Examen Analyse Statique
Durée: 3 heures

Exercice 1.

Soit f une application continue d'un treillis complet dans lui-même, et soit $k = inf(\{f^n(\top) \mid n \geq 0\})$.

1. Montrez que $f(k) \leq k$.
2. Montrez que pour tout point fixe y de f, $y \leq k$.

Exercice 2. On considère l'analyse des définitions possibles.

1. Rappelez la définition de cette analyse.
2. On considère le programme suivant :

   $[x := 0]^1; [x := 3]^2;($ if $[z = x]^3$ then $[z := 0]^4$ else $[z := x]^5); [y := x]^6; [x := y{+}z]^7$

   Déterminez les définitions possibles à l'entrée et à la sortie de chaque point.
3. Rappelez la définition des fonctions kill et gen utilisées pour faire l'analyse des définitions possibles dans le cas du langage WHILE.
4. Donnez le système d'équations que doivent satisfaire les fonctions DP_{entree} et DP_{sortie} qui définissent l'analyse des définitions possibles en chaque point du programme.
5. Ecrivez et résolvez le système d'équations qui permettent de réaliser l'analyse des définitions possibles du programme précédent. Qu'en déduisez-vous ?
6. Déterminez les chaînes UD et DU des différents points du programme précédent.
7. A quoi servent ces chaînes ?

Exercice 3.

1. Rappelez la définition d'un automate à pile
2. Montrez comment modéliser un programme écrit dans un langage où on autorise les appels de procédures par un automate à pile.
3. Comment peut-on représenter les ensembles de configurations d'un automate à pile ?

4. Soit $\mathcal{P}$ un automate à pile et $\mathcal{C}$ un ensemble régulier de configurations. Montrez que $pre^*_{\mathcal{P}}(\mathcal{C})$ est régulier en donnant un algorithme qui le calcule. Cet algorithme termine-t-il ? Pourquoi ?
5. Soit $\mathcal{P}$ un automate à pile et $\mathcal{C}$ un ensemble régulier de configurations. Montrez que $post^*_{\mathcal{P}}(\mathcal{C})$ est régulier en donnant un algorithme qui le calcule. Cet algorithme termine-t-il ? Pourquoi ?
6. Rappelez comment peut-on déterminer si une variable est vivante au point n dans le cas où notre langage permet d'avoir des appels de procédures.
7. Montrez comment peut-on déterminer si une expression est very busy au point n dans le cas où notre langage permet d'avoir des appels de procédures.
8. Montrez comment peut-on déterminer si une expression est valable au point n dans le cas où notre langage permet d'avoir des appels de procédures.
9. Montrez comment peut-on déterminer si une définition est possible (accessible) au point n dans le cas où notre langage permet d'avoir des appels de procédures.

Exercice 4.

Soit $\mathcal{P}$ l'automate à pile ayant les règles suivantes :
$(p_0, a) \rightarrow (p_0, \epsilon)$
$(p_0, a) \rightarrow (p_1, a)$
$(p_1, a) \rightarrow (p_0, ab)$

Soit C l'ensemble de configurations $(p_0, a(bb)^*b)$.

1. Calculer $post^*(C)$.
2. Calculer $pre^*(C)$.

Exercice 5.

On considère le programme suivant où i est entre 1 et 3.

```
decl g;
void main()
begin
level1();
level1();
if (!g) then
ERREUR
else
skip;
fi
end
```

```
void leveli()
begin
decl a,b,c;
if(g) then
a,b,c := 0,0,0;
while(!a|!b|!c) do
if (!a) then
a := 1;
elsif (!b) then
a,b := 0,1;
elsif (!c) then
a,b,c := 0,0,1;
fi
od
else
leveli+1(); leveli+1();
fi
g := !g;
end
```

1. Déterminez un modèle pour ce programme.
2. Analyser ce programme et déterminer si ERREUR est accessible.