Examen du cours de compilation
Tout document autorisé (durée: 1h30)
Ce sujet comprend 3 exercices présentés sur 2 pages.

Sandrine Blazy - ENSIIE

11 janvier 2008

Exercice 1 (sur 11 points) [1]

Soit la fonction de Fibonacci suivante :

```
int fib (int n) {
  int prec = 1;
  int cur = 1;
  int i;

  for (i=2; i <=n; i++) {
     int sav = cur;
     cur = prec + cur;
     prec = sav; }
return cur; }
```

1. Donner la sémantique de la boucle `for` de la fonction `fib` lorsque `n=1`.
2. Donner le graphe de contrôle de la fonction `fib`, sans bloc de base.
3. Donner le graphe de contrôle de la fonction `fib`, avec blocs de base.
4. Traduire la boucle `for` de la fonction `fib` en RTL.
5. Effectuer une analyse de vivacité de la fonction `fib`, sachant que seul le paramètre `n` est vivant à l'entrée de la fonction, et aucune variable ou paramètre n'est vivant à la sortie de la fonction.
6. Donner le graphe d'interférences de la fonction `fib`.
7. Effectuer un coloriage optimiste de ce graphe d'interférences, sachant que le nombre de registres disponibles est 3.
8. Le coloriage aurait-il été différent si un coloriage pessimiste avait été choisi ?
9. Que devrait-il se passer lors de l'étape de fusion ?
10. Quelles instructions doivent être rajoutées dans la fonction `fib` à l'issue de l'allocation de registres ?

[1] Le barême indiqué est indicatif, et pourra être modifié.

Exercice 2 (sur 2 points) [2]

1. Comment améliorer le code suivant ?

```
mv $v0, $a0
li $a2, 1
sub $v0, $v0, $a2
li $a1, 3
mul $v0, $v0, $a1
mv $a2, $a0
mv $a1, $a0
mul $a1, $a1, $a2
add $v0, $v0, $a1
```

2. Comment s'appelle l'optimisation transformant ce code en un code amélioré ?

Exercice 3 (sur 7 points)

Soit la fonction `fib` suivante, en langage LTL. Cette fonction est une traduction de la fonction C donnée dans l'exercice 1.

```
proc fib (1)
var 8
entry f1
f1:  newframe           -> f2          f12: move $t1, $v0      -> f13
f2:  sets local(0), $ra -> f3          f13: add $v0, $v0, $v1  -> f14
f3:  j                  -> f4          f14: move $v1, $t1      -> f15
f4:  sets local(4), $s0 -> f5          f15: add $t0, $t0, 1    -> f16
f5:  move $s0, $a0      -> f7          f16: j                  -> f10
f7:  li $v1, 1          -> f8          f17: gets $ra, local(0) -> f18
f8:  li $v0, 1          -> f9          f18: gets $s0, local(4) -> f19
f9:  li $t0, 2          -> f10         f19: delframe           -> f11
f10: bgt $t0, $a0       -> f11         f11: jr $ra
```

1. Dessiner les trames de pile allouées lors de l'exécution du programme qui se contente d'intialiser une variable globale `v` à la valeur 6 puis d'exécuter l'instruction `v = fib (v)`.
2. Que signifie l'instruction d'étiquette `f2` ? Quel est son rôle ?
3. Donner le code linéarisé (*i.e.* en langage LIN) correspondant et justifier les transformations effectuées.
4. Comment linéariser efficacement la boucle de la fonction `fib` ?
5. Donner le code assembleur correspondant au code linéarisé de la question 3 et justifier les transformations effectuées.

[2] Le barême indiqué est indicatif, et pourra être modifié.