Dokazovanje pravilnosti

Specifikacija: Kaj bi sploh radi imeli?
Opis, kaj naj koda počne

Implementacija: Koda, ki ustreza specifikaciji

Hoarova logika

Hoarove trojice

$\{P\}\ c\ \{Q\}$

predpogoj
precondition

program,
ukaz

končni pogoj
postcondition

$[P]\ c\ [Q]$

DELNA PRAVILNOST:
če velja P in se c ustavi,
potem velja Q

POPOLNA PRAVILNOST:
če velja P, potem
se c ustavi in velja Q

Primer: Zamenjaj vrednosti spremenljivk x in y:

$$\{x=m \wedge y=n\}\ c\ \{x=n \wedge y=m\}$$

① $t := x;\ x := y;\ y := t$ ✓ zadošča tudi $[x=m \wedge y=n]\ c\ [x=n \wedge y=m]$

② $x := y;\ y := x$ ✗

③ while true do skip done ✓ vendar ne zadošča
$[x=m \wedge y=n]\ c\ [x=n \wedge y=m]$

④ $x := 1;\ y := 1;\ m := 1;\ n := 1$ ✓ ni bilo mišljeno, da spreminjamo m in n.

Da se znebimo ④, zahtevamo, da m in n ne smemo spreminjati:
m in n sta duhova (ghost variable)

Popravimo zahtevo:

$$\{x=m \wedge y=n\}\ c\ \{x=n \wedge y=m\}$$

in m, n duhova

Primer: $\{\mathrm{true}\}\ c\ \{x \leq y\}$

Rešitev: $x := 0;\ y := 0$ ✓

"Uredi x in y po velikosti." m, n duhova

čudno: $\{x=m \wedge y=n\}\ c\ \{x \leq y \wedge (x=m \vee x=n) \wedge (y=m \vee y=n)\}$

$x := y$ ✓ ni mišljeno

2. poskus: $\{x=m \wedge y=n\}\ c\ \{(m<n \Rightarrow x=m \wedge y=n) \wedge (m \geq n \Rightarrow x=n \wedge y=m)\}$

$$\begin{array}{ccc}
(\underset{\perp}{m<n} \Rightarrow P) & \vee & (\underset{\perp}{m \geq n} \Rightarrow Q) \\
\perp \Rightarrow P & & \perp \Rightarrow Q \\
\top & & \top
\end{array}$$

$$\{x=m \wedge y=n\}\ c\ \{x=\min(m,n) \wedge y=\max(m,n)\}$$

Pišemo:

$$\begin{array}{c}
\{P\} \\
c \\
\{Q\}
\end{array}
\qquad\qquad
\begin{array}{c}
\{P_1\} \\
c_1; \\
\{P_2\} \\
c_2; \\
\{P_3\} \\
\vdots
\end{array}$$

$\{x \leq 10\}$

$x := x + 2$

$\{x \leq 12\}$ $\cancel{\{x \leq 100\}}$

Strongest postcondition
najmočnejši končni pogoj

$\cancel{\{x \leq 2\}}$ $\{x \leq 6\}$

$x := 2 * x$

$\{x \leq 12\}$

Weakest precondition,
najšibkejši predpogoj

Pravila sklepanja

$$\frac{P' \Rightarrow P \qquad \{P\}\ c\ \{Q\} \qquad Q \Rightarrow Q'}{\{P'\}\ c\ \{Q'\}}\ \text{I}$$

Uporaba:

$\{P_1\}$
C_1
$\{P_2\}$
preveri $P_2 \Rightarrow Q_3$
$\{Q_3\}$
C_2
$\{Q_2\}$
C_3
$\{Q_1\}$

$$\frac{\{P_1\}\ c\ \{Q_1\} \qquad \{P_2\}\ c\ \{Q_2\}}{\{P_1 \wedge P_2\}\ c\ \{Q_1 \wedge Q_2\}}\ \text{II}$$

