[a(i-1,j) + a(i+1,j) + a(i,j-1) + a(i,j+1)]$

▷ **Question 10.** Donner le GDR avec le type des dépendances (flot, anti, sortie) et le GDE pour $N = 5$. Imaginer une torsion de l'espace d'itération qui permette d'extraire une boucle parallèle.

Réponse. On peut effectuer la torsion suivante :



□