

Exercices reliés au chapitre 9

Exercices

Voici les exercices que je recommande de faire :

- **Exercices 9.1.4, 9.1.1, 9.1.2.** (L'exercice 10.1 de la 1ère édition est similaire.) Il est recommandé de commencer par 9.1.4 car c'est le plus facile.

Note : au numéro 9.1.1, les instructions de branchement sont omises. Assumez simplement qu'il s'agit de graphe du flot de contrôle d'un programme de la forme suivante :

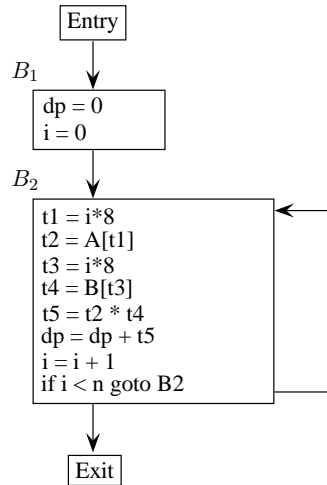
```
for(...)  
{  
    for(...)  
    {  
        // ...  
    }  
}
```

De plus, au numéro 9.1.1 (c) et (d), effectuez les optimisations si possible après avoir identifié les informations demandées.

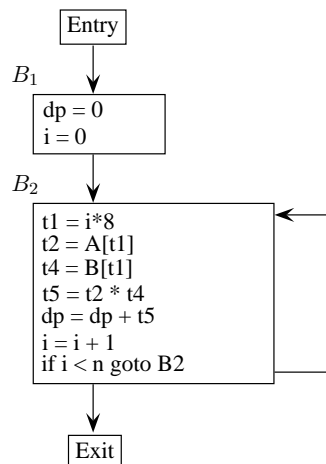
Réponses

9.1.4

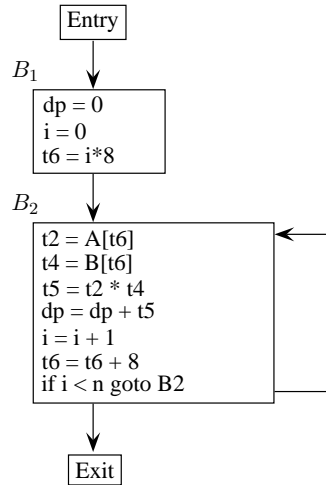
On commence par identifier les blocs de base et faire le diagramme de flot de contrôle :



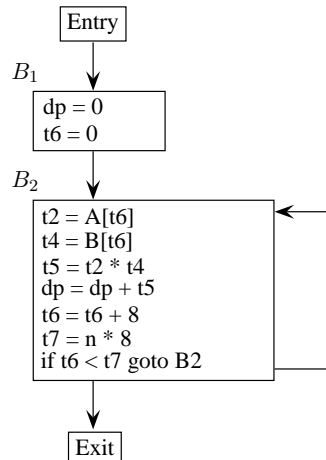
- **Expression commune.** t_1 et t_3 sont identiques, et t_1 et i ne sont pas modifiées lorsqu'on atteint t_3 . On peut donc remplacer $t_3 = i * 8$ par $t_3 = t_1$. De plus, par **propagation de copie**, on remplace les occurrences de t_3 par t_1 (car t_1 n'est pas modifiée entre son initialisation et les endroits où t_3 est utilisée). Cette opération rend la variable t_3 inutile et on la retire.



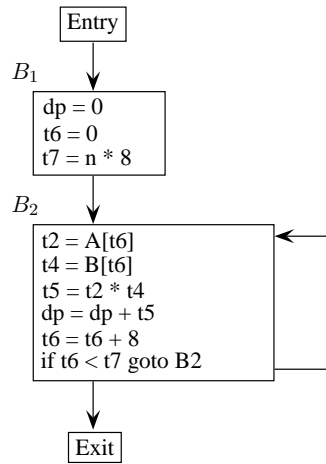
- **Affaiblissement.** On peut retirer la multiplication $i * 8$ et la remplacer par t_6 , qu'on incrémentera de 8 pour chaque incrémentation de i . La variable t_1 devient morte et on la retire.



- **Propagation de copie.** On remplace $t_6 = i * 8$ par $t_6 = 0 * 8$, car on a la certitude que i a la valeur de 0 à cet endroit. Par arithmétique, on simplifie ceci en $t_6 = 0$.
- **Variables d'induction.** On a deux variables d'induction (i et t_6). Utilisons $i = t_6/8$ pour retirer la variable i de la boucle (le $i = 0$ de B_1 est alors mort et retiré). La comparaison $i < n$ devient $t_6 < n * 8$.