Kaj pa $\{P\}\ c\ \{Q_1 \wedge Q_2\}$

$$\frac{P \Rightarrow P \wedge P \qquad \dfrac{\{P\}\ c\ \{Q_1\} \qquad \{P\}\ c\ \{Q_2\}}{\{P \wedge P\}\ c\ \{Q_1 \wedge Q_2\}}\ \text{II} \qquad Q_1 \wedge Q_2 \Rightarrow Q_1 \wedge Q_2}{\{P\}\ c\ \{Q_1 \wedge Q_2\}}\ \text{I}$$

$\{x \leq 7\}$
$y := x+3$
$\{x \leq 7\}$

$\{P\}$ → spremenljivke $x_1, \dots, x_n$
C → ne spreminja spremenljivk iz P
$\{P\}$

$\{x \leq 7\}$
$x := x-3$ ✓
$\{x \leq 7\}$

$FV(\dots)$ vse spremenljivke

$FA(\dots)$ vse, ki se pojavijo levo od $:=$

$FA(\text{ if } \mathit{false} \text{ then } \underline{x}:=5 \text{ else } \underline{y}:=7\,) = \{x, y\}$

Pogojni stavek:

$$\frac{\{P \land b\}\ c_1\ \{Q\} \qquad \{P \land \neg b\}\ c_2\ \{Q\}}{\{P\}\ \text{if } b \text{ then } c_1 \text{ else } c_2\ \{Q\}}$$

```
{P}
if b then
    {P ∧ b}
    c_1
    {Q}
else
    {P ∧ ¬b}
    c_2
    {Q}
{Q}
```

Pravilo za ;

$$\frac{\{P\}\ c_1\ \{Q\} \qquad \{Q\}\ c_2\ \{R\}}{\{P\}\ c_1;\ c_2\ \{R\}}$$

```
{P}
c_1 ;
{Q}
c_2
{R}
```

Zanka while:

$$\frac{\{P \land b\}\ c\ \{P\}}{\{P\}\ \text{while } b \text{ do } c \text{ done } \{P \land \neg b\}}$$

```
{$P$}
while $b$ do
    {$P \land b$}
    $c$
    {$P$}
done
{$P \land \neg b$}
```

P se imenuje invarianta zanke

```
{$P$}
 $\Downarrow$        uporabimo možgane
{$P'$}
while $b$ do
    {$P' \land b$}
    $c$
    {$P'$}
done
{$P' \land \neg b$}
 $\Downarrow$
{$Q$}
```

P' invarianta

```
{$P[x \mapsto e]$}
$x := e$
{$P$}
```

Zapis: $P[x \mapsto e]$

"V formuli P zamenjamo x z e"

substitucija ali zamenjava

$$(x \leq 8 \lor y + x = 3)\,[x \mapsto 3 + z]$$

$$\Downarrow$$

$$(3 + z \leq 8 \lor y + (3 + z) = 3)$$

```
{$7^3 - 5 \cdot 7 \leq 200$}
$x := 7$
{$x^3 - 5x \leq 200$}
```

$$\frac{}{\{P[x \mapsto e]\}\ x := e\ \{P\}}$$

```
s := 0;
i := 0;
while i < 100 do
    s := s + i;
    i := i + 1
done
```

64-bit

"e se zmanjša"

$$\frac{[P \wedge b \wedge e = z]\ c\ [P \wedge e < z]}{[P]\ \text{while } b \text{ do } c \text{ done } [P \wedge \neg b]}$$

z duh

e je naravno število, ali količina, ki se ne more v nedogled zmanjševati, npr.:

$$\underbrace{e \in \mathbb{Z} \wedge e > -17}_{e+17}$$

$$x \le y \Rightarrow \underbrace{x \le \frac{x+y}{2}} \le y$$

$$x \le y \Rightarrow x \le \frac{x+y}{2}$$

Preverimo:

$$x = \frac{x+x}{2} \overset{\text{ker } x \le y}{\le} \frac{x+y}{2}$$

Obi-wan error

off-by-one error

```
while i < b do
    p := p * a;
    i := i + 1
done
```

i	p	$p = a^i$
0	1	✓
1	a	✓
2	a^2	✓
3	a^3	✓