Corrigé de l'Examen Semestriel – Théorie de la Programmation (TPGO) – 2CS ESI 2013/2014

A) Complexité

1- Calculez la complexité de l'algorithme suivant :

i ← 1 ; j ← n ;
TQ (i < j)
i ← i+1 ; SI (i = j) j ← j div 2 ; i ← 1 FSI
FTQ

A chaque itération, la variable i est incrémentée, et à chaque fois qu'elle atteint j, elle est réinitialisée à 1 alors que j est divisée par 2, donc le nombre d'itérations total est : n + n/2 + n/4 + … + 1
Donc l'algorithme est en **O(n)**.

2- Que peut-on dire de l'intersection entre les classes P et NP-Complet ?
Il y a 2 possibilités, soit l'intersection est vide et donc P ≠ NP, soit l'intersection n'est pas vide (elle est dans ce cas égale à NP-Complet) et on aura donc P = NP.
Il n'y a donc **pas de réponse définitive** à cette question pour le moment.

3- Peut-on définir la classe NP sans faire référence au non-déterminisme ?
Oui, puisque pour montrer qu'un problème de décision est dans NP, il suffit d'exhiber un système de preuve polynomial.

B) Résolution de Problèmes

1- Qu'est-ce que le principe d'optimalité en programmation dynamique ?
Pour un problème d'optimisation, cela consiste à vérifier que toute sous-séquence de choix (ou de décision) d'une séquence optimale, reste aussi optimale.

2- Quelles sont les différences et les similitudes entre les méthodes de recherche 'Best First Search' et 'Branch & Bound' ?
Les deux méthodes sont basées sur la procédure du parcours en largeur.
Pour les différences, Best-First-Search ne trouve pas forcément la solution optimale et la file utilisée dans le parcours en largeur manipule des sommets, contrairement à Branch-and-Bound qui trouve la solution optimale et la file du parcours en largeur manipule des chemins.

3- Lors de l'évaluation MinMax d'une configuration J dans un jeu donné, obtient-on les mêmes valeurs avec ou sans élagage (coupe) α/β ?
Oui, la même valeur est retournée lors de l'évaluation d'une configuration, avec ou sans élagage α/β, car le principe de l'élagage repose justement sur l'abandon des sous-arbres qui ne peuvent pas influer sur la valeur à retourner.

C) Raisonnement formel sur les Programmes

1- Est-ce que les énoncés suivants sont des théorèmes dans le système formel de Hoare :

– *(z=1) { y ← 2 ; y ← z*2 - x + y ; x ← x+y } (x=4)*

Oui c'est un théorème, car :
L'énoncé est vrai par SEQ si 1) (z=1) { y ← 2 ; y ← z*2 - x + y } A et 2) A{ x ← x+y } (x=4) sont vrais.
L'énoncé 2) est vrai (axiome AFF) en choisissant A = (x+y = 4)
L'énoncé 1) est vrai par SEQ si 3) (z=1) { y ← 2 } B et 4) B { y ← z*2 - x + y } A sont vrais.
L'énoncé 4) est vrai (axiome AFF) en choisissant B = (2z + y = 4)
L'énoncé 3) est vrai par IMP1 car l'énoncé (2z + 2 = 4) { y ← 2 } (2z + y = 4) est un axiome (AFF) et l'implication (z = 1) ⇒ (2z + 2 = 4) est vraie.

