

# Élimination de dépendances

**Résumé:** Dans ce TD, nous allons étudier une méthode permettant d'éliminer certaines dépendances à l'aide de l'introduction de temporaires.

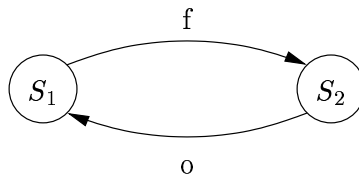
## 1 Un petit exemple simple pour commencer

▷ **Question 1.** Calculer le Graphe de Dépendances Réduit du code suivant :

**Pour**  $i = 1$  à  $N$   
 $S_1 : a(i) = b(i) + c(i)$   
 $S_2 : a(i + 1) = a(i) + 2d(i)$

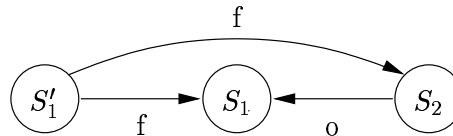
Essayer de casser le cycle en ajoutant une variable temporaire.

*Réponse.* Dans le code original, il y a une dépendance de sortie de  $S_2$  vers  $S_1$  car  $a(i + 1)$  est écrit dans  $S_2$  avant d'être ré-écrit dans  $S_1$  à l'itération suivante. Il y a également une dépendance de flot de  $S_1$  vers  $S_2$  à cause de  $a(i)$ . On obtient donc le GDR suivant :



L'ajout d'un temporaire permet d'éliminer ces cycles :

**Pour**  $i = 1$  à  $N$   
 $S'_1 : temp(i) = b(i) + c(i)$   
 $S_1 : a(i) = temp(i)$   
 $S_2 : a(i + 1) = temp(i) + 2d(i)$



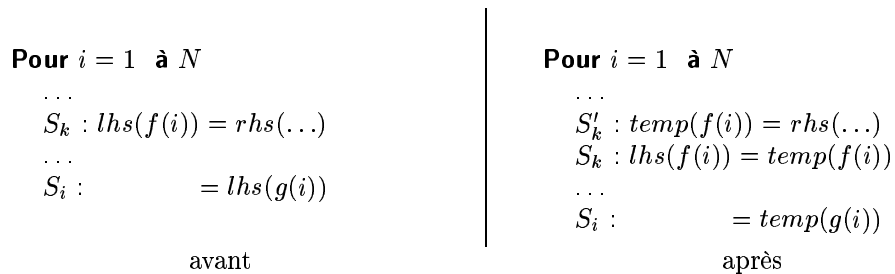
Le nouveau nid n'a pas de cycle et peut être vectorisé par exemple de la façon suivante :

**Pour**  $i = 1$  à  $N$  **en parallèle**  
 $S'_1 : temp(i) = b(i) + c(i)$   
 $a(1) = temp(1)$   
**Pour**  $i = 1$  à  $N$  **en parallèle**  
 $S_2 : a(i + 1) = temp(i) + 2d(i)$

□

## 2 Du cas particulier au cas général

▷ **Question 2.** Dans le cas général, la découpe de sommet s'applique ainsi :



Montrer ce que deviennent respectivement les six types de dépendance possibles (dépendances en entrée sur  $S$  de type flot, anti, output et dépendances en sortie de  $S$  de type flot, anti, output) sur le nouveau graphe avec  $S$  et  $S'$  (on supposera que la fonction d'accès  $lhs$  est injective).

*Réponse.* On obtient les transformation suivantes :

- **Dépendance de flot en entrée** (Figure 1). Supposons qu'une des données lue dans la partie droite de l'instruction  $S_k$  ait été produite en partie gauche de l'instruction  $S_i$ . Après la séparation, la donnée est lue en partie gauche de l'instruction  $S'_k$  et il y a donc une dépendance de flot de  $S_i$  vers  $S'_k$ . Il y a également une nouvelle dépendance de flot entre  $S'_k$  et  $S_k$ .