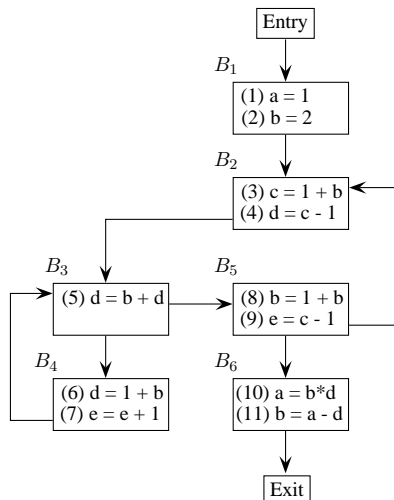


- **Déplacement de code.** $t_7 = n * 8$ peut être déplacé hors de la boucle car la valeur de n ne change pas, et cette expression n'a aucun effet de bord.



9.1.1

- (a)
 - B_3, B_4
 - B_2, B_3, B_4, B_5
- (b) Tous les blocs jusqu'à B_6 ne font que lire a , sans la modifier ; nous pouvons donc effectuer la propagation de copie dans les blocs B_1 à B_5 :

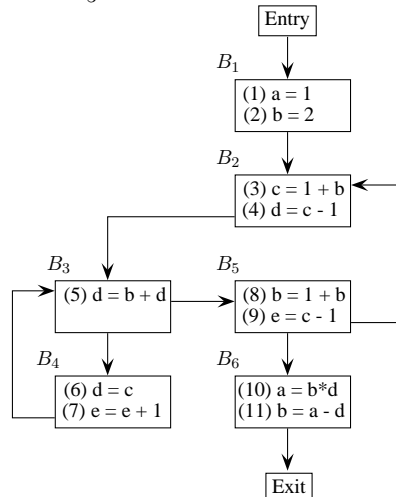


Notez que la variable b est modifiée par B_5 , nous ne pouvons donc pas propager la valeur de b .

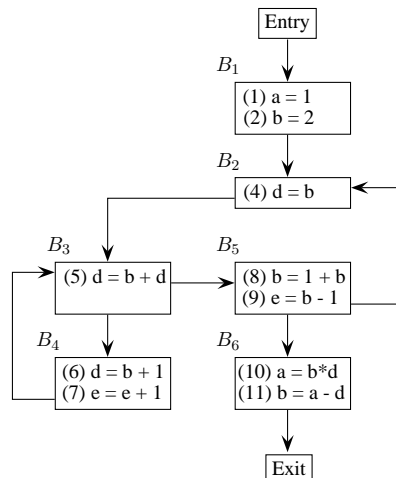
- (c) Les expressions communes sont :
 - $1 + b$
 - $c - 1$

On peut remplacer $d = 1 + b$ (bloc B_4) par $d = c$, car les valeurs de b et de c ne changent pas entre le bloc B_2 et le bloc B_4 .

Pour le cas de $c - 1$ (répétée au bloc B_5), il n'est pas possible d'effectuer l'optimisation, car la valeur de d change entre l'assignation $d = c - 1$ (au bloc B_2) et la réutilisation de $c - 1$ au bloc B_5 .



- (d)
 - Dans la boucle $B_2 - B_5$, b et c sont des variables d'inductions; en effet nous avons *généralement* $c = 1 + b$ (sauf entre (8) et (3), où $b = c$)
- Nous pouvons tenter d'optimiser en retirant la variable c .

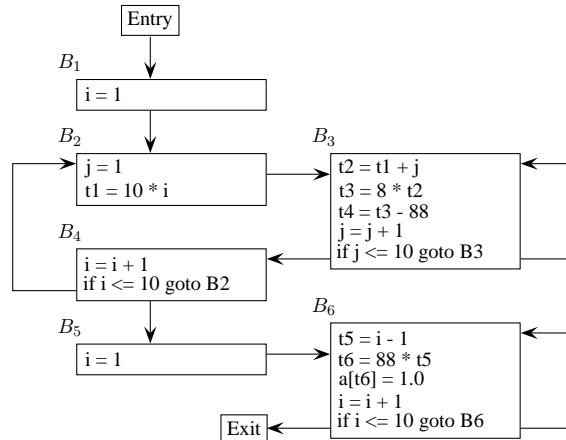


Notez que dans ce cas il n'est pas certain que cette optimisation soit bonne, puisque nous ajoutons ainsi une addition dans la boucle $B_3 - B_4$; la simple assignation qui s'y trouvait auparavant était vraisemblablement plus rapide.

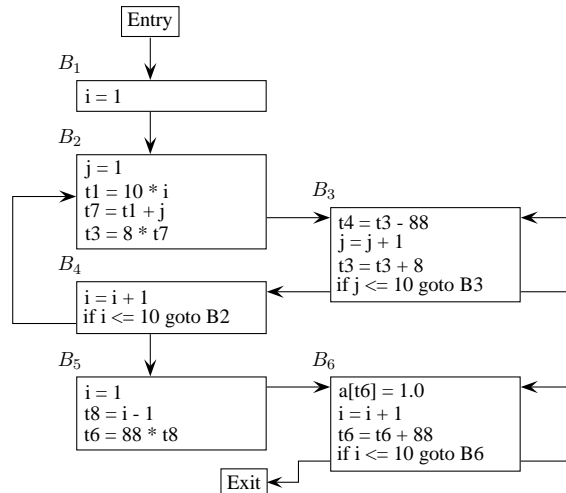
- (e) Dans le diagramme original, ou dans la tentative d'optimisation du numéro (d), on a l'invariant $d = b + 1$ (dans la boucle $B_3 - B_4$).

