Programmeren & Correctheid

Docent: Prof. dr. F.S. de Boer, email: frb@cwi.nl

Literatuur
Verification of Sequential and Concurrent Programs.
Krzysztof R. Apt, Frank S. de Boer, Ernst-Rüdiger Olderog.
Series: Texts in Computer Science. Springer.
3rd ed. 2nd Printing. ISBN: 978-1-84882-744-8.

Te behandelen stof
Course Towards Object-Oriented Program Verification (zie *Preface*: *Outlines of One-Semester Courses* en slides). Uit bovenstaand boek behandelen we de hoofdstukken 2, 3, 4, en 5 onderverdeeld in de volgende blokken B1-3:

	Onderwerp	Secties uit het boek
B1	Partiële Correctheid While Programma's	2.1, 2.2, 2.4, 2.5, 2.7, 3.1, 3.3, 3.4 3.10, 3.11.
B2	Totale Correctheid While Programma's	3.3, 3.4 en 3.8..
B3	Partiële Correctheid Recursieve Programma's	4.1, 4.3, 5.1, 5.2, 5.3.

Evaluatie

- De behandelde stof (zie de paragraaf "Te behandelen stof" hierboven) wordt schriftelijk geëxamineerd .
- Daarnaast wordt er voor elke blok een *deeltentamen* gegeven. In totaal zijn er dus drie deeltentamens (t.z.t. worden de data bekend gemaakt).
    1. **Het eerste deeltentamen vindt plaats op vrijdag 6 April, 13.30 - 15.00**. De stof betreft blok B1.

Voor het eindcijfer telt het schriftelijk tentamen voor 75%. Het gemiddelde van de deeltentamens telt voor 25%.

Agenda We maken in de opgaven (en de uitwerkingen) gebruik van de volgende afkortingen:

$$\bigwedge_{i=1}^{n} p_i \quad \text{staat voor} \quad p_1 \wedge ... \wedge p_n$$

$$
\begin{array}{rll}
s \leq t & \text{staat voor} & s < t \lor s = t \\
s \leq t < u & \text{staat voor} & s \leq t \land t < u \\
s \neq t & \text{staat voor} & \neg(s = t) \\
\exists x, y : p & \text{staat voor} & \exists x : \exists y : p \\
\forall x, y : p & \text{staat voor} & \forall x : \exists y : p \\
\exists x \leq t : p & \text{staat voor} & \exists x : (x \leq t \land p) \\
\forall x \leq t : p & \text{staat voor} & \forall x : (x \leq t \to p) \\
\forall x \in [s : t] : p & \text{staat voor} & \forall x : (s \leq x \leq t \to p)
\end{array}
$$

In de slides maken we ook gebruik van andere voor zich sprekende afkortingen (als $\forall n : s \leq n \leq t : p$ voor $\forall n : (s \leq n \land n \leq t) \to p)$).

16-2 Introduktie (zie slides 1 t/m 21).
Opgaven: Zie slide 21.

23-2 Hoofdstuk 2, secties 2.1, 2.2, , 2.4, 2.5 en 2.7, en sectie 3.1 van hoofdstuk 3. Zie ook slides 22–33.
Opgaven:

1. Bereken de volgende substituties:
   (a) $(x = Y \land y = X)[y := z][x := y][z := x]$
   (merk op dat $z := x; x := y; y := z$ de waarden van de integer variabelen x en y verwisselt ("swapt").

   Uitwerking ($\equiv$ staat voor syntactische gelijkheid)

   $$
   \begin{array}{ll}
   (x = Y \land y = X)[y := z][x := y][z := x] & \equiv \\
   (x = Y \land z = X)x := y][z := x] & \equiv \\
   (y = Y \land z = X)[z := x] & \equiv \\
   (y = Y \land x = X) &
   \end{array}
   $$

   (b) $(a[i] = Y \land a[j] = X)[a[j] := z][a[i] := a[j]][z := a[i]]$

Uitwerking We rekenen eerst $(a[i] = Y \wedge a[j] = X)[a[j] := z]$ uit:

$$
\begin{array}{c}
(a[i] = Y \wedge a[j] = X)[a[j] := z] \\
\equiv \\
(a[i] = Y)[a[j] := z] \wedge (a[j] = X)[a[j] := z] \\
\equiv \\
a[i][a[j] := z] = Y[a[j] := z] \wedge a[j][a[j] := z] = X[a[j] := z] \\
\equiv \\
\textbf{if } i[a[j] := z] = j \textbf{ then } z \textbf{ else } a[i[a[j] := z]] \textbf{ fi} = Y \\
\wedge \\
\textbf{if } j[a[j] := z] = j \textbf{ then } z \textbf{ else } a[j[a[j] := z]] \textbf{ fi} = X \\
\equiv \\
\textbf{if } i = j \textbf{ then } z \textbf{ else } a[i] \textbf{ fi} = Y \wedge \textbf{if } j = j \textbf{ then } z \textbf{ else } a[j] \textbf{ fi} = X \\
\leftrightarrow \\
(i = j \wedge z = Y \wedge z = X) \vee (i \neq j \wedge a[i] = Y \wedge z = X)
\end{array}
$$