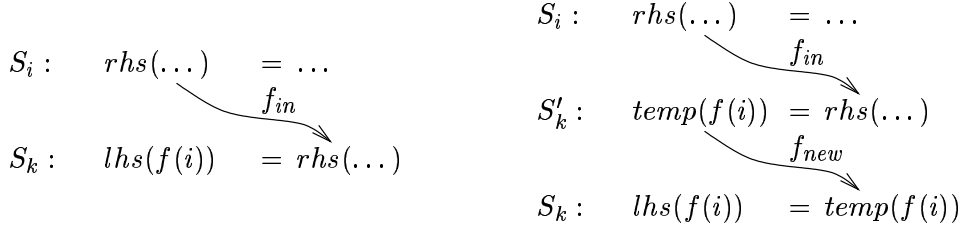


FIG. 1 – Dépendance de flot en entrée avant et après la transformation.

- **Anti-dépendance en entrée** (Figure 2). Supposons qu'une instruction  $S_i$  lise  $lhs(f(i))$  avant que  $S_k$  l'écrive. Après la transformation,  $lhs(f(i))$  est toujours lue par  $S_i$  et est toujours écrite par  $S_k$ . Ainsi, les anti-dépendances de  $S_i$  vers  $S_k$  sont inchangées.

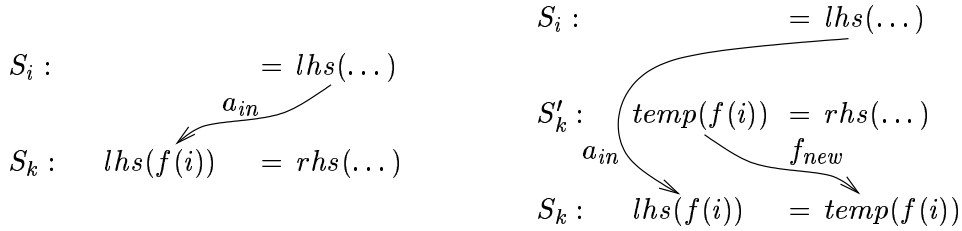


FIG. 2 – Anti-dépendance d'entrée avant et après la transformation.

- **Dépendance de sortie en entrée** (Figure 3). Supposons qu'une instruction  $S_i$  écrive  $lhs(f(i))$  avant que  $S_k$  ne l'écrive. Après la transformation,  $lhs(f(i))$  est toujours écrite par  $S_i$  et est toujours écrite par  $S_k$ . Ainsi, les dépendances de sortie de  $S_i$  vers  $S_k$  sont inchangées.

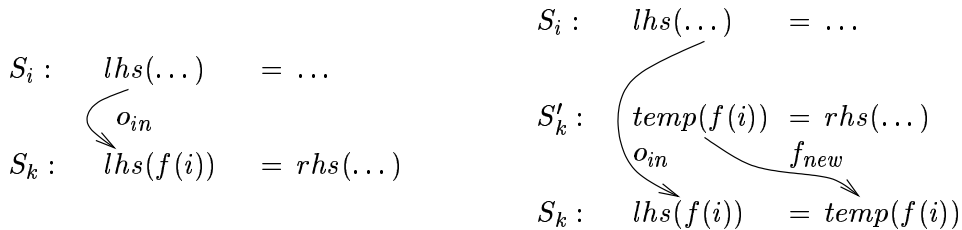


FIG. 3 – Dépendance de sortie en entrée avant et après la transformation.

- **Dépendance de flot en sortie** (Figure 4). Supposons qu'une instruction  $S_i$  lise la valeur de  $lhs(f(i))$  produite par l'instruction  $S_k$ . Ainsi, l'accès à  $lhs$  dans l'instruction  $S_i$  ( $lhs(g(i))$ ) a été remplacé par un accès à  $temp$  ( $temp(g(i))$ ). Ainsi, comme  $temp(f(i))$  est écrit par  $S'_k$ , il y a une dépendance de flot de  $S'_k$  vers  $S_i$ .

