TD: Machines à compteurs : simulation des MT

Définition informelle. Une machine à n compteurs (MàCn) a plusieurs registres x_1, x_2,..., x_n chacun capable de stocker un nombre naturel.

un programme - c'est un ensemble d'instructions de types suivants:

1. q: $x++$; **goto** p
2. q: $x--$; **goto** p
3. q:**if** $x = 0$ **then goto** p **else goto** r
4. q: **stop**

La tentative de calculer $0--$ produit une erreur. La configuration de la MàCn est le $n+1$-uplet $(q,x_1,x_2,\ldots,x_n)$ qui représente l'étiquette de l'instruction courante et le contenu de tous les compteurs. On suppose que pour l'entrée x la machine démarre dans la configuration (init,x,0,...,0) où "init" est une instruction particulière, et qu'elle calcule jusqu'à un **stop** . Le résultat de ce calcul est le contenu du registre x_1 après l'arrêt.

Le but de ce TD consiste à prouver l'indécidabilité du problème d'arrêt pour les machines à 4 compteurs (et équivalence des MàC4 aux MT)

1. Programmer une machine à 2 compteurs (MàC2) qui calcule $f(x) = x \bmod 2$.
2. Simuler une pile (pour simplicité sur $\{0,1\}$) par 2 compteurs:
   - Comment représenter le contenu de pile $a_0a_1a_2\ldots$ par des valeurs de x_1 et x_2?
   - Quelle transformation de x_1,x_2 correspond a l'opération Empiler(0)? Programmer cette transformation sur la MàC.
   - Même question pour Empiler(1)
   - Même question pour Dépiler;
   - Même question pour **if** Sommet=0 **then goto** p **else goto** r
3. Simuler 2 piles sur 4 compteurs
4. Prouver qu'une MàC4 peut calculer chaque fonction récursive partielle
5. Prouver que le problème d'arrêt est indécidable pour les MàC4

Questions subsidiaires

1. Prouver que le problème d'arrêt est indécidable pour les MàC2
2. Prouver que le problème d'arrêt est décidable pour les MàC1