Vervolgens rekenen we

$$(i = j \wedge z = Y \wedge z = X) \vee (i \neq j \wedge a[i] = Y \wedge z = X)[a[i] := a[j]]$$

uit (we slaan nu enkele triviale stappen over):

$$
\begin{array}{c}
(i = j \wedge z = Y \wedge z = X) \vee (i \neq j \wedge a[i] = Y \wedge z = X)[a[i] := a[j]] \\
\equiv \\
(i = j \wedge z = Y \wedge z = X) \vee (i \neq j \wedge a[i][a[i] := a[j]] = Y \wedge z = X) \\
\equiv \\
(i = j \wedge z = Y \wedge z = X) \vee (i \neq j \wedge \textbf{if } i = i \textbf{ then } a[j] \textbf{ else } a[i] \textbf{ fi} = Y \wedge z = X) \\
\leftrightarrow \\
(i = j \wedge z = Y \wedge z = X) \vee (i \neq j \wedge a[j] = Y \wedge z = X)
\end{array}
$$

Tot slot, hebben we dat

$$
\begin{array}{c}
((i = j \wedge z = Y \wedge z = X) \vee (i \neq j \wedge a[j] = Y \wedge z = X))[z := a[i]] \\
\equiv \\
((i = j \wedge a[i] = Y \wedge a[i] = X) \vee (i \neq j \wedge a[j] = Y \wedge a[i] = X) \\
\leftrightarrow \\
a[j] = Y \wedge a[i] = X
\end{array}
$$

2. Stel dat we geen *gebonden* variabelen in p herbenoemen als we een substitutie $[u := t]$ toepassen (zie Sectie 2.7 van het boek). Laat zien dat er dan *ongeldige* instanties zijn van het assignment axioma, d.w.z., vind een postconditie p en een assignment $u := t$ zodat $\{p[u := t]\}\ u := t\ \{p\}$ *niet* waar is.
   Antwoord Neem bijvoorbeeld voor p de assertie $\exists y : x = 2 * y$ (die uitdrukt dat x even is). Als we de gebonden variable y niet *herbenoemen* zouden we het assignment axiom alsvolgt kunnen instantiëren

$$\{\exists y : y = 2 * y\}\ x := y\ \{\exists y : x = 2 * y\}$$

Merk op dat **true** $\rightarrow \exists y : y = 2 * y$ en dus kunnen we m.b.v. de consequence regel

$$\{\textbf{true}\}\ x := y\ \{\exists y : x = 2 * y\}$$

afleiden.

2-3 Hoofdstuk 3, sectie 3.3 (Partial Correctness). Zie ook slides 34 – 38.
Opgaven:

1. Opgave 3.4 (Boek) en voor welke statements gelden de volgende correctheidsbeweringen.
   (a) $\{\textbf{true}\}S\{\textbf{true}\}$
   Antwoord Alle statements.
   (b) $\{\textbf{false}\}S\{p\}$
   Antwoord Alle statements.
   (c) $\{\textbf{true}\}S\{\textbf{false}\}$
   Antwoord Alle statements die *niet* termineren.
2. Onderzoek of de assertie

   $$a[a[n]] \leq \textbf{if}\ a[a[n]] = a[n]\ \textbf{then}\ a[a[n]]\ \textbf{else}\ a[a[a[n]]]\ \textbf{fi}$$

   waar is voor alle arrays a (van type **integer** $\rightarrow$ **integer**). Zo niet, geef een tegenvoorbeeld.
   Antwoord Voor een tegenvoorbeeld moet $a[a[n]] \neq a[n]$ zijn. Laat $a[3] = 4$, $a[4] = 5$ en $a[5] = 1$.
3. Welke programma's S waarin de variabele z niet voorkomt voldoen aan

   $$\{x = z * z\}S\{x = z\}$$

   voor partiële correctheid?
   Antwoord: Al die programma's die *niet* termineren als de preconditie $x = z * z$ waar is en x ongelijk aan nul. Want als S termineert dan geldt $x = z$ na afloop met x (en dus ook z) groter dan nul. Aangezien z niet in S voorkomt kunnen we in deze terminerende berekening z vervangen door $-z$. Maar dan zou na afloop weer $x = z$ moeten zijn en dus x kleiner dan nul moeten zijn.
4. Specificeer een programma dat de sectie $a[1 : n]$ van array a van type **integer** $\rightarrow$ **integer** sorteert (let op: hier is ook voor nodig dat het resultaat een permutatie is van de de initiele array a).
   Antwoord

   $$\{\forall i : 1 \leq i \leq n : a[i] = b[i]\}$$
   $$S$$
   $$\{\exists c : perm(c[1 : n]) \wedge \forall i \in [1 : n] : a[i] = b[c[i]] \wedge (i < n \rightarrow a[i] \leq a[i + 1])\}$$

waar $perm(c[1:n])$ staat voor de assertie

$$\forall i, j \in [1:n] : c[i] \in [1:n] \wedge (i \neq j \rightarrow c[i] \neq c[j])$$

die uitdrukt dat de array c (van type **integer** $\rightarrow$ **integer**) een *bijectie* is op het interval $[1:n]$ (ga na).