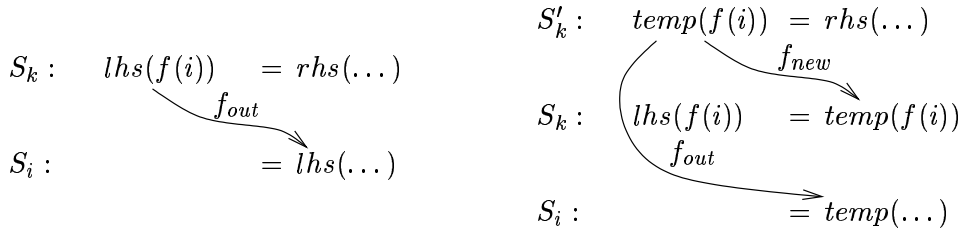


FIG. 4 – Dépendance de flot en sortie avant et après la transformation.

- **Anti-dépendance en sortie** (Figure 5). Supposons qu'une des données lue dans la partie droite de l'instruction  $S_k$  soit écrite après coup en partie gauche de l'instruction  $S_i$ . Après la transformation, la donnée est lue en partie droite de l'instruction  $S'_k$  et il y a donc une anti-dépendance de flot de  $S'_k$  vers  $S_i$ .

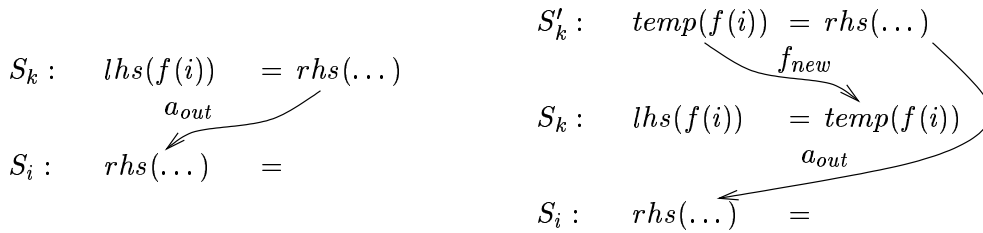


FIG. 5 – Anti-dépendance en sortie avant et après la transformation.

- **Dépendance de sortie en sortie** (Figure 6). Supposons qu'une instruction  $S_i$  écrive la valeur de  $lhs(f(i))$  après qu'elle ait été écrite par l'instruction  $S_k$ . Après la transformation,  $lhs(f(i))$  est toujours écrite par  $S_i$  puis par  $S_k$ . Il y a donc toujours une dépendance de sortie de  $S_k$  vers  $S_i$ .

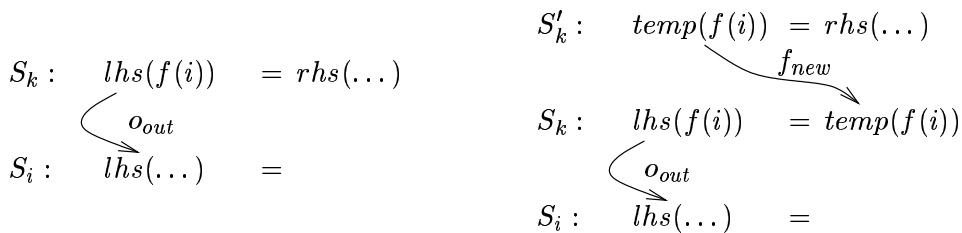


FIG. 6 – Dépendance de sortie en sortie avant et après la transformation.

Une fois la découpe du sommet  $S_k$  effectuée, on obtient donc le graphe de dépendances réduit suivant de la figure 7 □

Le théorème suivant démontre l'intérêt de cette transformation :

