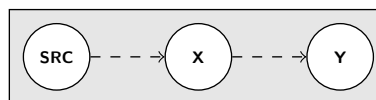


Rappels de définitions

1 Dominance

Un nœud X domine strictement un nœud Y si X apparaît sur tous les chemins de l'entrée SRC vers Y , avec $X \neq Y$. Nous écrivons alors $X \gg Y$. L'ensemble des nœuds qui dominent strictement Y est noté $Dom(Y)$.

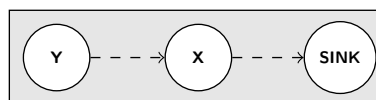
$$Dom(Y) = \left(\bigcap_{p \in \text{preds}(Y)} Dom(p) \right)$$



2 Post-Dominance

Un nœud X post-domine strictement un nœud Y si X apparaît sur tous les chemins de Y vers le puit $SINK$, avec $X \neq Y$. Nous écrivons alors $X \gg_p Y$. L'ensemble des nœuds qui post-dominent strictement Y est noté $PDom(Y)$.

$$PDom(Y) = \left(\bigcap_{p \in \text{succs}(Y)} PDom(p) \right)$$



I Dominance et Post-dominance dans un graphe

Le but de cette partie est d'apprendre à manipuler les notions de dominance et de post-dominance dans un graphe. Un graphe vous est fourni dans le répertoire *CODE* (voir fichier **graph_test.pdf**).

Les questions suivantes sont à réaliser sur ce graphe exemple.

Q.1: Pour chaque nœud, donner la liste des nœuds le dominant strictement.

Q.2: Même question pour les nœuds le post-dominant strictement.

Q.3: Pour chaque nœud v , donner l'ensemble des nœuds non strictement dominés par ce nœud v mais dont un prédécesseur est dominé par v (frontière de dominance)

Q.4: Même question pour la frontière de post-dominance.

II Dominance et Post-dominance dans GCC

Nous allons maintenant appliquer ces notions dans GCC.

Vous pouvez partir des plugins du TP précédent pour réaliser les questions suivantes.

Q.5: Le fichier **dominance.h** regroupe la plupart des fonctions pour la dominance. Trouver et utiliser les fonctions permettant d'initialiser les informations de **dominance**, et parcourir ensuite pour chaque nœud la liste des basic blocs qu'il domine. Comparer la sortie avec la question 1. Que remarquez-vous ?

Q.6: Modifier l'affichage pour n'afficher que les nœuds strictement dominés.

Q.7: Faire de même pour la postdomination