

Examen de compilation

M1 Informatique Fondamentale, ENS Lyon
Mercredi 4 janvier 2012

Durée: 3 heures. Tous les documents sont autorisés.

Cet examen est constitué de 4 exercices indépendants.

Exercice 1. Analyse lexicale

Sur l'alphabet $\Sigma = \{a, b\}$, on considère le langage défini par l'expression régulière $(a|b)^*abb$.

Q1) Donner la construction de Thompson correspondante, *sans détailler*.

Q2) Déterminer l'automate obtenu en Q1), *sans détailler*.

Q3) Minimiser l'automate obtenu en Q2), *en détaillant*.

Exercice 2. Analyse syntaxique

Sur l'alphabet terminal $\Sigma = \{y, +\}$, on considère la grammaire suivante, dont l'axiome est S :

$$\begin{aligned} S &\rightarrow A \\ A &\rightarrow A + A \mid B + + \\ B &\rightarrow y \end{aligned}$$

Q4) Dessiner l'arbre de dérivation pour l'entrée " $y + + + y + +$ ".

Q5) Dérouler l'exécution d'un analyseur LR(1) sur cette entrée. Le résultat sera présenté avec un tableau à deux colonnes (pile, action), *sans donner l'automate LR(1)*.

Q6) Donner les ensembles PREMIER des non-terminaux de la grammaire, *sans détailler*.

Q7) Donner la clôture LR(1) de l'item $[S \rightarrow \bullet A, \$]$

Q8) L'automate LR(1) comporte un état avec les items $\{[A \rightarrow A + A \bullet, \$/+], [A \rightarrow A \bullet + A, \$/+]\}$. De cet état, part une transition étiquetée par $+$. Donner l'état destination.

Bonus: Construire l'automate LR(1) complet (8 états).

Exercice 3. Allocation de registres

On considère une machine avec $K = 3$ registres physiques r1, r2 et r3. On cherche à allouer des registres pour le code intermédiaire suivant:

```
a := r2
b := r3
c := r1
r1 := 5
d := r1
L1: r1 := d
d := d + r1
r1 := d
if d < 10 goto L1
r3 := b
r2 := a
return
LIVE-OUT: r1, r2, r3
```

Q9) Donner les ensembles LIVE-IN et LIVE-OUT pour chacune des instructions, *sans détailler*.

Q10) Tracer le graphe d'interférence. On n'oubliera pas de mettre des noeuds pour r1, r2 et r3. Ces noeuds seront préférablement colorés par r1, r2 et r3.

Q11) *Détailler* les étapes de l'algorithme Iterated Register Coalescing. On utilisera le critère de Briggs pour le coalesce conservatif.

Indication: Le premier noeud à être spillé sera a.

Q12) Donner le code final. On indiquera le code de spill et les copies éliminées.

Exercice 4. Appels de fonctions

On considère le programme suivant:

```
program test;

  procedure b (function h (n : integer) : integer) ;
    var m : integer ;
    begin m := 3 ; writeln(h (2)) end ;

  procedure c ;
    var m : integer ;

    function f (n : integer) : integer;
      begin f := m + n end

    procedure r ;
      var m : integer ;
      begin m := 7 ; b (f) end

    begin m := 0; r end

begin
  c
end.
```

Q13) Dessiner l'arbre d'activation de ce programme.

Q14) Dessiner l'état de la pile d'appel à l'exécution de **f**, et l'état du **display**.

Q15) Donner $\rho(m)$ à la compilation de **f**. Combien vaut **m** à l'exécution de **f**? Pourquoi est-ce dangereux?

Q16) On préfère garder la valeur de **m** de l'environnement textuel ($m = 7$). Expliquer en quelques phrases comment il faudrait procéder.