Compilation: TD numéro 3
Sélection d'instructions

Semestre 3, 2015-2016

Exercice 1

1. Proposer une traduction naïve de la condition `not (true and not (x + 4 <> z))`
2. En plus des règles pour `add` vues en cours, on considère le système de réécriture suivant :

$$\begin{aligned}
\texttt{and}(\texttt{li}_0, E) &\to \texttt{li}_0\\
\texttt{and}(\texttt{li}_1, E) &\to E\\
\texttt{and}(E, \texttt{li}_1) &\to E\\
\texttt{and}(E, \texttt{li}_0) &\to \texttt{li}_0\\
\texttt{not}(\texttt{sne}(E_1, E_2) &\to \texttt{seq}(E_1, E_2)\\
\texttt{not}(\texttt{not}(E)) &\to E
\end{aligned}$$

   Donner la ou les forme(s) normale(s) de la traduction naïve de la question précédente.
3. Le système de réécriture termine-t-il ?
4. Le système de réécriture est-il confluent ? Si non, comment le rendre confluent ?
5. Que penser de la quatrième règle ? Et de la première ?

Exercice 2

1. Donner une traduction naïve en Untyped Pseudo Pascal de l'expression `4 * (x + 2)`

Pour optimiser la multiplication, on propose la règle suivante :

$$\texttt{mul}(E, \texttt{li}_{2^k}) \to \texttt{sll}_k(E) \tag{1}$$

(On rappelle que `sll` est l'opération MIPS qui décale les bits contenu dans un registre vers la gauche.) Pour pouvoir traiter également le cas où la constante est à gauche du `mul`, on ajoute également la règle suivante :

$$\texttt{mul}(\texttt{li}_n, E) \to \texttt{mul}(E, \texttt{li}_n) \tag{2}$$

2. Donner une forme normale de la traduction obtenue à la question 1.
3. Le système de réécriture est-il terminant ? Justifier.
4. Calculer les paires critiques du système. Lesquelles sont-elles joignables ?
5. Le système de réécriture est-il confluent ? Justifier.