INTRODUCTION À LA THÉORIE DE L'INFORMATIQUE

ANNÉE ACADÉMIQUE 2013-2014
RÉPÉTITION 2

Machines d'états.

(1) Soit un **robot se déplaçant** sur une grille entière à deux dimensions. L'état du **robot** est spécifié par les coordonnées entières $(x; y)$ représentant sa position courante sur la grille. Le robot se trouve initialement à l'origine. À chaque pas, le robot a le choix entre ces 4 types de pas :

$$(+2; -1)\ , \quad (+1; -2)\ , \quad (+1; +1)\ , \quad (-3; 0).$$

(a) Modéliser ce problème par une machine d'état.

(b) Déterminer si le robot peut atteindre l'état $(0; 2)$. Dans l'affirmative, donner une suite de déplacements menant à $(0; 2)$. Dans la négative, donner une preuve utilisant le théorème d'invariant.

(2) Montrer que l'algorithme de **calcul du pgcd** figurant sur le transparent 93 du cours théorique est totalement correct pour les précondition et postcondition suivantes :

$\mathrm{Pre}((x; y)) \quad = \quad$ "$x > 0$ et $y > 0$" ;
$\mathrm{Post}((x; y)) \quad = \quad$ "x est le plus grand commun diviseur de a et b".

(3) D'après la rumeur, la procédure ci-dessous permet de **multiplier incognito** deux naturels quelconques de façon efficace.

procedure *multiply*(x, y: nonnegative integers)
$r := x$;
$s := y$;
$a := 0$;
while $s \neq 0$ **do**
    if $3 \mid s$ **then**
        $r := r + r + r$;
        $s := s/3$;
    else if $3 \mid (s - 1)$ **then**
        $a := a + r$;
        $r := r + r + r$;
        $s := (s - 1)/3$;
    else
        $a := a + r + r$;
        $r := r + r + r$;
        $s := (s - 2)/3$;
return a;

(a) Modéliser ce problème par une machine d'état.

(b) En utilisant le théorème d'invariant, montrer que l'algorithme est partiellement correct pour les précondition et postcondition suivantes :
Pre$((x\,;\,y))$ = "$x, y \in \mathbb{N}$;"
Post$((x\,;y))$ = "$a = x.y$"."

(c) Montrer que l'algorithme se termine toujours après au plus $\lfloor \log_3(y) \rfloor + 1$ exécutions du corps de la boucle.