

# Examen final de compilation

ÉNSIIE, semestre 3

jeudi 8 janvier 2015

Durée : 1h45.

Tout document personnel autorisé (pas de prêt entre voisins).

Ce sujet comporte cinq exercices indépendants, qui peuvent être traités dans l'ordre voulu.

Il contient 3 pages.

Le barème est donné à titre indicatif, il est susceptible d'être modifié. Le total est sur 20 points.

## Exercice 1 : Syntaxe (2,5 points)

Donner l'arbre de syntaxe abstraite des instructions Pseudo Pascal suivantes :

1. `x := t[j]`
2. `x[i] := z`
3. `t[j][i] := z`
4. `begin if c < 0 then a := 3 else b := 2; c := 3 end`
5. `if c < 0 then a := 3 else begin b := 2; c := 3 end`

## Exercice 2 : Analyse syntaxique (3 points)

On considère la grammaire suivante dont le symbole de départ est  $L$  :

$$L \rightarrow aC$$

$$C \rightarrow b$$

$$| Cc$$

1. Quel est le langage reconnu par cette grammaire ?
2. Construire l'automate déterministe LR(0) pour cette grammaire.
3. Cette grammaire est-elle LR(0) ? Justifier.
4. Proposer une grammaire LR(0) reconnaissant le même langage.

### Exercice 3 : Sélection d'instructions (4,5 points)

1. Donner une traduction naïve en Untyped Pseudo Pascal de l'expression  $1 * (x + y)$ .

Pour optimiser la multiplication, on propose les règles suivantes :

$$\text{mul}(E, \text{li}(1)) \rightarrow E \quad (1)$$

$$\text{mul}(E, \text{li}(0)) \rightarrow \text{sub}(E, E) \quad (2)$$

Pour pouvoir traiter le cas où la constante est à gauche du `mul`, on ajoute également la règle suivante :

$$\text{mul}(\text{li}(n), E) \rightarrow \text{mul}(E, \text{li}(n)) \quad (3)$$

2. Donner une forme normale de la traduction obtenue à la question 1.
3. Le système de réécriture est-il fortement normalisant ? (Autrement dit, existe-t-il ou non une suite infinie d'étape de réécriture ?) Justifier.
4. Calculer les paires critiques du système. Lesquelles sont-elles joignables ?
5. Le système de réécriture est-il confluent ? Justifier.
6. À quelle condition peut-on appliquer la règle (2) ?

### Exercice 4 : Allocation de registres (6 points)

Soit le programme Pseudo Pascal suivant :

```
begin
  while a < 0 and c > 0 do
    begin
      a := d - c;
      c := c - 1
    end;
  b := a;
  e := a * a;
  d := b;
  a := e * d
end
```

1. Donner le graphe de flot de contrôle correspondant. Ne pas oublier le comportement coupe-circuit du `and`.
2. En déduire sa traduction en langage RTL. On supposera que les variables `a`, `b`, `c`, `d` et `e` sont stockées respectivement dans les pseudo-registres `%0`, `%1`, `%2`, `%3` et `%4`.
3. Indiquer dans un tableau quelles variables (pseudo-registres) sont vivantes en chacun des points du programme.
4. Donner le graphe d'interférence du programme (avec les arêtes de préférence). On indiquera sur les arêtes le numéro d'une instruction qui a causé l'interférence ou la préférence.

5. Essayer de 2-colorier ce graphe en appliquant l'algorithme de George et Appel. On détaillera bien chaque étape. On utilisera le critère de George pour justifier les fusions éventuelles.
6. Donner le code RTL après allocation des registres. On utilisera les registres `$s0` et `$s1` comme registres dans lesquels on a alloué les variables, et `$t0` et `$t1` pour sauvegarder et restaurer les variables éventuellement spillées. On éliminera les instructions `move` inutiles.

## Exercice 5 : Convention d'appel (4 points)

On considère le programme Pseudo Pascal suivant :

```

1 program
2 var x : integer;
3
4 function f(x, y : integer) : integer;
5   var z : integer;
6   begin
7     z := y * x;
8     f := 6 - z
9   end;
10
11 function g(u : integer, v : integer) : integer;
12   begin
13     x := u + v;
14     x := f(x, u * v);
15     g := u - x
16   end;
17
18 begin
19   x := 2;
20   x := g(x, x)
21 end.
```

Dans cet exercice, on suppose qu'on utilise la convention d'appel MIPS, et on suppose que la variable locale `z` de `f` est stockée dans le registre callee-saved `$s0`.

1. Quel(s) registre(s) les fonctions `f` et `g` doivent-elles respectivement sauvegarder ?
2. En déduire la trame des fonctions `f` et `g`.
3. Indiquer le code à rajouter lors de l'explicitation des appels :
  - avant la ligne 7;
  - après la ligne 8;
  - avant la ligne 13;
  - avant et après l'appel à `f` ligne 14;
  - après la ligne 15.