SEMINAR 6 - VVSS, 2019-2020, 11.05.2020

A. Dezvoltarea algoritmilor corecți din specificații

- 4 reguli
  - regula atribuirii (R1)
  - regula compunerii secvențiale (R2)
  - regula alternanței (R3)
  - regula iterației (R4)

1. Împărțire întreagă

$\varphi(X): (x \geq 0) \wedge (y > 0)$ $\qquad X = (x, y)$

$\psi(X, Z): (x = q * y + r) \wedge (0 \leq r < y)$ $\qquad Z = (q, r)$

$A_0: \boxed{[\varphi, \psi]} \overset{R2}{\leq} A_1$

program abstract, care trebuie rafinat / descompus a.î. să se ajungă la un program format doar din structurile: secvențiale, alternativă și iterativă

- putem identifica un predicat η, $\eta \Rightarrow \psi$ → se aplică R2

$\eta: (x = q * y + r) \wedge (0 \leq r)$

$A_1: \boxed{\begin{matrix} [\varphi, \eta] \\ [\eta, \psi] \end{matrix}} \overset{R1}{\leq} A_2$

- pentru $q = 0$, $r = x \Rightarrow \eta$ - adevărat → se aplică R1

$A_2: \boxed{\begin{matrix} (q, r) = (0, x) \\ [\eta, \psi] \end{matrix}} \overset{R4}{\leq} A_3$

$\eta : (x = q * y + r) \wedge (0 \le r)$

$\psi : (x = q * y + r) \wedge (0 \le r < y)$

$[\eta, \psi] = [\eta, \eta \wedge (r < y)] \rightarrow$ se poate aplica R4

$\neg cond = \neg(r < y) = r \geq y$

$$A3 : \boxed{\begin{array}{l} (q, r) = (0, x) \\ \text{DO} \quad r \geq y \rightarrow [\eta \wedge (r \geq y), \eta \wedge TC] \\ \text{OD} \end{array}} \overset{R1}{\le} A4$$

- x, y - nu își modifică valoarea
- q, r - se modifică

$\Big| \Rightarrow$ pentru ca instr. DO

să se termine, condiția $r \geq y$ trebuie să devină falsă $\Rightarrow$

$\Rightarrow r < y$ - adică r să scadă (y - nu se modifică)

↳ este inițial x

- r se poate diminua la fiecare iterație cu y $\Rightarrow$

$\Rightarrow r \Leftarrow r - y$

dar, η - invariant în $[\underline{\eta} \wedge (r \geq y), \underline{\eta} \wedge TC]$ (pre: , post:) $\Big| \Rightarrow$

$\Rightarrow$ q - crește

pre: $x = q * y + r$ post: $x = q * y + r$

$= q * y + \underbrace{r - y}_{} + y \quad r = r - y$

$q \Leftarrow q + 1 \quad \longleftarrow = (q + 1) * y + \underline{r - y}$

$\Rightarrow$ se aplică R1

A4:
$$\boxed{\begin{array}{l} (q, r) = (0, x) \\ \text{DO} \\ \quad r \geq y \rightarrow (q, r) = (q+1, r-y) \\ \text{OD} \end{array}}$$

---

2. Rădăcina pătrată întreagă

$\varphi(X): n > 1$ $\qquad X = (n)$

$\psi(X, Z): r^2 \leq n < (r+1)^2$ $\qquad Z = (r)$

A0: $\boxed{[\varphi, \psi]} \overset{R2}{\leq} A_1$

- introducem notația $q = r+1 \Rightarrow$ construim ψ_0

$\psi_0: \underbrace{(r^2 \leq n < q^2)}_{\eta} \wedge (q = r+1)$

$\Rightarrow \psi_0 \equiv \psi$

- identificăm η, o parte din ψ_0, a.î. $\eta \Rightarrow \psi_0 \equiv \psi$ $\rightarrow$ se aplică R2

$\eta: r^2 \leq n < q^2$

A1: $\boxed{\begin{array}{l} [\varphi, \eta] \\ [\eta, \psi_0] \\ \{\psi\} \end{array}}$ $\overset{R1}{\leq} A_2$

- η este adevărat pentru $r = 0$, $q = n+1$ $\rightarrow$ se aplică R1

A2: $\boxed{\begin{array}{l} (r, q) = (0, n+1) \\ [\eta, \psi_0] \\ \{\psi\} \end{array}}$ $= A_3$

$\eta: r^2 \le n < q^2$

$\varphi_0: (r^2 \le n < q^2) \wedge (q = r+1)$

$\varphi_0: \eta \wedge (q = r+1)$

A3: $\boxed{\begin{array}{l} (r,q) = (0, n+1) \\ [r, \eta \wedge (q = r+1)] \end{array}}$ R4 $\le$ A4

se aplică R4

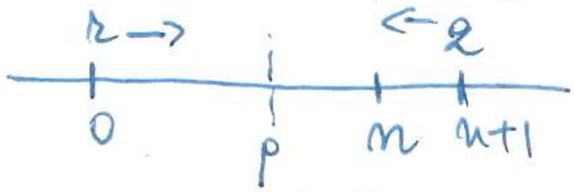
$\neg cond = \neg(q = r+1) = (q \neq r+1)$

A4: $\boxed{\begin{array}{l} (r,q) = (0, n+1) \\ \text{DO} \\ q \neq r+1 \rightarrow [\eta \wedge (q \neq r+1), \eta \wedge \neg C] \\ \text{OD} \end{array}}$ R3, R1 $\le$ A5

- q = valoare foarte mare => trebuie să scadă
- r = valoare foarte mică => trebuie să crească | =>

=> r trebuie să crească până când $q = r+1$, $q - r = 1$, adică instr. DO se termină

- η - adevărat în pre și post => fie r (crește), fie q (scade) se modifică



$p = \frac{r+q}{2}$

$r < p < q \mid ()^2$

$r^2 < p^2 < q^2$

- comparăm p^2 cu n

se aplică R3: $p^2 \le n \Rightarrow r = p$; $p^2 > n \Rightarrow q = p$ | - se aplică R1

A5:

$(r,q) = (0, n+1)$

DO
$q \neq r+1 \rightarrow p = \frac{r+q}{2}$
IF $p^2 \leq n \rightarrow r = p$
$\square$ $p^2 > n \rightarrow q = p$
FI
OD

B. Limbajul JML (Java Modeling Language)

- vezi prezentare
  - ESC2Java - verificare statică
  - jmlc + jmlrac - verificare dinamică
- openjml.org

C. Activități studiate în cadrul VVSS

- vezi diagrama