– *$(y > 0)$ { $y \leftarrow y-1$; $z \leftarrow x+y$ } $(x < z)$*

Non ce n'est pas un théorème, car :
Pour que l'énoncé soit vrai par SEQ, il aurait fallu que :
L'énoncé 1) $(y > 0)$ { $y \leftarrow y-1$ } A et l'énoncé 2) A { $z \leftarrow x+y$ } $(x < z)$ soient vrais.
L'énoncé 2) est vrai car c'est un axiome AFF en choisissant A = $(x < x+y)$, c-a-d A = $(y>0)$
Pour que l'énoncé 1) soit vrai par IMP1, il aurait fallu que :
Les énoncés 3) $(y>0) \Rightarrow (y-1 > 0)$ et 4) $(y-1 > 0)$ { $y \leftarrow y-1$ } $(y>0)$ soient vrais aussi.
L'énoncé 4 est vrai car c'est un axiome AFF, par contre l'implication 3) n'est pas vraie (contre exemple y=1)
Donc l'énoncé 1) ne peut pas être vrai.

– *$(x=3)$ { SI ($x > y$) $z \leftarrow 2$ SINON $x \leftarrow 1$ FSI ; SI ($y = x$) $z \leftarrow z-1$ FSI ; $x \leftarrow y+1$ } $(x=z)$*

Non ce n'est pas un théorème, car :
Pour que l'énoncé soit vrai par SEQ, il aurait fallu que :
L'énoncé 1) $(x=3)$ { SI ($x > y$) $z \leftarrow 2$ SINON $x \leftarrow 1$ FSI ; SI ($y = x$) $z \leftarrow z-1$ FSI } A
et l'énoncé 2) A { $x \leftarrow y+1$ } $(x=z)$ soient vrais tous les deux.
L'énoncé 2) est vrai car c'est un axiome AFF en choisissant A = $(y+1=z)$
Pour que l'énoncé 1) soit vrai par SEQ, il aurait fallu que :
Les énoncés 3) $(x=3)$ { SI ($x > y$) $z \leftarrow 2$ SINON $x \leftarrow 1$ FSI } B
et 4) B { SI ($y = x$) $z \leftarrow z-1$ FSI } $(y+1=z)$ soient vrais aussi.
L'énoncé 4) est vrai par CND1 car on peut trouver un prédicat B, vérifiant :
l'énoncé 5) B & $y \neq x \Rightarrow (y+1=z)$ et l'énoncé 6) B & $y=x$ { $z \leftarrow z-1$ } $(y+1=z)$
Soit B = ($y=x$ & $y=z-2$) ou ($y \neq x$ & $y+1=z$) la plus faible précondition vérifiant 5) et 6) (par IMP1 et AFF)
Pour que l'énoncé 3) soit vrai, il aurait fallu, d'après la règle CND2, qu'on ait :
Les énoncés 7) ($x=3$ & $x > y$) { $z \leftarrow 2$ } B et 8) ($x=3$ & $x \leq y$) { $x \leftarrow 1$ } B, vrais tous les deux.
Pour que 7) soit vrai par AFF et IMP1, il aurait fallu que :
($x=3$ & $x > y$) $\Rightarrow$ [($y=x$ & $y=0$) ou ($y \neq x$ & $y=1$)] (énoncé 7a)
Pour que 8) soit vrai par AFF et IMP1, il aurait fallu que :
($x=3$ & $x \leq y$) $\Rightarrow$ [($y=1$ & $y=z-2$) ou ($y \neq 1$ & $y+1=z$)] (énoncé 8a)
Cette dernière implication (8a) ne peut pas être vraie (contre exemple : x=3, y=4, z=1)
Donc l'énoncé 1) ne peut pas être vrai.

2- Soit P le programme suivant :

```
SI ( a > b )   a ← a-b ;   b ← b+a ;   a ← b-a   FSI ;
i ← 1 ;
TQ ( (i*b mod a) ≠ 0   ET   i < a )
       i ← i+1
FTQ
```

– *Donnez l'arbre de preuve pour la démonstration de l'énoncé : (a>0 & b>0) { P } (ppcm(a,b) = i*b), en mettant en évidence toutes les implications qui restent à vérifier.*
(ppcm(a,b) représente le plus petit multiple commun aux entiers a et b).

