

tentamen inleiding theoretische informatica
22 augustus 2007

Opgave 1.

- a. Gegeven de volgende equationele specificatie voor stacks over een data-domein $\text{data} = \{d_1, \dots, d_n\}$, met een error-element `error`.

`sorts data, stack`

`functions`

```
d1:          -> data
...
dn:          -> data
error :      -> data
empty :      -> stack
push  : data # stack -> stack
pop   : stack   -> stack
top   : stack   -> data
```

`equations`

```
pop(empty)    = empty
top(empty)     = error
pop(push(x,s)) = s
top(push(x,s)) = x
```

- (a) Geef vergelijkingen voor de functie

`reverse : stack -> stack`

die een stack omdraait. Bijvoorbeeld:

```
reverse(push(d,push(e,push(f,empty))))
= push(f,push(e,push(d,empty)))
```

(5 punten)

- (b) Geef vergelijkingen voor de functie

`concatenate : stack # stack -> stack`

die twee stacks aan elkaar plakt. Bijvoorbeeld:

```
concatenate(push(d,push(e,empty)),push(f,push(g,empty)))
= push(d,push(e,push(f,push(g,empty))))
```

(5 punten)

Opgave 2. Gegeven een signatuur met soort S en functiesymbolen $a : \rightarrow S$, $b : \rightarrow S$, $c : \rightarrow S$ en $f : S \rightarrow S$. De bijbehorende equationele specificatie bestaat uit twee vergelijkingen:

$$\begin{aligned} f(f(b)) &= a \\ f(c) &= a \end{aligned}$$

- Geef een afleiding voor $f(f(c)) = f(f(f(b)))$. (3 punten)
- Geef een model voor deze equationele specificatie, met een drager die uit vier elementen bestaat. Dit model mag wel confusion bevatten, maar geen junk. (9 punten)
- ~~Beschrijf in detail het termmodel als een initieel model voor deze equationele specificatie.~~ (12 punten)

Opgave 3.

- De term M is gedefinieerd als

$$(\lambda uv. u u) ((\lambda x. x) z) ((\lambda y. y) z)$$

Teken M als boom.

- (2 punten)
- Geef een reductierij van M naar normaalvorm met zo min mogelijk stappen. Schrijf elke stap expliciet op. (5 punten)
- Geef een voorbeeld van een reductiestap waarbij α -conversie nodig is. (4 punten)
- Geef een voorbeeld waaruit blijkt dat de leftmost-outermost (call by name) strategie onnodige stappen kan doen. (5 punten)

Opgave 4.

- We representeren de booleans als $\text{true} = \lambda xy. x$ en $\text{false} = \lambda xy. y$.
Geef een λ -term and voor conjunctie en laat zien: $\text{and true true} \rightarrow_{\beta} \text{true}$. (5 punten)
- We gebruiken de definitie van de Church numerals: $c_n = \lambda s. \lambda z. s^n(z)$.
Geef een λ -term S voor successor en laat zien dat geldt: $S c_1 \rightarrow_{\beta} c_2$. (5 punten)

Opgave 5.

- a. Laat zien dat

$$Y = \lambda f. (\lambda x. f (x x)) (\lambda x. f (x x))$$

een fixed point combinator is.

(5 punten)

- b. Geef een λ -term som voor de functie die als input een lijst van natuurlijke getallen neemt, en als output geeft de som van alle elementen van die lijst (en 0 als de input-lijst leeg is).

Je mag gebruikmaken van de functies `empty`, `plus` (voor optellen), `head`, `tail`, en een fixed point combinator.

(5 punten)

- c. Laat zien dat de term $\lambda x. \lambda y. x y y$ typeerbaar is in de lege omgeving door een typeringsafleiding te geven.

(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 6. We gebruiken het datatype voor binaire bomen met integer labels op de knopen:

```
type intbintree =  
  Leaf  
| Node of intbintree * int * intbintree ;;
```

- a. Geef een definitie in OCAML van de functie

`size_tree : intbintree -> int`

die als input een binaire boom neemt en als output het aantal knopen van die boom geeft.

(5 punten)

- b. Geef een definitie in OCAML van de functie

```
insert : int -> intbintree -> intbintree
```

die het input-getal toevoegt aan de input-boom als het nog niet als label voorkomt, en de input-boom niet verandert als het getal al wel als label voorkomt. De input-boom is een binaire zoek-boom, en de output-boom moet ook een binaire zoek-boom zijn.

(5 punten)

c. Geef een definitie in OCAML van de functie

```
make_search_tree : int list -> intbintree
```

die als input een lijst gehele getallen neemt en als output een binaire zoek-boom geeft met daarin de elementen van de lijst, maar zonder dubbele.

(5 punten)

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