TP4

Utilisez le projet fourni dans tp4.tar.gz, comme pour les TPs précédents (importez-le sous Eclipse, configurez l'exécution de MyTest.java pour qu'il reçoive un fichier en entrée).

Ce TP comporte trois parties et s'appuie sur le CM4, à partir de la slide 29 sur l'analyse des expressions disponibles.
Il y a 2 fichiers à rendre au final.

Partie 1: analyse des expressions disponibles (fichier TP4AvailableExpressions.java)

Complétez les fonctions build(), onePass() et isFixedPoint() du fichier pour construire une analyse des expressions disponibles. Lancer MyTest pour voir les différentes étapes du calcul de point fixe de l'analyse obtenue.
Attention, lors du CM4, nous avons laissé un ? sur la définition de Kill et Gen (slide 33) pour cette analyse. Vous devez d'abord trouver l'algorithme puis l'implémenter. N'hésitez à faire valider votre algorithme par le chargé de TP en séance.
Pour cette partie, nous vous demandons de ne pas traiter les expressions de type ReadExpr. Ce sera l'objet de la partie 3.

Partie 2: optimisation par élimination des expressions communes (fichier TP4CSE.java)

En utilisant le résultat de l'analyse de la partie 1, implémentez l'optimisation présentée sur le slide 36 du CM4.
Afin d'optimiser l'usage d'une expression e en un point n,

```
n: x = e
```

par une affectation plus simple

```
n: x = tmp
```

il vous faudra vous assurer que e est bien *disponible* au point n, mais aussi déterminer les différents points de définitions qui ont stocké l'évaluation de e dans une variable. Pour ce faire, nous vous conseillons de faire un parcours (de graphe) en arrière à partir du point n pour rechercher les différentes définitions correspondantes.
Vous aurez besoin de générer des noms de variables fraiches et nous vous fournissons pour cela la classe FreshIdentGenerator avec un exemple d'utilisation.

Partie 3: expression symboliques pour les lectures mémoires (fichierTP4AvailableExpressions.java)

Reprenez la partie 1, en ajoutant la gestion des expressions de type lecture mémoire. Complétez pour cela la classe ReadExpr et modifiez l'analyse de flot de donnée. Cette modification devra permettre à votre transformation de la partie 2 de gérer des programmes avec des lectures redondantes.

Nous vous invitons à tester le résultat final de ce TP sur le programme examples/rtl/TP4Challenge.rtl