

TP



Le dossier à remettre est une archive compressée (zip) contenant trois dossiers ex1, ex2 et ex3. Chaque dossier répond aux questions en fournissant les éléments suivants:

- ex.txt: explications sur ce qui a été fait et sur les éventuels problèmes rencontrés pour mettre au point la solution. En particulier, vous pourrez indiquer les constantes utilisées.
- ex.toolbox
- ex.tla
- ex.cfg

Le dossier portera le nom suivant `tp-<nom>.zip` et sera envoyé à `dominique.mery@loria.fr` avec le sujet "mvsi-tp".

Exercice 1 On considère le programme calculant le cube d'un nombre entier positif et on retrouvera ce programme dans le fichier `qq1.tla`. Le fichier contient l'algorithme PlusCal et l'annotation écrite à la suite du programme. Définir l'invariant de l'annotation proposée, puis vérifier la correction partielle et l'absence d'erreurs à l'exécution en définissant les bonnes questions. Vous complétez le fichier et vous essayez pour une valeur assez représentative par exemple 3. Au besoin vous ajoutez des explications comme commentaires dans le fichier `r1.tla` qui correspond à votre réponse.

Exercice 2 Nous considérons le fichier `qq2.tla` qui contient un algorithme PlusCal que vous devez vérifier. Cet algorithme est supposé calculer la puissance x^{x^2} quand cette valeur a un sens. En utilisant la boîte à outils de TLA/TLA⁺, produire un fichier `r2.tla` qui analyse la correction partielle et l'absence d'erreurs à l'exécution. Vous pourrez ajouter un commentaire dans le même fichier.

Exercice 3

Soit le petit algorithme annoté de la partie gauche de la figure ci-dessous. On construit l'algorithme PlusCal se trouvant à droite pour vérifier l'annotation. On suppose que a et b sont des constantes entières et que x, y, z et t sont des variables entières.

$$\begin{aligned} \ell_1 : & x = a \wedge z = x \wedge y = b * b \wedge t = b \\ & y := x * z + y + z * t \\ \ell_2 : & y = (x + t)^2 \end{aligned}$$

L'algorithme réalise un calcul et on demande d'utiliser TLA et son outil TLAPS pour montrer les éléments suivants:

- la correction des annotations.
- l'étude des dépassements de mémoire.

La réponse sera écrite dans le fichier `r3.tla`

Exercice 4

$$\begin{aligned} \ell_1 : & x = 81 \wedge z = 2 * c \wedge y = (z + 1)^2 \\ & y := x + z + 1 \\ \ell_2 : & x = 81 \wedge z = 2 * c \wedge y = (c + 1)^2 \end{aligned}$$

On suppose que la valeur de c est entière ($c \in \mathbb{Z}$) et on demande d'étudier les éléments suivants:

- la correction des annotations.
- l'étude des dépassements de mémoire.

Vous devez choisir une valeur de c et la donner explicitement dans votre fichier.

Fin de l'énoncé

```

----- MODULE q1 -----
EXTENDS Integers,TLC
CONSTANTS x  (* x is the input *)
-----

```

```

(*
-wfNext
--algorithm cube {
variables z=0,v=0,w=1,t=3,u=0;
{
10:skip;
w:while (u < x){
  11:
    z:=z+v+w;
  13:
    v:=v+t;
  14:
    t:=t+6;
  15:
    w:=w+3;
  16:
    u:=u+1;
  17:skip;
};
12:print <<z>>;
}
}
*)

```

```

Inv ==
pc \in {"10","w","11","12"}
=>
/\ w=3*u+1
/\ v=3*u^2
/\ z+v+w=(u+1)^3
/\ v+t=3*(u+1)^2
/\ t=3*(2*u+1)

```

```

I13 ==
pc="13"
=>
/\ w=3*u+1
/\ v=3*u^2
/\ z=(u+1)^3
/\ v+t=3*(u+1)^2

```

```

I14 ==
pc="14"
=>
/\ w=3*u+1
/\ v=3*(u+1)^2
/\ z=(u+1)^3
/\ v=3*(u+1)^2

```

```

I15 ==
pc="15"
=>
/\ w=3*u+1
/\ v=3*(u+1)^2
/\ z=(u+1)^3
/\ v=3*(u+1)^2

```

```

I16 ==
pc="16"

```

```

=>
/\ w=3*(u+1)+1
/\ v=3*(u+1)^2
/\ z=(u+1)^3
/\ v=3*(u+1)^2

I17 ==
pc="l7"
=>
/\ w=3*u+1
/\ v=3*u^2
/\ z+v+w=(u+1)^3
/\ v+t=3*(u+1)^2
/\ z=u^3

I12 == pc="l2" => w=3*x+1 /\ z=x^3

```

=====

```

----- MODULE qq2 -----
EXTENDS TLC, Integers, Naturals

CONSTANTS x1,x2

(*
--fair algorithm q2 {
  variables
    y1;
    y2;
    y3;
    z;
{
  10: print <<x1, x2>>;
  y1:=x1;
  y2:=x2;
  y3:=1;
  11:while (y2 /= 0) {
    12:if ( y2 % 2 # 0) {
      13:y2:=y2-1;
      14:y3:=y3*y1;
      15:skip;
    };
    16:y1 := y1*y1;
    17:y2:= y2 \div 2;
    18:skip;
  };
  19: z := y3;
  110: print <<x1, x2,z>>;
}
}
*)

```

=====