

Compilation

TD5 : Traduction dirigée par la syntaxe

C. ALIAS

Le but de ce TD est de mettre en application et de construire quelques règles de traduction directe du langage C vers du code machine DIGMIPS.

Exercice 1. *Récursion*

On considère le programme suivant:

```
int sum(int n)
{
  if(n == 1) return 1;
  else return n + sum(n-1);
}
```

Q1) Donner $\llbracket \text{sum} \rrbracket$. On laissera indéterminé le nombre de temporaires.

Exercice 2. *Boucles imbriquées*

On considère la portion de code suivante:

```
for(i=0; i<10; i++)
  for(j=0; j<10; j++)
    sum = sum + a[i][j];
```

Q1) Donner $\llbracket a[i][j] \rrbracket_\rho^t$. On laissera $\rho(a)$, $\rho(i)$ et $\rho(j)$ tels quels.

Q2) Comment améliorer les performances du code?

Exercice 3. *Evaluation paresseuse*

Q1) Proposez une règle de traduction pour le ET logique, $\llbracket e_1 \ \&\& \ e_2 \rrbracket_\rho^t$. On suivra un schéma d'évaluation paresseuse: il suffit qu'un terme e_i soit faux pour que l'ensemble soit faux (ce qui évite d'évaluer l'autre terme).

Q2) De même, proposez une règle de traduction paresseuse pour le OU logique, $\llbracket e_1 \ || \ e_2 \rrbracket_\rho^t$