Partiel Analyse Statique
Durée: 3h

Exercice 1.

1. Qu'est ce qu'une fonction continue ?
2. Qu'est ce qu'une fonction monotone ?
3. Montrez que toute fonction continue est monotone.
4. Qu'est ce qu'un minorant ? un majorant ? une borne inférieure ? une borne supérieure ?
5. Soit $E = \{a, b, c, d, e, f, g, h, i\}$ ordonnés comme suit : $a \sqsubseteq e, b \sqsubseteq e, c \sqsubseteq f, d \sqsubseteq f, g \sqsubseteq a, g \sqsubseteq c, h \sqsubseteq b, h \sqsubseteq d, i \sqsubseteq g, i \sqsubseteq h$. Déterminez les majorants, minorants, bornes inférieures, et bornes supérieures des ensembles suivants : $\{a, b\}$; $\{c, d\}$; $\{a, b, c, d\}$; $\{a, b, g, h\}$; $\{c, d, g, h\}$; $\{a, b, c, d, g, h\}$.
6. Qu'est ce qu'un treillis complet ?
7. Soit $E = \{A, B, C, D, E\}$ un ensemble. Quel est le treillis naturel qu'on peut définir à partir des éléments de E ?
8. Soit $(E, \leq)$ un treillis complet. Montrez que si $E_1 \subseteq E_2 \subseteq E$, alors $inf(E_2) \leq inf(E_1) \leq sup(E_1) \leq sup(E_2)$, où inf dénote la borne inférieure, et sup, la borne supérieure.
9. Montrez que si E est un ensemble ordonné fini admettant un plus petit élément $\perp$, alors pour toute fonction monotone f de E dans lui-même il existe $k \leq card(E)$ tel que le plus petit point fixe de f est $f^k(\perp)$.

Exercice 2.

On considère le programme suivant :

```
r:=1; z:=1;
y:= x+8; z:= r+y-9;
while (x>0) do
  { y:=r+5 ;  z:=x+8 ;
    r:=0 ;
    while (z>0) do
       { r:= r+y-9;
         z:=z-1;
       }
    y:=0 ;  x:=x-1 ;
  }
```

1. Dessinez le graphe de flot de contrôle de ce programme.
2. Déterminez les variables vivantes en chaque point.
3. Formalisez cette analyse, en détaillant chaque étape :
   (a) Donnez les fonctions *kill* et *gen* pour cette analyse
   (b) Donnez le système d'équations qui définit cette analyse
   (c) Rappelez les théorèmes de Tarski et de Kleene
   (d) Montrez comment ces théorèmes peuvent être utilisés pour résoudre le système d'équations. En particulier, montrez que ce système d'équations définit une fonction continue et monotone.
   (e) Appliquez ceci pour l'analyse de cet exemple. VALABLES
4. Déterminez les expressions disponibles en chaque point.
5. Formalisez cette analyse, en détaillant chaque étape :
   (a) Donnez les fonctions *kill* et *gen* pour cette analyse
   (b) Donnez le système d'équations qui définit cette analyse
   (c) Montrez que ce système d'équations définit une fonction continue et monotone, et expliquez comment on peut résoudre ce système d'équations.
   (d) Appliquez ceci pour l'analyse de cet exemple.
6. Déterminez les invariants de boucles.
7. Utilisez les résultats précédents pour optimiser le programme.

Exercice 3.

On considère le programme suivant :

$$[x := 1]^1; [x := x - 1]^2; [x := 2]^3$$

Il est clair que x est mort aux sorties des points 2 et 3. Mais, x est vivant à la sortie de 1 même si sa seule utilisation est de calculer une nouvelle valeur pour une variable qui est morte. On dira qu'une variable est *faible* si elle est morte ou si elle n'est utilisée que pour calculer de nouvelles valeurs pour les variables faibles. Dans l'exemple, x est faible aux sorties des points 1, 2 et 3. Définissez une analyse qui détecte les variables faibles.

Exercice 4.

1. Proposez un modèle du programme ci-dessous sans tenir compte des valeurs des variables.
2. Est-ce qu'un point ERREUR est accessible dans ce modèle ?
3. Proposez un autre modèle qui tient compte des variables.
4. Est-ce qu'un point ERREUR est accessible dans ce nouveau modèle ?

5. En quoi consiste la technique CEGAR ?
6. Appliquez ce principe pour déterminer si les points ERREUR sont accessibles dans le programme.

```
bool l; /* global variable */

void lock(){
if (l) ERROR;
l := 1;
/* acquire a lock */
}

void unlock() {
if (!l) ERROR;
/* release the lock */
l := 0;
}


bool g (bool x) {
return !x;
}

void main() {
bool a,b;
l,a := 0,0;
lock();
b := g(a);
unlock();
}
```