Corrigé examen Calculabilité et Complexité

Information

Les exo 4 de la page 6 sur les langages de programmation PA et 2 de la dernière page sur les points fixes et transformation ne seront pas demandé.

Exo 2 année 2008/09 page 6

Q : Considérez la fonction suivante :

$$\forall n \in \mathbb{N}, f(n) = \begin{cases} f(n \dot{-} 2) * 2 & si\ n > 0 \\ 0 & si\ n = 0 \end{cases}$$

Est-ce que f est primitive récursive ? Est-ce que f est p.p.r. ?

R :

$f(0) = 0$

$f(1) = f(0) * 2$

$f(2) = f(0) * 2$

$f(3) = f(1) * 2 = 0$

$f(4) = f(2) * 2 = 0$

Je démontre par induction sur n que $f(n) = 0$

Base : $n = 0, f(0) = 0$

Hypothèse d'induction : $\forall i \leq n\ f(i) = 0$

On démontre que $f(n+1) = 0$

$f(n+1) = f\big((n+1) \dot{-} 2\big) * 2 = f(n \dot{-} 1) * 2$

Ainsi par hypothèse d'induction, $f(n \dot{-} 1) = 0$

Donc $f(n) = \mathcal{Z}()$

De ce fait, f est pr, et donc p.p.r.

Exo 3 année 2008/09 page 6

Q : Trouver la fonction calculée par le programme du listing ci-contre. Rappelons que les macros $\prod_g(n)$ et $\prod_d(n)$, vues enc ours, servent à extraire la partie gauche de n et la partie droite de n, respectivement.

Par exemple, $\prod_g(n) = x$ et $\prod_d(n) = y$ si $n = < x, y >$

$R_1 = \prod_g(R_0)$

$R_3 = \prod_d(R_0)$

$R_3 = R_3 + 1$

$R_3 = R_3 \dot{-} 1$

$R_1 = R_1 + 1$

$if\ R_3\ \neq 0\ then\ gotob\ 2$

R :

PC	R_0	R_1	R_3
1	$< x, y >$	0	0
2	$< x, y >$	x	0
3	$< x, y >$	x	y
4	$< x, y >$	x	$y + 1$
5	$< x, y >$	x	y
6	$< x, y >$	$x + 1$	y

$y = 0$

PC	R_0	R_1	R_3
7	$< x, y >$	$x + 1$	y

$y > 0$

PC	R_0	R_1	R_3
4	$< x, y >$	$x + 1$	y
5	$< x, y >$	$x + 1$	$y \dot{-} 1$
6	$< x, y >$	$x + 2$	$y \dot{-} 1$

$R_1 = x + y + 1$

$R_3 = 0$

$f(< x, y >) = \sigma(\prod_g(< x, y >) + \prod_d(< x, y >))$

Exo 3 année 2006/07 page 1

Q : Ecrivez un programme RAM dont la fonction calculée n'est pas primitive récursive

R : On sait que tout les programmes RAM sont p.p.r., donc pour en donner un non pr, il faut donc qu'il rende bottom

Par exemple un programme qui boucle à l'infini comme celui-ci :

$R_1 = R_1 + 1$

$if\ R_1\ \neq 0\ then\ gotob\ 1$

Exo 2 année 2007/08 page 3

Q : Considérez la fonction suivante :

$$\forall n \in \mathbb{N}, f(n) = \begin{cases} f(n/2) + 2 & si\ n > 0 \\ 0 & si\ n = 0 \end{cases}$$

Est-ce que f est primitive récursive ? Est-ce que f est p.p.r. ?

R :

$f(0) = 0$

$f(1) = f(0) + 2 = 2$

$f(2) = f(1) + 2 = 4$

$f(3) = f(1) + 2 = 4$

$f(4) = f(2) + 2 = 6$

$f(5) = f(2) + 2 = 6$

$f(6) = f(3) + 2 = 6$

$f(7) = f(3) + 2 = 6$

$f(8) = f(4) + 2 = 8$

$\Longrightarrow$

$f(2^0) = 2$

$f(2^1) = 2^2$

$f(2^2) = 2^2 + 2$

$f(2^3) = 2^3$

$f(2^4) = 2^3 + 2$

$f(2^5) = 2^3 + 2^2$

$f(2^6) = 2^3 + 2^2 + 2$

$$f(n) = \begin{cases} \sum_{i=1}^{k} 2^i, \quad tel\ que\ 2^n = 2^{\sum_{i=0}^{k} i} \quad avec\ k\ definie\ par\ minimalisation\ born\acute{e} \\ de\ l'ordre\ de\ \lfloor \log_2 n \rfloor \end{cases}$$

k existe toujours ici, donc $\sum_{i=1}^{k} 2^i$, $tel\ que\ 2^n = 2^{\sum_{i=0}^{k} i}$ est suffisant