Allocation de registres par coloriage de graphes

David Delahaye

David.Delahaye@lirmm.fr

Faculté des Sciences

Master M1 2020-2021

De l'allocation de registres au coloriage de graphes

Graphe d'interférences

Nous avons construit un graphe d'interférences dont :

- Les sommets sont les pseudo-registres et les registres physiques allouables ($v0-$v1, $a0-$a3, $ra, $t0-$t9, $s0-$s7) ;
- Une arête d'interférence relie deux sommets qui doivent recevoir des emplacements distincts ;
- Une arête de préférence ou arête « move » relie deux sommets à qui on souhaiterait attribuer le même emplacement.

De l'allocation de registres au coloriage de graphes

Coloriage de graphes

Supposons que l'on dispose de k registres physiques allouables. Alors le problème de l'allocation de registres semble se résumer à :

- Attribuer une couleur parmi k à chaque sommet représentant un pseudo-registre ;
- De façon à ce que deux sommets reliés par une arête d'interférence ne reçoivent jamais la même couleur ;
- Et si possible de façon à ce que deux sommets reliés par une arête de préférence reçoivent la même couleur.

Le graphe est dit k-colorable si ce nouveau problème admet une solution.

Historique et problèmes

Historique

- L'idée de réduire l'allocation de registres au coloriage de graphes date des années 1960, mais a été mise en pratique pour la première fois par Chaitin en 1981 ;
- Ce cadre théorique est attirant de par sa simplicité, mais quelques problèmes demeurent.

Premier problème

- Le problème du coloriage de graphes est NP-complet, d'où, en pratique, impossibilité de construire une solution optimale.

En réponse, tous les compilateurs actuels utilisent des heuristiques de complexité linéaire ou quasi-linéaire.

Historique et problèmes

Deuxième problème

- Si le graphe n'est pas k-colorable ou si on ne trouve pas de k-coloriage, que faire ?

L'idée la plus simple est de permettre à certains sommets de rester non colorés et de réaliser ensuite ces pseudo-registres par des emplacements de pile. On parle alors de « spill ».

Les détails de ce processus sont plus subtils qu'il n'y paraît, et ce problème, dans toute sa généralité, offre lui aussi un espace de choix colossal.

Troisième problème

- Certaines architectures existantes n'offrent pas k registres physiques indépendants et interchangeables.

Heureusement, on peut modifier l'algorithme de coloriage de graphes pour refléter les irrégularités et particularités les plus courantes.

Algorithmes de coloriage

Simplification

- Kempe (1879) et Chaitin (1981) ont observé qu'un sommet s de degré strictement inférieur à k est trivialement colorable : le graphe G est k-colorable si et seulement si G privé de s est k-colorable ;
- On peut répéter cette simplification autant de fois que possible. De plus, le fait de supprimer un sommet trivialement colorable peut rendre d'autres sommets trivialement colorables.

Algorithme de Chaitin

procédure Colorier(G)

si il existe un sommet s trivialement colorable

alors $\begin{cases} \text{Colorier}(G \backslash s) \\ \text{attribuer une couleur disponible à } s \end{cases}$

sinon s'il existe un sommet s alors $\begin{cases} \text{Colorier}(G \backslash s) \\ \text{« spiller » } s \end{cases}$

Algorithmes de coloriage

Choix des sommets

- Le choix d'un sommet trivialement colorable, lorsqu'il en existe plusieurs, n'est pas fondamental ;
- Le choix d'un sommet à « spiller » est critique :
  - Pour une meilleure efficacité, il faut choisir un pseudo-registre peu utilisé ou utilisé en des points peu critiques du code ;
  - Pour faciliter la suite du coloriage, il vaut mieux choisir un sommet de fort degré ;
  - On fait appel à une fonction de coût qui combine ces critères.

Algorithmes de coloriage

Emploi des registres « callee-save »

- Cette heuristique permet un bon emploi des registres « callee-save » ;
- En effet, les pseudo-registres introduits pour sauvegarder le contenu des registres « callee-save » sont peu utilisés (une écriture et une lecture) et ont une très longue durée de vie, donc un fort degré ;
- Ils seront donc « spillés » de préférence, ainsi, les registres « callee-save » seront sauvegardés à l'entrée, restaurés à la sortie, et disponibles entre les deux.

Algorithmes de coloriage

Choix des couleurs

- Le choix de la couleur attribuée à un sommet s trivialement colorable après coloriage de $G\backslash s$ est important : on a ici une occasion de respecter les souhaits exprimés par les arêtes de préférence ;
- Supposons t relié à s par une arête de préférence, on attribuera à s la couleur déjà attribuée à t, s'il est déjà coloré, ou bien une couleur encore permise pour t, s'il n'est pas encore coloré, etc. ;
- Cette technique de coloriage biaisé est simple mais limitée, le « coalescing » lui sera supérieur.

Exercice

Colorier le graphe d'interférences du programme avec $k = 3$ et $k = 2$

$v := 0;$
$y := z + t;$
$u := t;$
$z := x + y;$
$v := z$

On suppose qu'à la fin du programme, il n'y a aucune variable vivante.

Exercice

Colorier le graphe d'interférences du programme avec $k = 4$

$g = j + 12;$
$h = k - 1;$
$f = g \times h;$
$e = j + 8;$
$m = j + 16;$
$b = f;$
$c = e + 8;$
$d = c;$
$k = m + 4;$
$j = b$

On suppose qu'à la fin du programme, les variables vivantes sont d, k, et j.

Implantation

Code Java

- Implanter l'algorithme de coloriage de graphe ;
- Utiliser le coloriage optimiste (« spiller » que si nécessaire) ;
- Pas de code source fourni (à écrire « from scratch ») ;
- Définir sa propre structure de graphe.

Rendu

- Dernier rendu pour la partie compilation native (youpi ☺ !) ;
- Un unique fichier Java ;
- Date limite : **25 octobre 2020**.