

Optimisation en informatique (RCP104)
Projet 2009 : Compilation et Optimisation de l'allocation des registres

Objectif du projet

Un compilateur effectue une série de transformations pour générer un programme écrit dans un langage cible à partir d'un programme écrit dans un langage source. L'allocation de registres est une de ces transformations; c'est une des plus difficiles à mettre en oeuvre. Les registres servent à stocker les valeurs des variables du programme à compiler. Très souvent, dans un programme, il y a beaucoup de variables et peu de registres. Lorsque tous les registres sont occupés, il est nécessaire de vider en mémoire certains registres, sachant que le stockage d'une valeur en mémoire est beaucoup plus coûteux en termes de temps d'accès à la variable que le stockage d'une valeur dans un registre. L'algorithme d'allocation de registres a pour but de proposer une utilisation optimale des registres, c'est-à-dire minimisant l'occupation de la mémoire durant l'exécution d'un programme. L'objectif de ce projet est de proposer de résoudre le problème de l'allocation de registres de deux façons : par une méthode heuristique d'une part et par la programmation linéaire en nombres entiers d'autre part.

Description du projet

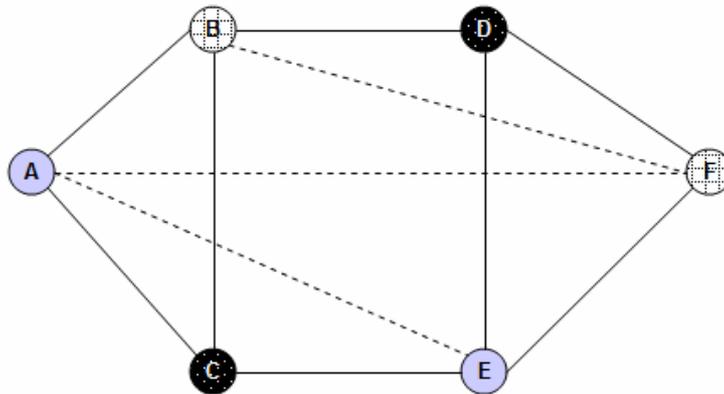
Les registres sont des zones mémoires intégrées au processeur, en nombre limité (moins d'une vingtaine sur les machines actuelles) et d'accès très rapide. Lors de la compilation d'un programme, il convient de décider quelles seront les variables dont la valeur sera stockée en registre (accès rapide) et celles dont la valeur sera stockée en mémoire (accès plus lent), étant donné que toutes les variables ne peuvent tenir en registres à cause de leur nombre limité et connu au départ : K . Une analyse préliminaire du programme, appelée analyse de vivacité, permet de déterminer quelles sont, à tout moment du programme (i.e. à chaque point du programme situé entre deux instructions), les variables dites « vivantes » (c'est-à-dire utilisées par le programme) et celles qui ne le sont pas. Deux variables vivantes au même instant sont dites interférentes. Deux variables qui interfèrent ne peuvent occuper simultanément le même registre. On modélise par un graphe, appelé **graphe d'interférences**, cette situation : les sommets représentent les variables du programme étudié, et une arête, appelée **arête d'interférence**, relie deux sommets/variables qui interfèrent et ne peuvent donc être placés dans le même registre. Puisque l'on dispose de K registres, il convient donc de trouver dans ce graphe une partition des sommets en K ensembles stables, s'il en existe une. Il s'agit donc de trouver une K -coloration, chaque couleur étant associée à un registre. S'il n'existe pas de K -coloration du graphe, des sommets ne sont pas colorés ; ces sommets correspondent alors aux variables qui seront stockées en zone mémoire.

Le graphe d'interférences possède en réalité deux types d'arêtes :

- Les arêtes d'interférence dont on vient de parler ;
- Des arêtes dites **arêtes de préférence**, représentées en pointillé dans la figure ci-dessous. Elles sont valuées par un poids, w_{ij} (poids de l'arête $[i, j]$). L'existence d'une arête de préférence entre deux sommets/variables i et j indique qu'il est souhaitable que ces variables soient stockées dans le même registre (à des moments différents de l'exécution du programme). Cette volonté de stocker deux variables dans le même registre provient de l'existence dans le programme d'une instruction « move » entre deux registres dans lesquels sont stockées ces variables, (et il n'existe pas d'arête d'interférence entre ces sommets), pour que l'une des deux variables prenne la valeur

de l'autre : si elles sont déjà dans le même registre, il n'est pas nécessaire d'exécuter l'instruction move et l'on gagne du temps d'exécution. Le poids de l'arête correspond à la fréquence de l'instruction move (si tous les poids valent 1, on ne pondère pas les arêtes).

Le problème d'allocation de registres auquel nous nous intéressons dans ce projet est simplifié. Il consiste, étant donné un nombre K de registres et **un graphe d'interférence K -colorable** (il existe une K -coloration, certaines variables ayant déjà été stockées en mémoire), à déterminer dans ce graphe une K -coloration qui maximise les préférences, c'est-à-dire lorsque le poids des arêtes de préférence sont tous égaux à 1, qui minimise le nombre d'arêtes de préférences dont les extrémités sont de couleurs différentes.



Instance résolue du problème d'allocation de registres dans laquelle on dispose de 3 registres : aucune 3-coloration ne permet de respecter les 3 préférences

Travail à réaliser

1. Modéliser la recherche d'une K -coloration avec préférences dans un graphe d'interférence K -colorable à l'aide d'un programme mathématique (linéaire ou quadratique, en variables entières ou 0-1).
2. Proposer une méthode heuristique de recherche d'une bonne solution admissible (pas nécessairement optimale). Vous testerez votre heuristique sur les instances proposées sur le site de l'UE.
3. A l'aide du solveur glpk, déterminer une borne (supérieure ou inférieure compte tenu de la modélisation proposée) pour les instances du problème proposées sur le site de l'UE, et, lorsque c'est possible, la solution optimale.

Echéancier

La phase 1 du travail à réaliser est à rendre pour le 10 avril 2009. Elle donnera lieu, si elle est correcte, à un point supplémentaire sur la note finale. Une modélisation sera ensuite proposée lors de la séance du 10/04.

Un rapport présentant le travail réalisé pour les phases 2 et 3 est à rendre le jour de la soutenance du projet. Les soutenances seront organisées la semaine du 10 juillet 2009.

Responsables : S. Elloumi (sourour.elloumi@cnam.fr) et Eric Soutif (eric.soutif@cnam.fr)