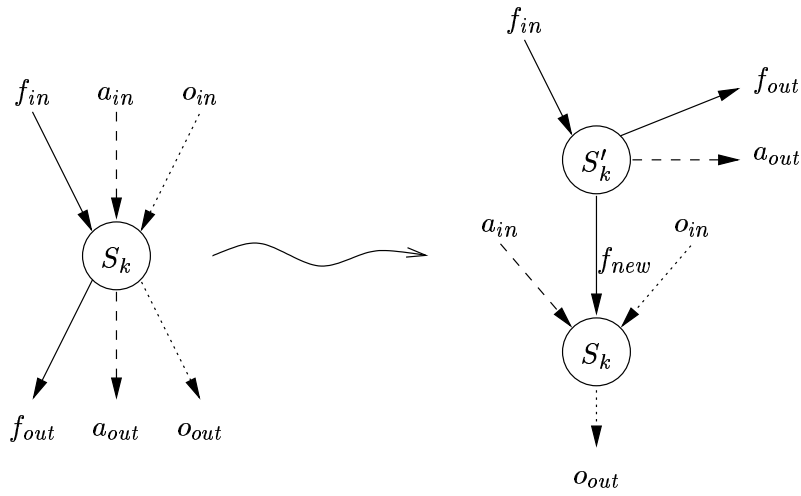


FIG. 7 – Transformation des dépendances d’une instruction  $S$  lors de sa découpe.

*Théorème 1.* Soit  $G$  le graphe de dépendances réduit d’un nid de boucles  $L$  et soit  $G'$  le graphe obtenu à partir de  $G$  en découpant tous ses sommets. Alors un cycle  $C$  de  $G'$  n’est composé soit que de dépendances de flot, soit que de dépendances de sortie. De plus, tout cycle  $C$  de  $G'$  correspond à un cycle déjà existant dans  $G$ .

► **Question 3.** Démontrer le théorème précédent.

*Réponse.* La figure 7 permet de mieux suivre la démonstration. Soit  $C$  un cycle de  $G'$ . Considérons une arête  $e$  de  $C$ .  $e$  correspond soit à une dépendance de flot, soit à une anti-dépendance, soit à une dépendance de sortie :

- Si  $e$  correspond à **dépendance de sortie**, alors, d’après la figure 7,  $e$  est une arête de  $S_k$  vers  $S_i$ . Les seules arêtes sortant de  $S_i$  étant des arêtes représentant des dépendances de sortie, l’arête suivant  $e$  dans le cycle  $C$  est une arête de dépendance de sortie. Ainsi,  $C$  est uniquement composée d’arêtes représentant des dépendances de sortie. De plus, toutes les arêtes de  $C$  sont des arêtes qui étaient déjà présentes dans  $G$ . La dissection des sommets n’introduit donc pas de nouveau cycle.
- Si  $e$  correspond à une **anti-dépendance**, alors  $e$  va d’un sommet  $S'_k$  à un sommet  $S_i$ . Les seules arêtes sortant de  $S_i$  étant des dépendances de sortie, l’arête suivant  $e$  dans  $C$  est une arête de dépendance de sortie. En utilisant le raisonnement précédent, on en déduit donc que  $C$  est composée uniquement d’arêtes représentant des dépendances de sortie, ce qui entre en contradiction avec la nature de  $e$ . Aucun cycle ne peut donc contenir d’arête représentant une anti-dépendance.
- Si  $e$  représente une **dépendance de flot**, alors soit  $e$  est une arête créée lors de la dissection des sommets et va donc d’un sommet  $S'_k$  vers un sommet  $S_k$ , soit  $e$  va d’un sommet  $S'_k$  vers un somme  $S'_i$  :
  - $e : S'_k \xrightarrow{f_{new}} S_k$ . Les seules arêtes provenant d’un sommet  $S_k$  étant des dépendances de sortie, l’arête suivant  $e$  dans  $C$  est une dépendance de sortie. On en déduit donc que  $C$  est composée uniquement d’arêtes représentant des dépendances de sortie, ce qui entre en contradiction avec la nature de  $e$ .
  - $e : S'_k \xrightarrow{f} S'_i$ . Il peut y avoir une dépendance de flot ou une anti-dépendance en provenance d’un sommet  $S'_l$ . Cependant, l’arêtes qui suit  $e$  dans  $C$  ne peut être une anti-dépendance (on a vu qu’il ne pouvait y avoir d’anti-dépendance dans un cycle). L’arête qui suit  $e$  dans  $C$  représente donc nécessairement une dépendance de flot.  $C$  est donc uniquement composé de dépendances de flot. De plus, ces arêtes ne correspondent pas à des dépendances de flot créées lors de la dissection des sommet. La dissection n’a donc pas introduit de nouveau cycle.

