Inleiding Theoretische Informatica – Lambda Calculus

Uitwerkingen van geselecteerde opgaven

Martijn Vermaat (mvermaat@cs.vu.nl)

28 februari 2008

1 Termen en reductie

2. Een functie die twee inputs neemt en ze afbeelt op de constante $\mathsf{0}$ is $\lambda x.\lambda y.\mathsf{0}$ of $\lambda xy.\mathsf{0}$.

3. Met alle haakjes en alle λ's.

   (a) $((\lambda x.(\lambda y.(x\,(y\,z)))) \quad (\lambda x.((y\,x)\,x)))$

4. Met zo min mogelijk haakjes.

   (b) $(\lambda x.x) \quad (\lambda x.\lambda y.x\,y\,y) \quad \lambda x.\lambda y.x\,(x\,y)$

7. Zijn de termen α-convertibel?

   (a) Nee.
   (c) Nee.
   (e) Nee.
   (h) Ja.

8. Termen na uitvoeren van de substituties.

   (b) $(\lambda x.\mathsf{mul}\,x\,y)$
   (c) $(\lambda x.\mathsf{mul}\,x\,5)$
   (h) $(\lambda z.(\lambda y.\mathsf{plus}\,x\,y)\,z)$

10. De termen worden gereduceerd naar normaalvorm en β-redexen zijn steeds onderstreept.

    (a) $\underline{(\lambda y.\mathsf{mul}\,3\,y)\,7} \quad \rightarrow_\beta \quad \mathsf{mul}\,3\,7 \quad \rightarrow_\delta \quad 21$
    (d)

$$\begin{aligned} \underline{(\lambda f\,x.f\,x) \quad (\lambda y.\mathsf{plus}\,x\,y)} \quad 3 \quad &\rightarrow_\beta \quad \underline{(\lambda z.\underline{(\lambda y.\mathsf{plus}\,x\,y)\,z}) \quad 3} \\ &\rightarrow_\beta \quad \underline{(\lambda z.\mathsf{plus}\,x\,z) \quad 3} \\ &\rightarrow_\beta \quad \mathsf{plus}\,x\,3 \end{aligned}$$

(f)

$$
\begin{aligned}
\underline{(\lambda f\,x.f\,(f\,(x)))\;(\lambda x.\mathsf{mul}\,x\,7)}\;3 \quad &\rightarrow_\beta \quad \underline{(\lambda x.\underline{(\lambda x.\mathsf{mul}\,x\,7)\;(\underline{(\lambda x.\mathsf{mul}\,x\,7)\;(x)})})\;3} \\
&\rightarrow_\beta \quad \underline{(\lambda x.\underline{(\lambda x.\mathsf{mul}\,x\,7)\;(\mathsf{mul}\,x\,7)})\;3} \\
&\rightarrow_\beta \quad \underline{(\lambda x.\mathsf{mul}\,(\mathsf{mul}\,x\,7)\;7)\;3} \\
&\rightarrow_\beta \quad \mathsf{mul}\,(\mathsf{mul}\,3\,7)\;7 \\
&\rightarrow_\delta \quad \mathsf{mul}\,21\,7 \\
&\rightarrow_\delta \quad 147
\end{aligned}
$$

2 Reductiestrategieën

2. (a) Een reductierij volgens de leftmost-innermost strategie. Er is steeds maar één redex, dus is er geen verschil met het toepassen van bijvoorbeeld de leftmost-outermost strategie.

$$
\begin{aligned}
(\lambda x.x\,x\,x) \quad \lambda x.x\,x\,x \quad &\rightarrow_\beta \quad (\lambda x.x\,x\,x) \quad (\lambda x.x\,x\,x) \quad \lambda x.x\,x\,x \\
&\rightarrow_\beta \quad (\lambda x.x\,x\,x) \quad (\lambda x.x\,x\,x) \quad (\lambda x.x\,x\,x) \quad \lambda x.x\,x\,x \\
&\rightarrow_\beta \quad (\lambda x.x\,x\,x) \quad (\lambda x.x\,x\,x) \quad (\lambda x.x\,x\,x) \quad (\lambda x.x\,x\,x) \quad \lambda x.x\,x\,x \\
&\rightarrow_\beta \quad \ldots \text{etc.}
\end{aligned}
$$

(c) De leftmost-outermost strategie vindt altijd een normaalvorm als deze bestaat. We vinden in dit geval geen normaalvorm, dus heeft deze term geen normaalvorm.

