Academiejaar 2007-2008, 14 januari 2008, 08.30u

Examen: Toepassingsgerichte Formele Logica 1

1. Voor het programmeren in lambdacalculus maken we gebruik van de lambdacombinatoren **Tru** en **Fls** die gedefinieerd zijn als $\mathbf{Tru} = \lambda xy.x$ en $\mathbf{Fls} = \lambda xy.y$. Merk op dat $\mathbf{Tru}\,MN = M$ en $\mathbf{Fls}\,MN = N$ voor willekeurige lambdatermen M en N. Voor het voorstellen van het natuurlijk getal n beschikken we over de lambdacombinator $\overline{\mathbf{n}}$. Verder hebben we de combinatoren **Iszero**, **Pred**, **Succ** en **Prod** voor het rekenen met natuurlijke getallen. Je mag er vanuit gaan dat deze combinatoren zich op de verwachte manier gedragen, m.a.w. $\mathbf{Iszero}\,\overline{\mathbf{n}} = \mathbf{Fls}$ indien $n \neq 0$ en $\mathbf{Iszero}\,\overline{\mathbf{0}} = \mathbf{Tru}$. Verder geldt $\mathbf{Pred}\,\overline{\mathbf{n}} = \overline{\mathbf{n-1}}$ indien $n \neq 0$ en $\mathbf{Pred}\,\overline{\mathbf{0}} = \mathbf{Fls}$. Tenslotte geldt $\mathbf{Succ}\,\overline{\mathbf{n}} = \overline{\mathbf{n+1}}$ en $\mathbf{Prod}\,\overline{\mathbf{n}}\,\overline{\mathbf{m}} = \overline{\mathbf{n \cdot m}}$.

   (a) Ontwerp een lambdacombinator **Fac** om de faculteit van een natuurlijk getal te berekenen. Voor de implementatie van recursie kan je gebruik maken van de fixpuntcombinator $\mathbf{Y} = \lambda x.(\lambda y.x(yy))(\lambda y.x(yy))$.

   (b) Toon aan dat je lambdacombinator het juiste gedrag vertoont voor inputgegevens 0 en 3, m.a.w. toon aan dat $\mathbf{Fac}\,\overline{\mathbf{0}} = \overline{\mathbf{1}}$ en dat $\mathbf{Fac}\,\overline{\mathbf{3}} = \overline{\mathbf{6}}$.

2. (a) Geef een calculationeel bewijs voor stelling 59(a)

$$(x \Rightarrow y) \wedge (y \Rightarrow z) \Rightarrow (x \Rightarrow z)$$

   Je mag gebruik maken van de axioma's en van stellingen 1 t.e.m. 58 zonder deze afzonderlijk te bewijzen. Indien je gebruik wenst te maken van andere stellingen, dien je die wel uitdrukkelijk te bewijzen.

   (b) Leg het verschil tussen een tautologie en een stelling uit.

   (c) Is onderstaande gevolgtrekking geldig of niet? Staaf je antwoord met een calculationeel bewijs of een tegenvoorbeeld.

   De uitvoering van het programma stopt niet of n wordt uiteindelijk nul (of allebei). Als n nul wordt, dan wordt m uiteindelijk ook nul. De uitvoering van het programma stopt. Dus m wordt uiteindelijk nul.

3. In deze opgave mag je steunen op de eigenschap $(\forall x|Rx\,.\,Px) \wedge Q \Rightarrow (\forall x|Rx\,.\,Px \wedge Q)$ met x niet vrij in Q.

   (a) Toon aan dat

$$(\exists x|Rx\,.\,Px \vee Q) \Rightarrow (\exists x|Rx\,.\,Px) \vee Q$$

   met x niet vrij in Q. Lever een calculationeel bewijs.

   (b) Toon aan dat

$$(\exists x|Rx\,.\,Q) \Rightarrow Q$$

   met x niet vrij in Q. Lever een calculationeel bewijs.

(c) Stel dat X een niet lege verzameling van locaties is. Voor locaties x en y uit X betekent $x \succ y$ dat x ten noorden ligt van y. Een regio is een deelverzameling van X. Er zijn verschillende manieren om te definiëren dat regio A ten noorden ligt van regio B. Een voorbeeld hiervan is:

$$N(A, B) \equiv (\forall x \,.\, x \in A \Rightarrow (\exists y \,.\, x \succ y \wedge y \in B))$$

Bewijs dat voor willekeurige regio's A, B en C geldt dat A ten noorden ligt van C als A ten noorden ligt van B en B ten noorden ligt van C. Je mag er hierbij vanuit gaan dat de relatie $\succ$ transitief is, m.a.w. $x \succ y \wedge y \succ z \Rightarrow x \succ z$ voor willekeurige locaties x, y en z. Lever een calculationeel bewijs.

4. Welke lijst wordt gegenereerd door deze Haskell-code?

```
[e | e <- [1..20], [x | x <- [1..e], x * x == e] == []]
```

Veel succes!