Met werkexemplaar

Korrektheidsbewijs van een priemgetallenprogramma

Maarten Fokkinga, 22 aug 1985.

We bewijzen de korrektheid van een slim programma dat de lijst van alle priemgetallen voortbrengt, ~~gevonden in [G.F. van der Hoeven 1984]~~. Daarbij ontkomen we er niet aan om het volgende, niet bewezen, beginsel te gebruiken:

> Als twee lijsten voor willekeurige k (k eindig) gelijk zijn aan elkaar op de eerste k plaatsen, dan zijn de lijsten zelf gelijk aan elkaar.

* * *

Beschouw het volgende programma, genomen uit [G.F. van der Hoeven 1984].

```
primes = 2: 3: z
z = sieve [2] (3: z)
sieve P(q:Q) = (filter [p²+1 .. q²-1] P) ++ sieve (q:P) Q
                 where p = hd P
```

We nemen aan dat

$$\text{filter } X\ Y = \{x \leftarrow X;\ x \text{ niet deelbaar door enige } y \text{ uit } Y\}$$

Voor iedere aanroep sieve P (q:Q) wordt ervoor gezorgd dat



$P^R$ ++ (q:Q) de lijst van alle priemgetallen voorstelt, ook al zijn (nog) niet alle elementen van Q bepaald. Per aanroep van sieve wordt één element van (q:Q) overgeheveld naar P en worden tevens ál die priemgetallen bepaald ~~waarvoor~~ (die nog niet bepaald waren) waarvoor "ondeelbaarheid door enig lid van P" voldoende is voor hun "priemheid". Aldus wordt ieder getal slechts met de priemgetallen kleiner dan de wortel eruit vergeleken; dit is efficiënter dan het standaard zeefproces. Wellicht dat het formele korrektheidsbewijs deze informele verklaring nog verduidelijkt, (dat is wat mijzelf betreft wel het geval).

Notatie. We laten $p_1, p_2, p_3, \ldots$ de priemgetallen in opklimmende volgorde zijn. Soms benoemen we een subexpressie (met een griekse letter als prefix superscript); de naam van die subexpressie gebruiken we dan als "afkorting" ervoor. Dit kan ook met recursieve where-clauses en gewone identifiers worden gedaan, maar onze notatie is ietwat korter. [Deze zelfde notatie kan gebruikt worden om sharing tijdens lazy evaluation heel precies en toch beknopt weer te geven, zie mijn notitie "Lazy evaluation op expressie-nivo uitgelegd".]

De kern van de korrektheidsargumentatie ligt besloten in het volgende lemma plus bewijs.

<u>Hoofdlemma.</u> Zij $p_m$ de grootste priem kleiner dan $p_i^2$ en evenzo $p_n$ de grootste priem kleiner dan $p_{i+1}^2$. Dan

$$[p_1, \dots, p_m] \mathbin{++} {}^{\alpha}(\text{sieve } [p_1, \dots, p_i]^R \; ([p_{i+1}, \dots, p_m] \mathbin{++} \alpha))$$
$$= [p_1, \dots, p_n] \mathbin{++} {}^{\beta}(\text{sieve } [p_1, \dots, p_{i+1}]^R \; ([p_{i+2}, \dots, p_n] \mathbin{++} \beta))$$

<u>Bewijs</u>

Volgens de rekenkunde is $m \geq i+1$ (dus $p_m > p_i$); (dit is niet triviaal).

We gebruiken louter conversieregels (= ~~stam~~ betekenis van standaardoperatoren en definities van sieve en filter) en geen inductiebeginsel! In feite ontwikkelen we één aanroep van sieve.

$$[p_1, \dots, p_m] \mathbin{++} {}^{\alpha}(\text{sieve } [p_1, \dots, p_i]^R \; ([p_{i+1}, \dots, p_m] \mathbin{++} \alpha))$$

[Bijgevolg is ~~[illegible]~~ $hd([p_{i+1}, \dots, p_m] \mathbin{++} \alpha) = p_{i+1}$ -- of liever: 2de arg van sieve in de vorm $q:Q$ brengen]

= volgens def van sieve:

$$[p_1, \dots, p_m] \mathbin{++} {}^{\alpha}(\text{filter } [p_i^2+1 .. p_{i+1}^2-1] \; [p_1, \dots, p_i]^R) \mathbin{++} (\text{sieve } [p_1, \dots, p_i, p_{i+1}]^R \; ([p_{i+2}, \dots, p_m] \mathbin{++} \alpha))$$

= volgens aanname omtrent filter, plus rekenkunde:

$$[p_1, \dots, p_m] \mathbin{++} {}^{\alpha}([p_{m+1}, \dots, p_n] \mathbin{++} \text{sieve } [p_1, \dots, p_{i+1}]^R \; ([p_{i+2}, \dots, p_m] \mathbin{++} \alpha))$$

= volgens "definitie" van $\alpha$:

$$[p_1, \dots, p_m] \mathbin{++} ([p_{m+1}, \dots, p_n] \mathbin{++} {}^{\beta}(\text{sieve } [p_1, \dots, p_{i+1}]^R \; ([p_{i+2}, \dots, p_m] \mathbin{++} ([p_{m+1}, \dots, p_n] \mathbin{++} \beta))))$$

= associativiteit van ++ en [ ]-notatie:

$$[p_1, \dots, p_n] \mathbin{++} {}^{\beta}(\text{sieve } [p_1, \dots, p_{i+1}]^R \; ([p_{i+2}, \dots, p_n] \mathbin{++} \beta))$$

Q.E.D.