9.1.2

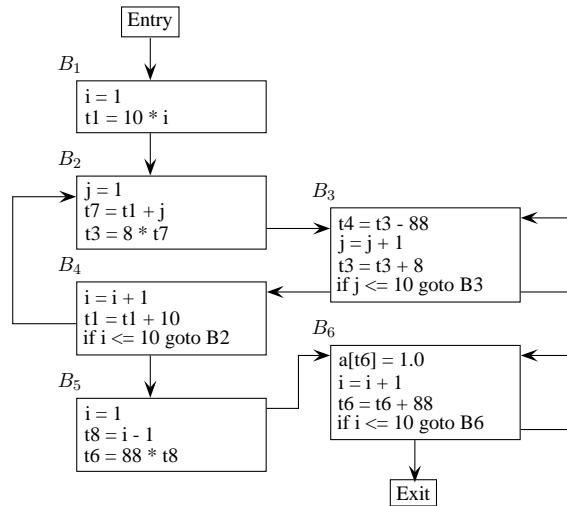
- **Déplacement de code.** B_3 calcule 10 fois l'expression $10 * i$ alors que la valeur de i ne change pas. On déplace donc le calcul de t_1 hors de la boucle interne.



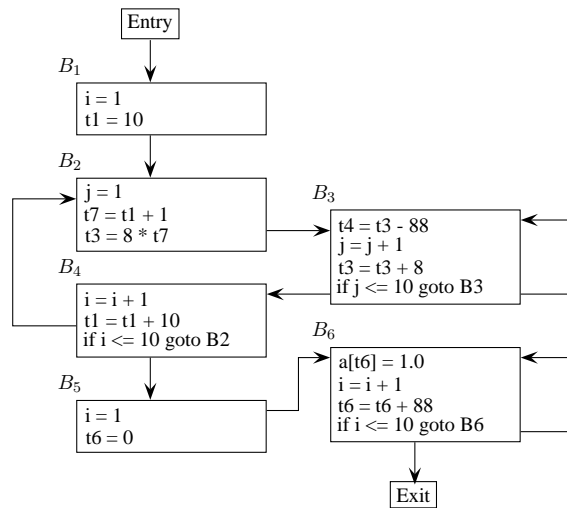
- **Affaiblissement.** $t_3 = 8 * t_2$ peut être transformé pour utiliser une instruction moins coûteuse que la multiplication. Après cette modification, t_2 n'est plus utilisée et est retirée. De même, dans le bloc B_6 , l'expression $t_6 = 88 * t_5$ peut subir le même traitement.



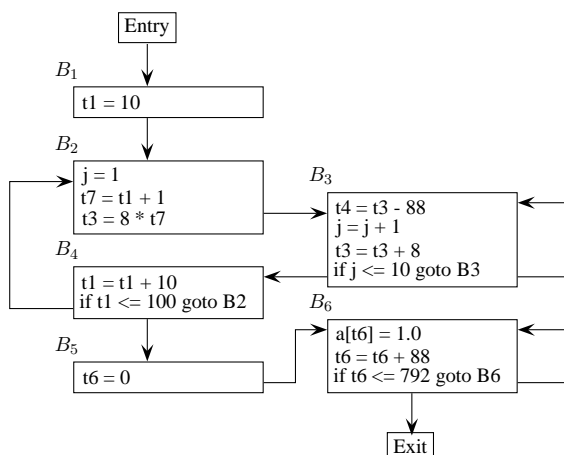
- **Affaiblissement.** $t_1 = 10 * i$ peut être transformé pour utiliser une instruction moins coûteuse que la multiplication.



- **Propagation de valeur.** $t_1 = 10 * i$ (B_1) devient $t_1 = 10$ car $i = 1$; $t_7 = t_1 + j$ (B_2) devient $t_7 = t_1 + 1$ car $j = 1$; $t_8 = i - 1$ (B_5) devient $t_8 = 0$ car $i = 1$. Ensuite, on propage $t_8 = 0$ dans $t_6 = 88 * t_8$, et par simplification arithmétique on trouve $t_6 = 0$. t_8 est alors une variable morte et peut être retirée.



- **Variables d'induction.** Les variables i et t_1 sont liées dans la première boucle. On retire i de cette boucle pour ne conserver que t_1 . De même, dans B_5 et B_6 , i et t_6 évoluent de façon proportionnelle; on élimine là aussi la variable i et on ne conserve que t_6 (quand $i = 10$, $t_6 = 792$).



- **Arithmétique.** Enfin, on optimise $8 * t_7$ (B_2) en $t_7 \ll 3$.