Les cycles composés uniquement dépendances de flot ou uniquement de dépendance de sortie ne sont donc pas brisés lors de la dissection des sommets mais tous les autres cycles sont rompus.  $\square$

► **Question 4.** Appliquez les techniques précédentes au code suivant :

Pour  $i = 4$  à  $N$

$$S_1 : a(i + 5) = c(i - 3) + b(2i + 2)$$

$$S_2 : b(2i) = a(i - 1) + 1$$

$$S_3 : a(i) = c(i + 5) + 1$$

$$S_4 : c(i) = b(2i - 4)$$

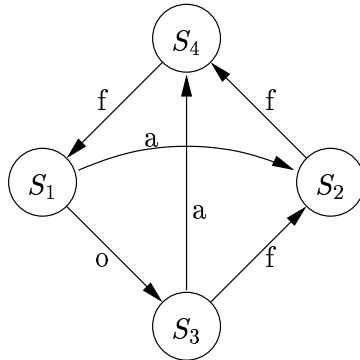


FIG. 8 – Le graphe de dépendance avant sa transformation.

*Réponse.* Le graphe de dépendances est représenté sur la figure 8. Il y a six dépendances dans la boucle :

**3 dépendances de flot** de  $S_3$  vers  $S_2$  (à cause de  $a$ ), de  $S_2$  vers  $S_4$  (à cause de  $b$ ) et de  $S_4$  vers  $S_1$  (à cause de  $c$ ),

**2 anti-dépendances** de  $S_1$  vers  $S_2$  (à cause de  $b$ ) et de  $S_3$  vers  $S_4$  (à cause de  $c$ ),

**1 dépendance de sortie** de  $S_1$  vers  $S_3$  (à cause de  $a$ ).

La dissection des sommets  $S_2$  et  $S_3$  du graphe de dépendance conduit au graphe de dépendance de la figure 9. On peut donc réécrire le nid de boucle en introduisant les variables temporaires  $a_{temp}$  (pour la dissection de

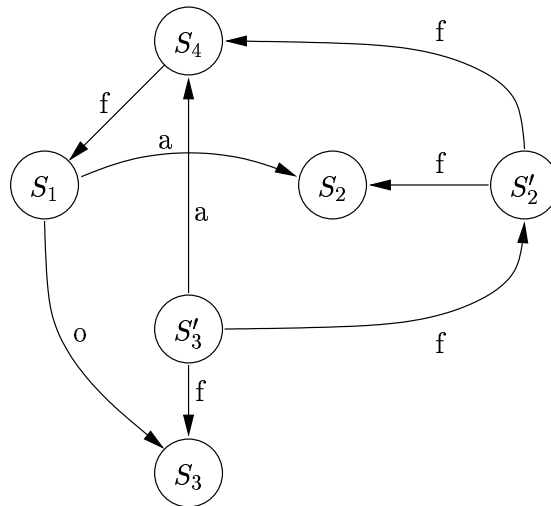


FIG. 9 – Le graphe de dépendance après sa transformation.

$S_3$ ) et  $b_{temp}$  (pour la dissection de  $S_2$ ).

**Pour**  $i = 4$  à  $N$

$$S_1 : a(i + 5) = c(i - 3) + b(2i + 2)$$

$$S_2' : b_{temp}(2i) = \begin{cases} \mathbf{Si} \ i \geq 5 \ \mathbf{Alors} \ a_{temp}(i - 1) + 1 \\ \mathbf{Sinon} \ a(i - 1) + 1 \end{cases}$$

$$S_2 : b(2i) = b_{temp}(2i)$$

$$S_3' : a_{temp}(i) := c(i + 5) - 1$$

$$S_3 : a(i) = a_{temp}(i)$$

$$S_4 : c(i) = \begin{cases} \mathbf{Si} \ i \geq 6 \ \mathbf{Alors} \ b_{temp}(2i - 4) \\ \mathbf{Sinon} \ b(2i - 4) \end{cases}$$

Les affectations conditionnelles sont nécessaires si on veut prendre en compte des dépendances provenant de l'extérieur du nid.  $\square$