Door herhaaldelijk toepassen van dit lemma zien we

$$[2, 3] \mathbin{++} {}^{\alpha}(\text{sieve } [2]^R \; ([3] \mathbin{++} \alpha))$$
$$= [2, 3, 5, 7] \mathbin{++} {}^{\beta}(\text{sieve } [2, 3]^R \; ([5, 7] \mathbin{++} \beta))$$
$$= [2, 3, 5, 7, 11, 13, 17, 23] \mathbin{++} {}^{\gamma}(\text{sieve } [2, 3, 5]^R \; ([7, 11, 13, 17, 23] \mathbin{++} \gamma))$$
$$= [2, 3, 5, \dots, 23, \dots, 47] \mathbin{++} {}^{\delta}(\text{sieve } [2, 3, 5, 7]^R \; ([11, \dots, 23, \dots, 47] \mathbin{++} \delta))$$



$$= [p_1, \dots, p_m] \mathbin{++} {}^{\varepsilon}(\text{sieve } [p_1, \dots, p_i]^R \; [p_{i+1}, \dots, p_m] \mathbin{++} \varepsilon)$$

met $i$ willekeurig (en $p_m$ de grootste priem $< p_i^2$), ofwel met $m$ willekeurig groot (en $p_i$ de kleinste priem $> \sqrt{p_m}$).

Dus voor willekeurig groot beginstuk vallen $[p_1, p_2, p_3, \dots]$ en primes (= $[2,3] \mathbin{++} {}^{\alpha}(\text{sieve } [2]^R \; ([3] \mathbin{++} \alpha))$ ) samen. Op grond van het beginsel

"Als twee lijsten voor willekeurige eindige $k$ elementsgewijze aan elkaar gelijk zijn op de eerste $k$ plaatsen, dan zijn de lijsten zelf ook aan elkaar gelijk."

konkluderen we nu dat $[p_1, p_2, p_3, \dots] =$ primes. Hiermee is dan het korrektheidsbewijs voltooid.

* * *

[Turner 1982] geeft voor het volvoeren van korrektheidsbewijzen een aantal inductiebeginselen, namelijk:

- volledige inductie; konkludeer: $\forall n.\ P(n)$ uit: $P(0)$ <u>en</u> $\forall n.\ P(n) \Rightarrow P(n+1)$
- data-inductie voor eindige lijsten; konkludeer: $\forall$ eindige lijsten $X$. $P(X)$ uit: $P([\,])$ <u>en</u> $\forall x$, eindige $X$. $P(X) \Rightarrow P(x{:}X)$.
- <u>partial object induction</u> voor (on)eindige lijsten die niet eindigen met $[\,]$; konkludeer: $\forall$ "partiele" lijsten $X$.

$P(X)$ uit: $P(\perp)$ en $\forall x$,particle $X$. $P(X) \Rightarrow P(x{:}X)$.

Bij de eerste twee inductiebeginselen staat $P$ voor een willekeurig predicaat; bij partial object induction staat $P$ voor een zogenaamd "directedlycomplete" predicaat zoals formules van de vorm $e = e'$ waarbij $e$ en $e'$ programma's zijn. En $\perp$ staat voor "divergentie".

Mijn opmerking is nu dat ik er (nog) niet in geslaagd ben de korrektheid van primes te bewijzen met Turner's inductiebeginselen. Evenmin is "mijn" inductiebeginsel te bewijzen met die van Turner, althans als we alléén maar formules van de vorm $e = e'$ mogen gebruiken. Wanneer we echter ook een begrip $\sqsubseteq$ (semantische approximatie) invoeren en formules van de vorm $e \sqsubseteq e'$ toelaten (zij lijken mij wel "directedly complete" te zijn), dan zou partial object induction voldoende zijn om mijn beginsel af te leiden.

We konkluderen dat een beter begrip van "(semantische) gelijkheid" dringend nodig is opdat we niet keer op keer bij het korrekt bewijzen van programma's onze toevlucht moeten nemen tot het poneren van nieuwe inductieprincipes en nieuwe wetten voor $=$.

—#—



Literatuur

G.F. van der Hoeven: Preliminary Report on the Language Twentel. Memorandum INF-84-5, T.H.T, 1984.

Turner, D. A.: Functional Programming and proofs of program correctness. In: Tools and Notions of Program construction - an advanced course. (Ed. D. Neel) Cambridge Univ Press, Cambridge, U.K., 1982 pp 187-210.

<u>Naschrift</u>

Ook om de gelijkheid van primes aan (zeef (from 2)) te bewijzen, met

$$zeef\ (x{:}X) = x{:}\ zeef\ \{y \leftarrow X;\ x \text{ deelt } y \text{ niet}\}$$

lijkt toepassing van "mijn" inductiebeginsel ~~ook~~ onvermijdelijk (tenzij we het begrip semantische approximatie, $\sqsubseteq$, invoeren). En misschien is het nog wel het eenvoudigst om eerst primes $= [p_1, p_2, p_3, \ldots]$ en (zeef (from 2)) $= [p_1, p_2, p_3 \cdots]$ te bewijzen en dan daaruit te konkluderen dat primes $=$ (zeef (from 2)).