

Dans ce TD, nous allons nous servir des structures de GCC pour calculer la frontière de post-dominance et la frontière de post-dominance itérée afin d'afficher un warning sur les nœuds provoquant un problème potentiel. Il sera alors nécessaire d'étendre ces notions à un ensemble de nœuds et de préparer le graphe de flot de contrôle afin de numérotter les appels aux collectives MPI.

## Rappels

### 1 Environnement

Comme pour les séances précédentes, il est nécessaire de charger l'environnement du compilateur GCC 12.2.0. en utilisant la commande suivante :

```
$ source /home/patrick.carribault/setenv.sh
Loading Binutils 2.37 environment
Loading GCC 12.2.0 environment
Loading MPI environment
$
```

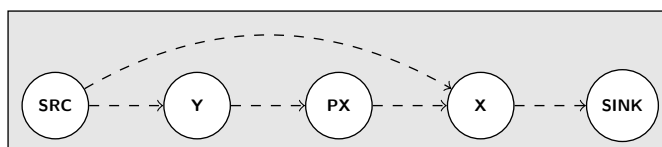
De plus, les données relatives à ce TD sont disponibles dans le répertoire suivant :

/home/patrick.carribault/LOCAL/TDs/TD5/

### 2 Frontière de dominance

La frontière de dominance d'un nœud  $y$ , notée  $DF(y)$ , est l'ensemble des nœuds  $x$  tels que  $y$  domine un prédécesseur de  $x$ , mais ne domine pas strictement  $x$ . Cela signifie que  $x$  est le premier nœud qui se trouve à la jonction du chemin allant de  $y$  vers le puit et du chemin allant de la source (SRC) vers le puit (SINK) qui ne passe pas par  $y$ .

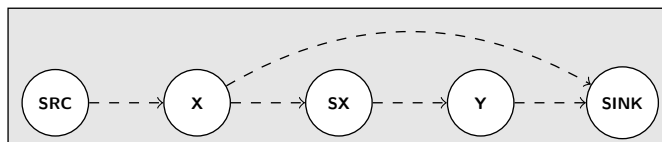
$$DF(y) = \{x \mid \exists Px \in \text{preds}(x), \text{ tel que } y \gg Px \text{ et } y \neq x\}$$



### 3 Frontière de post-dominance

La frontière de post-dominance d'un nœud  $y$ , notée  $PDF(y)$ , est l'ensemble des nœuds  $x$  tels que  $y$  post-domine un successeur de  $x$ , mais ne post-domine pas strictement  $x$ . Cela signifie que  $x$  est le premier nœud qui se trouve à la jonction du chemin allant de la source vers  $y$  et du chemin allant de la source (SRC) vers le puit (SINK) qui ne passe pas par  $y$ .

$$PDF(y) = \{x \mid \exists Sx \in \text{succs}(x), \text{ tel que } y \gg_p Sx \text{ et } y \not\gg_p y\}$$



## I Frontière de post-dominance d'un nœud

**Q.1:** Utiliser les bitmaps pour calculer et afficher la frontière de post-dominance de chaque basic block. Inspirez vous de l'algorithme fourni dans l'article "*A Simple, Fast Dominance Algorithm*". Pour l'utilisation des bitmaps, prendre exemple sur l'implémentation de la fonction `df_md_local_compute`, également présente dans les sources de GCC, pour comprendre la manipulation des bitmaps avec les fonctions définies dans `bitmap.h`.

## II Rang dans une séquence de collectives

La notion de post-dominance (et de sa frontière) est importante pour notre problème, mais il est nécessaire de l'étendre à un ensemble de nœuds. La première étape est donc de créer ces ensembles cibles. Il s'agit de groupement de collectives du même type (par exemple `MPI_Reduce`) qui pourront être appelées au même moment dans une séquence. Ainsi, seront dans le même ensemble, des collectives qui seront, par exemple, appelées en premier dans la fonction cible. Le rang peut se calculer grâce à l'équation *dataflow* suivante :

$$Rang(n) = \text{Max}_{x \in \text{predcs}(n)} Rang(x)$$

Pour permettre une implémentation à convergence de cette équation, il est nécessaire de simplifier le CFG en enlevant les arcs retours afin d'éviter les boucles qui feront tendre le rang vers l'infini.

**Q.2:** Calculer les arcs qui font partie du nouveau CFG nommé  $CFG'$  qui contient tous les arcs du CFG initial moins les arcs retours qui forment les boucles. Ce nouveau graphe  $CFG'$  peut être encodé dans un bitmap (un bit par arc) qui désigne si un arc du CFG initial fait partie de  $CFG'$ . L'algorithme peut se baser sur un parcours en largeur du CFG (en partant de la source) et en marquant les nœuds traversés. Un arc dont la destination n'est pas encore marquée fera partie de  $CFG'$ . Un arc parcouru dont la destination a déjà été marquée par l'algorithme sera exclu.

**Q.3:** En raisonnant sur  $CFG'$ , implémenter le calcul du rang pour tous les nœuds comportant un appel MPI concerné.

### III Post-dominance d'un ensemble

Les exercices précédents ont permis d'avoir à disposition la frontière de post-dominance d'un nœud ainsi que la notion de rang (dans une séquence) de collective. Il est maintenant possible de calculer la notion de post-dominance d'un ensemble de nœuds. Intuitivement, un ensemble de nœuds  $S$  post-dominent un nœud  $X$  si tous les chemins allant de  $X$  au puit passent par au moins un nœud appartenant à  $S$ .

#### 1 Post-dominance d'un ensemble

**Q.4:** À l'aide des bitmaps (comme pour les exercices précédents), construire un ensemble de nœuds regroupant tous ceux qui contiennent un appel à la bibliothèque MPI de même nature (c'est-à-dire une même collective de même rang). Calculer la post-dominance d'un ensemble  $E$  grâce à la formule suivante :

$$E \gg_p X \iff \forall S \in \text{succs}(X), \exists Y \in E, \text{ tel que } Y \gg_p S$$

Utiliser le fait que la post-dominance (relation  $\gg_p$ ) entre deux nœuds a déjà été calculée dans une question précédente.

#### 2 Frontière de post-dominance d'un ensemble

**Q.5:** Calculer la frontière de post-dominance des ensembles de nœuds (même collective, même rang) grâce à la formule suivante :

$$PDF(E) = \{X \mid \exists S \in \text{succs}(X), \text{ tel que } E \gg_p S \text{ et } E \not\gg_p X\}$$

Se baser sur les résultats de la question précédente et sur l'algorithme de calcul de la frontière d'un nœud.

#### 3 Premiers warnings

**Q.6:** Après avoir vérifié la PDF de l'ensemble des nœuds avec un appel MPI, afficher un message pour dire si il y a un problème ou non.

### IV Post-dominance itérée et warnings

#### 1 Post-dominance itérée

**Q.7:** A partir de cette dernière PDF, calculer la PDF itérée. Pour cela, il faut remonter l'arbre des BBs selon leur PDF jusqu'à arriver à un état stable.

**Q.8:** A partir de cette PDF itérée, afficher les basic blocks et les lignes de codes générant le problème potentiel.