Soient le découpage suivant : P=[P1 ; P2] , P1=[P3 ; i ← 1] , P3=[SI (a>b) P4 FSI],
P4=[P5 ; a ← b-a], P5=[a ← a-b ; b ← b+a], p2=[TQ ((i*b mod a) ≠ 0 ET i < a) i← i+1 FTQ]
Soient E le prédicat d'entrée : (a>0 & b>0), et S le prédicat de sortie : (ppcm(a,b) = i*b).
E { P } S est vrai par SEQ si les énoncés (1) et (2) suivants sont vrais :

```
|_(1) E {P1} F  ___et ___ (2) F{P2}S
   | (SEQ)                 |(IMP2)
   |                       |___(3) F {P2} ( F & ((i*b mod a)=0 ou i ≥ a) )
   |                       |     |(ITE)
   |                       |     |______ (5) (F & (i*b mod a) ≠ 0 & i<a) { i← i+1 } F
   |                       |               |(IMP1)
   |                       |               |__ (6) (F & (i*b mod a) ≠ 0 & i<a) ⇒ G
   |                       |               |__ (7) G { i← i+1 } F
   |                       |
   |                       |___(4) ( F & ((i*b mod a)=0 ou i ≥ a) ) ⇒ S    (F étant l'invariant de boucle)
```

```
…
 (1)
   |__ (8) E {P3} H
   |       |(CND1)
   |       |__ (10) (E & a>b) { P4 } H
   |       |       |(SEQ)
   |       |       |__ (12) (E & a>b) { P5 } I
   |       |       |       |(SEQ)
   |       |       |       |__ (14)  (E & a>b) { a ← a-b } J
   |       |       |       |       |(IMP1)
   |       |       |       |       |__ (16) (E & a>b) ⇒ K
   |       |       |       |       |__ (17) K { a ← a-b } J
   |       |       |       |
   |       |       |       |__ (15)  J { b ← b+a } I
   |       |       |
   |       |       |__ (13) I { a ← b-a } H
   |       |
   |       |__ (11) (E & a ≤ b)  ⇒ H
   |
   |__ (9) H { i ← 1} F
```

Les énoncés feuilles : 7, 9, 13, 15 et 17 sont des axiomes de l'affectation (AFF), donc considérés vrais.
Le reste des énoncés feuilles : 4, 6, 11 et 16 sont des implications à démontrer.

– *Trouvez le bon invariant de boucle et montrez toutes les implications de l'arbre précédent pour compléter la preuve.*

Pour trouver le **bon invariant F**, on va se focaliser principalement sur l'implication (4) :

$(F \ \& \ ((i*b \bmod a)=0 \text{ ou } i \geq a)\ ) \Rightarrow (ppcm(a,b) = i*b)$

On peut choisir pour F, le prédicat : **$i*b \leq ppcm(a,b) \leq a*b$**, car en analysant la boucle TQ, on remarque que le PPCM est obtenu en cherchant parmi les multiples de b (c-a-d : b, 2b, 3b, …, ib …) et dans cet ordre, le premier qui est aussi un multiple de a. Donc à chaque itération i, on est sûr que $i*b \leq ppcm(a,b) \leq a*b$.

<u>Montrons que l'implication (4) est vraie :</u>

(4) $(\ i*b \leq ppcm(a,b) \leq a*b \ \& \ ((i*b \bmod a)=0 \text{ ou } i \geq a)\ ) \Rightarrow (ppcm(a,b) = i*b)$

(i) si (i*b mod a)=0, donc ib est un multiple de a et comme on a dans l'antécédent de l'implication que le plus petit multiple commun à a et à b est compris entre ib et ab, on en déduit donc que le ppcm est forcément = ib.