3 Datatypes

9. (a) Een specificatie voor de operatie *exclusive or*.

$$
\begin{aligned}
\mathsf{xor\,true\,true} &=_\beta \mathsf{false} \\
\mathsf{xor\,true\,false} &=_\beta \mathsf{true} \\
\mathsf{xor\,false\,true} &=_\beta \mathsf{true} \\
\mathsf{xor\,false\,false} &=_\beta \mathsf{false}
\end{aligned}
$$

(b) Een mogelijke definitie voor de *exclusive or* operatie als λ-term. Volg de tip bij de vraag op voor een idee hoe te beginnen aan deze definitie.
$\mathsf{xor} := \lambda ab.b\,(a\,\mathsf{false\,true})\,a$

(c) Een afleiding voor de tweede regel van de specificatie bij (a). We vullen de definities van xor, true en false in en laten zien dat de term reduceert naar true.

$$
\begin{aligned}
\mathsf{xor\,true\,false} \quad &= \quad (\lambda ab.b\,(a\,\mathsf{false\,true})\,a)\;(\lambda xy.x)\;\lambda xy.y \\
&\rightarrow_\beta \quad (\lambda b.b\,((\lambda xy.x)\,\mathsf{false\,true})\,(\lambda xy.x))\;\lambda xy.y \\
&\rightarrow_\beta \quad (\lambda xy.y)\;((\lambda xy.x)\,\mathsf{false\,true})\;\lambda xy.x \\
&\rightarrow_\beta \quad (\lambda y.y)\;\lambda xy.x \\
&\rightarrow_\beta \quad \lambda xy.x \\
&= \quad \mathsf{true}
\end{aligned}
$$

4 Recursie

7. We zijn op zoek naar een term map die het volgende gedrag vertoont:

$$\mathsf{map} =_\beta \lambda f l.(\mathsf{empty}\, l)\, \mathsf{nil}\, (\mathsf{cons}\, (f\, (\mathsf{head}\, l))\, (\mathsf{map}\, f\, (\mathsf{tail}\, l)))$$

Dit kunnen we ook schrijven als

$$\mathsf{map} =_\beta (\lambda m.\lambda f l.(\mathsf{empty}\, l)\, \mathsf{nil}\, (\mathsf{cons}\, (f\, (\mathsf{head}\, l))\, (m\, f\, (\mathsf{tail}\, l))))\, \mathsf{map}$$

Nu kunnen we de fixed-point combinator Y gebruiken om de recursie op te lossen en een uiteindelijke definitie van map te maken:

$$\mathsf{map} = \mathsf{Y}\, \lambda m.\lambda f l.(\mathsf{empty}\, l)\, \mathsf{nil}\, (\mathsf{cons}\, (f\, (\mathsf{head}\, l))\, (\mathsf{map}\, f\, (\mathsf{tail}\, l)))$$

5 Getypeerde λ-calculus

1. (c) De term $\mathsf{K}\,\mathsf{I}$ is gelijk aan $(\lambda xy.x)\, \lambda x.x$.

$$\dfrac{\dfrac{x : C \vdash x : C}{\vdash \lambda x.x : C \to C} \qquad \dfrac{\dfrac{x : C \to C, y : B \vdash x : C \to C}{x : C \to C \vdash \lambda y.x : B \to C \to C}}{\vdash \lambda xy.x : (C \to C) \to B \to C \to C}}{\vdash (\lambda xy.x)\, \lambda x.x : B \to C \to C}$$