Hertentamen Programmatuur
27 april 2009

Het tentamen bestaat uit 5 opgaven. Lees de opgaven zorgvuldig, schrijf duidelijk en beargumenteer je antwoorden. Schrijf op elk ingeleverd blad je naam en je collegekaartnummer. Je mag gebruik maken van het boek *Programma-Algebra, een Inleiding tot de Programmatuur*, I. Bethke en A. Ponse, Vossiuspers UvA, 2003. Het gebruik van een rekenmachine is NIET toegestaan. Succes!

1. Zij $X = +a; (\#2; (b; +c; !)^\omega; c)^\omega$ en $Y = +a; \#4; +c; (!; b; +a; !; b; +a)^\omega$.
   (a) Bepaal voor X en Y een eerste canonieke vorm. Laat duidelijk zien welke van de vergelijkingen PGA1–4 je gebruikt!
   (b) Zijn X en Y instructierij-congruent? Beargumenteer!
   (c) Bepaal voor X en Y de minimale tweede canonieke vorm. Laat duidelijk zien welke van de vergelijkingen PGA1–8 je gebruikt!
   (d) Zijn X en Y structureel congruent? Beargumenteer!

2. Zij $X = +a; !; +b$ en $Y = -a; \#2; !; b$.
   (a) Laat zien dat X en Y gedragsequivalent zijn.
   (b) Zijn $+c; X^\omega$ en $-c; \#3; Y^\omega$ gedragsequivalent? Zo ja, bewijs dan met volledige inductie dat $\pi_n(|+c; X^\omega|) = \pi_n(|-c; \#3; Y^\omega|)$ voor alle $n \in \mathbb{N}$. Zo nee, vind dan een $n \in \mathbb{N}$ zo dat $\pi_n(|+c; X^\omega|) \neq \pi_n(|-c; \#3; Y^\omega|)$.

3. Een *stapel* is een datastructuur voor de opslag van een aantal elementen waarbij geldt dat het element, dat het laatst werd toegevoegd, het eerst wordt opgehaald. Een stapel is te vergelijken met een stapel borden: het laatste bord, dat je op de stapel hebt gelegd, pak je het eerst weer van af. Een nog betere vergelijking is een *bordenwagen*, waarbij alleen het bovenste element zichtbaar is, en de eventuele rest in het interieur verdwijnt.

   In deze opgave beschouwen we het input/outputmodel op stapels over het alfabet $\{0, 1\}$. We geven met ε de lege stapel weer en met $0s$ en $1s$ stapels met topelement 0 respectievelijk 1 (d.w.z. we schrijven stapels gemakshalve horizontaal van links naar rechts en niet verticaal van boven naar beneden). De acties zijn $push0, push1, pop, top0$ en $top1$ met

   $$effect_{push0}(s) = 0s, \qquad effect_{push1}(s) = 1s,$$
   $$effect_{pop}(s) = \begin{cases} s' & \text{als } s = 0s' \text{ of } s = 1s' \\ \varepsilon & \text{als } s = \varepsilon, \end{cases}$$
   $$effect_{top0}(s) = s, \qquad effect_{top1}(s) = s.$$

   Dus $push0$ legt een 0 boven op de stapel s en $push1$ een 1; pop verwijdert het bovenste element uit een niet-lege stapel; $top0$ en $top1$ dienen als testinstructies en veranderen een stapel niet. Als yieldfuncties gebruiken we $y_{push0}(s) = y_{push1}(s) = \mathtt{true}$ en

   $$y_{pop}(s) = \begin{cases} \mathtt{true} & \text{als } s \neq \varepsilon, \\ \mathtt{false} & \text{anders,} \end{cases} \qquad y_{topb}(s) = \begin{cases} \mathtt{true} & \text{als } s = bs', \\ \mathtt{false} & \text{anders} \end{cases}$$

   $(b \in \{0, 1\})$.

(a) Bereken de volgende paden:

i. $|(+top0; \#3; -top1; !; pop)^\omega| \bullet^{vpad} 110\varepsilon$

ii. $|(-pop; !; push0; push0; push1)^\omega| \bullet^{vsvpad} 0\varepsilon$

(b) We duiden met 1^+ alle niet-lege stapels aan waarop slechts 1'en liggen. Schrijf een PGLE-programma X met de eigenschap

$$|X|_{pgle} \bullet^{vsvpad} s = \begin{cases} \varepsilon, \surd & \text{als } s = \varepsilon, \\ s, \ldots, 1\varepsilon, \surd & \text{als } s \in 1^+, \\ s, \ldots, 0\varepsilon, \surd & \text{anders.} \end{cases}$$

4. Zij X het PGLE-programma $+a; \#\#\pounds1; \#\#\pounds2; \pounds1; b; \#\#\pounds1; \pounds2; c; !$.

(a) Pas uitgaande van X de projecties `pgle2pgldg`; `pgldg2pgld`, `pgld2pglc`, `pglc2pglb` en `pglb2pgla` toe.

(b) Geef een PGAucw-programma bestaande uit hoogstens 3 instructies dat hetzelfde gedrag definieert als X.

5. In deze opgave beschouwen we een op MPP gebaseerd model voor de *gehele* getallen. Ieder geheel getal heeft in dit model een s-veld dat naar zijn opvolger en een p-veld dat naar zijn voorganger verwijst. Getallen kleiner dan 0 hebben bovendien een reflexief *neg*-veld en getallen groter dan 0 een reflexief *pos*-veld. 0 heeft geen reflexief veld. Het model bestaat uit de volgende programma's:

$setZero = Z = new; y = Z$
$pred = -x/p\{; x. + p; y = new; y. + neg; y. + s; y.s = x; x.p = y; \}\{; y = x.p; \}$
$succ = -x/s\{; x. + s; y = new; y. + pos; y. + p; y.p = x; x.s = y; \}\{; y = x.s; \}$

(a) Teken de molecule die door het programma

$$setZero; x = y; succ; x = Z; pred$$

wordt verkregen.

(b) Op de gehele getallen kunnen we aftrekken recursief definieren als volgt:

$$x1 - x2 = \begin{cases} x1 & \text{als } x2 = 0, \\ (x1 + 1) - (x2 + 1) & \text{als } x2 < 0, \\ (x1 - 1) - (x2 - 1) & \text{als } x2 > 0. \end{cases}$$

Schrijf een PGLEc-programma dat deze definitie implementeert. Je mag hierbij gebruik maken van de afkortingen *setZero*, *succ* en *pred* (zoals onder (a)).