5. Opgave 2.2(i) uit het boek en bereken
   (a) $(\exists z : z = x - 1)[x := z]$
   Antwoord $\exists y : y = z - 1$
   (b) $(\exists y : y = x - 1)[x := z]$
   Antwoord $\exists y : y = z - 1$

9-3 Hoofdstuk 3, sectie 3.4 (Proof Outlines en Partial Correctness). Zie ook slides 39 – 47.
Opgaven:

1. Opgaven 3.5, 3.8 en 3.9(ii) (uit het boek).
   Gegeven het het volgende programma S:

   $y := 0; u := 0; v := 1;$
   while $u + v \leq x$
   do $y := y + 1;$
   $u := u + v;$
   $v := v + 2$
   od

   Bewijs door middel van een proof-outline dat

   $$\{x \geq 0\}\ S\ \{y^2 \leq x < (y+1)^2\}$$

   Uitwerking

   $\{0 \leq x\}$
   $\{0 = 0^2 \wedge 0^2 \leq x \wedge 1 = 2 \times 0 + 1\}$
   $y := 0; u := 0; v := 1;$
   $\{u = y^2 \wedge y^2 \leq x \wedge v = 2 \times y + 1\}$
   while $u + v \leq x$
   do $\{u = y^2 \wedge y^2 \leq x \wedge v = 2 \times y + 1 \wedge u + v \leq x\}$
   $\{u + v = (y+1)^2 \wedge (y+1)^2 \leq x \wedge v + 2 = 2 \times (y+1) + 1\}$
   $y := y + 1;$
   $\{u + v = y^2 \wedge y^2 \leq x \wedge v + 2 = 2 \times y + 1\}$
   $u := u + v;$
   $\{u = y^2 \wedge y^2 \leq x \wedge v + 2 = 2 \times y + 1\}$
   $v := v + 2$

$\{u = y^2 \wedge y^2 \leq x \wedge v = 2 \times y + 1\}$
od
$\{u = y^2 \wedge y^2 \leq x \wedge v = 2 \times y + 1 \wedge x < u + v\}$
$\{y^2 \leq x \wedge x < (y+1)^2\}$

Merk op dat we gebruik maken van de formule

$$(y+1) = y^2 + 2 * y + 1$$

2. (Array right-shift) Bewijs door middel van een proof-outline dat

$$\{\forall n \in [1:k] : a[n] = b[n]\}$$
$$i := k-1; \textbf{while } 1 \leq i \textbf{ do } a[i+1] := a[i]; i := i-1 \textbf{ od}$$
$$\{\forall n \in [1:k-1] : a[n+1] = b[n]\}$$

Uitwerking

$\{\forall n \in [1:k] : a[n] = b[n]\}$
$\{\forall n \in [k:k-1] : a[n+1] = b[n] \wedge \forall n \in [1:k-1] : a[n] = b[n]\}$
$i := k-1;$
$\{\forall n \in [i+1, k-1] : a[n+1] = b[n] \wedge \forall n \in [1:i] : a[n] = b[n]\}$
while $1 \leq i$
do $\{\forall n \in [i+1, k-1] : a[n+1] = b[n] \wedge \forall n \in [1:i] : a[n] = b[n]\}$
$\{\forall n \in [i+1, k-1] : a[n+1] = b[n] \wedge a[i] = b[i] \wedge \forall n : n \in [1:i-1] : a[n] = b[n]\}$
$a[i+1] := a[i];$
$\{\forall n \in [i+1, k-1] : a[n+1] = b[n] \wedge a[i+1] = b[i] \wedge \forall n \in [1:i-1] : a[n] = b[n]\}$
$\{\forall n \in [i, k-1] : a[n+1] = b[n] \wedge \forall n \in [1:i-1] : a[n] = b[n]\}$
$i := i-1$
$\{\forall n \in [i+1, k-1] : a[n+1] = b[n] \wedge \forall n \in [1:i] : a[n] = b[n]\}$
od
$\{\forall n \in [i+1, k-1] : a[n+1] = b[n] \wedge \forall n \in [1:i] : a[n] = b[n] \wedge i < 1\}$
$\{\forall n \in [1:k-1] : a[n+1] = b[n]\}$

Ga zelf de logische implicaties na en bewijs (m.b.v. het assignment axioma) dat

$$\{\forall n : n \in [i+1:k-1] : a[n+1] = b[n] \wedge a[i] = b[i] \wedge \forall n \in [1:i-1] : a[n] = b[n]\}$$
$$a[i+1] := a[i];$$
$$\{\forall n \in [i+1:k-1] : a[n+1] = b[n] \wedge a[i+1] = b[i] \wedge \forall n \in [1:i-1] : a[n] = b[n]\}$$

23-3 Hoofdstuk 3, sectie 3.11 (Case Study: Minimum-Sum Section Problem). Zie ook slides 48 – 56.
Opgaven: Zie slides 57 – 63.

13-4 Hoofdstuk 3, sectie 3.4 (Total Correctness) en sectie 3.8 (Auxiliary Axioms and Rules). Zie ook slides 64 – 72.