(ii) si par contre on a que $i \geq a$, et comme dans l'antécédent de l'implication on a dans F que le ppcm est compris entre ib et ab (c-a-d ib <= ab donc i <= a), on en déduit forcément que i = a et donc ppcm = ab = ib

<u>Montrons que l'implication (6) est vraie :</u>

(6) $(\ i*b \leq ppcm(a,b) \leq a*b \ \& \ (i*b \bmod a) \neq 0 \ \& \ i<a) \Rightarrow G$

G étant la plus faible précondition pour que l'énoncé (7) soit vrai. C-a-d G : $(i+1)*b \leq ppcm(a,b) \leq a*b$.

Si le PPCM est compris entre ib et ab et si ib n'est pas un multiple de a (c-a-d : (ib mod a) ≠ 0), donc forcément le PPCM de a et b doit être compris entre (i+1)b et ab (sachant que i < a).

<u>Montrons que l'implication (11) est vraie :</u>

(11) $(a>0 \ \& \ b>0 \ \& \ a \leq b) \Rightarrow H$

H étant la plus faible précondition pour que l'énoncé (9) soit vrai. C-a-d, H : $b \leq ppcm(a,b) \leq ab$.

Lorsque $a \leq b$, le plus petit multiple commun entre a et b est forcément compris entre b et ab.

Montrons que l'implication (16) est vraie :
(16) (a>0 & b>0 & a>b) ⇒ K
K étant la plus faible précondition pour que l'énoncé (17) soit vrai, H = [a-b/a]J
J étant la plus faible précondition pour que l'énoncé (15) soit vrai, J = [b+a/b]I
I étant la plus faible précondition pour que l'énoncé (13) soit vrai, I = [b-a/a]H
Ce qui donne : (pour H : b ≤ ppcm(a,b) ≤ ab)
I : b ≤ ppcm(b-a,b) ≤ (b-a)b
J : (b+a) ≤ ppcm(b+a-a,b+a) ≤ (b+a-a)(b+a)
K : a ≤ ppcm(b,a) ≤ ba
L'implication (16) est donc vraie puisque on a que a>b dans l'antécédent, donc le PPCM est forcément compris entre a et ab.

Toutes les feuilles de l'arbre de preuve étant vraies, la racine (l'énoncé initial : E{P} S) est donc un théorème.

3- Démontrez que : (n > 0) { Q } (r = n/(n+1)) est vrai, avec Q le corps de la procédure 'Somme' suivante :

```
Somme( n:entier ; var r:réel )
  SI ( n = 1 )        r ← 1/2
  SINON      Somme( n-1, r ) ;   r ← r + 1/(n*n + n)
  FSI
```

Posons E : (n > 0) et S : (r = n/(n+1))
Il s'agit donc de démontrer l'énoncé (1) E {Q} S.
Par la règle CND2, (1) sera vrai si :
(2) (E & n=1){ r ← 1/2 }S et (3) (E & n ≠ 1){ Somme(n-1, r) ; r ← r + 1/(n*n + n) } S
L'énoncé (2) est vrai par IMP1 car :
(1/2 =n/(n+1)){r ← 1/2}(r=n/(n+1)) est un axiome AFF et (n>0 & n=1) ⇒ (1/2 = n/(n+1)) est vraie.
L'énoncé (3) sera vrai par SEQ si :
(4) (E & n ≠ 1){ Somme(n-1, r)} F et (5) F {r ← r + 1/(n*n + n) } S
L'énoncé (5) est vrai car c'est un axiome AFF à condition de choisir pour F le prédicat suivant :
(r + 1/(n*n + n) = n/(n+1))
L'énoncé (4) est vrai par IMP1 car :
(n>0 & n ≠ 1) ⇒ (n-1 > 0) est vraie et,
l'énoncé : (n-1 > 0) { Somme(n-1, r)} F est aussi vrai,
car on remarque que F :(r + 1/(n*n + n) = n/(n+1)) ≡ (r = (n-1)/n), or l'énoncé :
(n-1 > 0) { Somme(n-1, r)} (r = (n-1)/n) est vrai par hypothèse que les appels récursifs internes sont vrais.