

tentamen inleiding theoretische informatica
24 augustus 2005

Opgave 1.

- (a) Teken de volgende term als boom:

$$M = (\lambda u. (\lambda v. u) v u) (\lambda x. x y x)$$

(2 punten)

- (b) Geef twee verschillende α -equivalente varianten van de term M uit 1a.

(4 punten)

- (c) Reduceer de term $(\lambda x. x x) (\lambda y. (\lambda z. y) z)$ naar β -normaalkvorm (schrijf elke stap op).

(4 punten)

Opgave 2.

- (a) We representeren paren met de paringsoperator $\pi = \lambda l r z. z l r$.

Geef de eerste projectiefunctie π_1 en laat zien dat er geldt $\pi_1 (\pi P Q) \rightarrow_\beta P$.

(5 punten)

- (b) Geef de definitie in de λ -calculus van de booleans true en false.

(5 punten)

- (c) Geef de definitie in de λ -calculus van not en laat zien dat not true reduceert naar false.

(5 punten)

- (d) Stel $\text{true} : \text{bool}$. Wat is dan het type van $(\lambda x. x) \text{true}$?

(De typeringsregels staan hieronder.)

(5 punten)

De typeringsregels van simpel getypeerde λ -calculus:

$$\Gamma, x : A \vdash x : A$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x. M) : A \rightarrow B}$$

$$\frac{\Gamma \vdash F : A \rightarrow B \quad \Gamma \vdash M : A}{\Gamma \vdash (F M) : B}$$

Opgave 3. We gebruiken hier de notatie $I = \lambda x. x$ en $\Omega = (\lambda x. x x) (\lambda x. x x)$.

- (a) Reduceer de term $(\lambda x. y) (I \Omega)$ volgens de leftmost-innermost reductiestrategie.
(2 punten)
- (b) Is de leftmost-innermost reductiestrategie normalizerend? Waarom (niet)?
(3 punten)
- (c) Wat is de weak head normal form (WHNF) van $(\lambda x. \lambda y. x) \Omega$?
(5 punten)
- (d) Heeft de term $\Omega \Omega \Omega$ en WHNF? Waarom (niet)?
(5 punten)

Opgave 4. Gegeven is de volgende specificatie **Herk**:

module **Herk**

sorts **elt**

functions

n	:		$\rightarrow$	elt
e	:		$\rightarrow$	elt
p	:	elt	$\rightarrow$	elt
s	:	elt	$\rightarrow$	elt

equations

[E1]	:	$p(s(x)) = x$
[E2]	:	$s(p(x)) = x$
[E3]	:	$s(n) = e$

end **Herk**

Voor deze specificatie beschouwen we de volgende Σ -algebra's $\mathcal{K}$, $\mathcal{L}$ en $\mathcal{M}$.

$\mathcal{K}$: $K = \{-1, 0, 1\}$, $n_{\mathcal{K}} = -1$, $e_{\mathcal{K}} = 1$, $s_{\mathcal{K}}(x) = -x$, $p_{\mathcal{K}}(x) = -x$

$\mathcal{L}$: $L = \mathbb{Z}$, $n_{\mathcal{L}} = 0$, $e_{\mathcal{L}} = 1$, $s_{\mathcal{L}}(x) = x + 1$, $p_{\mathcal{L}}(x) = x - 1$

$\mathcal{M}$: $M = \mathbb{R}$, $n_{\mathcal{M}} = 0$, $e_{\mathcal{M}} = 1$, $s_{\mathcal{M}}(x) = x + 1$, $p_{\mathcal{M}}(x) = x - 1$

De algebra's $\mathcal{K}$, $\mathcal{L}$, $\mathcal{M}$ zijn modellen voor de specificatie **Herk**. Dit hoeft u niet te bewijzen.

- (a) Geef een afleiding voor de volgende vergelijking:

$$s(s(p(s(n)))) = s(e)$$

(5 punten)

- (b) Geef een vergelijking [E3] die waar is in $\mathcal{K}$, maar niet in $\mathcal{L}$.

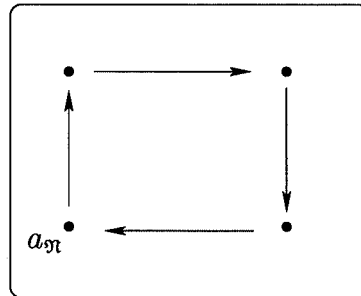
(3 punten)

- (c) Ga voor elk van de algebra's $\mathcal{K}$, $\mathcal{L}$, $\mathcal{M}$ na of het een initieel model is voor $\mathbf{Herk}$. Geef voor de modellen die niet initieel zijn de junk en/of confusion aan.

(10 punten)

Opgave 5. Gegeven is een signatuur Σ met een constante, a en een eenplaatsig functiesymbool f . Verder is gegeven de Σ -algebra $\mathfrak{N}$, die hieronder is getekend. (de interpretatie van f is met pijlen aangegeven.)

$\mathfrak{N}$:



- (a) Geef een initieel correcte specificatie E van $\mathfrak{N}$.

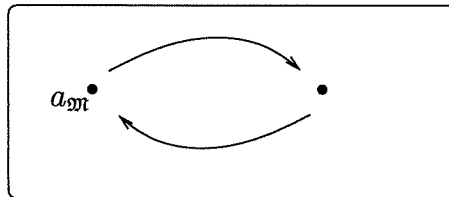
(5 punten)

- (b) Geef voor de gevonden specificatie een kanonieke termalgebra.

(5 punten)

- (c) Voor dezelfde signatuur is hier nog een Σ -algebra $\mathfrak{M}$ getekend.

$\mathfrak{M}$:



- (i) Bestaat er een homomorfisme φ van $\mathfrak{N}$ naar $\mathfrak{M}$?

- (ii) Bestaat er omgekeerd een homomorfisme ψ van $\mathfrak{M}$ naar $\mathfrak{N}$?

Motiveer je antwoorden, in het geval er een homomorfisme bestaat door het te tekenen.

(5 punten)

Opgave 6. Deze opgave gaat over OCaml.

- (a) Definieer in OCaml een functie `som` die met een lijst van integers als input de som van die integers retourneert.

Dus het gebruik van `som` moet bijvoorbeeld het volgende effect hebben:

```
# som [5;6;2;0];;  
- : int = 13
```

(4 punten)

- (b) Definieer in OCaml een functie `at_end` die gegeven een data-element en een lijst dat element aan het eind van de lijst toevoegt.

Dus het gebruik van `at_end` moet bijvoorbeeld het volgende effect hebben:

```
# at_end 4 [1;6;2];;  
- : int list = [1;6;2;4]  
# at_end 4 [];;  
- : int list = [4]
```

(4 punten)

- (c) Definieer in OCaml een functie `keer` die de volgorde van de elementen in een lijst omkeert.

Dus het gebruik van `keer` moet bijvoorbeeld het volgende effect hebben:

```
# keer [1;6;2;4];;  
- : int list = [4;2;6;1]  
# keer [12];;  
- : int list = [12]
```

Je kunt hier de functie `at_end` uit onderdeel (b) gebruiken.

(4 punten)

Het tentamencijfer is (het totaal aantal punten plus 10) gedeeld door 10.