Pravilnost programov

Specifikacija: opis/zahteva, kaj naj bi program delal

Implementacija: koda, ki jo napišemo, da bi zadostili specifikaciji

Hoarove trojke:

$$\underset{\substack{\uparrow\\ \text{precondition}\\ \text{predpogoj}}}{\{P\}}\quad \overset{\substack{\text{koda}\\ \swarrow}}{C}\quad \underset{\substack{\uparrow\\ \text{postcondition}\\ \text{postpogoj} \leftarrow \text{ChatGPT}\\ \text{končni pogoj}}}{\{Q\}}$$

P, Q logični formuli, ki govorita o vrednostih spremenljivk

Ko preverjamo pravilnost, nalogo razdelimo:

1. Ali se program ustavi?
2. Če predpostavimo, da se program ustavi, ali zadošča dani specifikaciji?

Delna pravilnost:

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

predpostavka

Če velja P in če se c ustavi, po izvedbi c velja Q.

Popolna pravilnost:

$[P]\ c\ [Q]$

sklep

Če velja P, potem se c ustavi in po izvedbi c velja Q.

Pišemo:

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

$\{P\}$
c_1
$\{Q_1\}$
c_2
$\{Q_2\}$
$\vdots$
c_n
$\{Q\}$

Primer:

① $\{x = m \wedge y = n\}$

```
Z := m;
x := y;
y := z
```

$\{x = n \wedge y = m\}$ ✓

② $\{x = m \wedge y = n\}$

```
Z := x;
x := y;
y := z
```

$\{x = n \wedge y = m\}$ ✓

③ $\{x = m \wedge y = n\}$

```
x := y;
y := x
```

$\{x = n \wedge y = m\}$ ✗

④ $\{x = m \wedge y = n\}$

```
x := 5;
n := 5;
y := 42;
m := 42
```

$\{x = n \wedge y = m\}$ ✓ ?!

Uporabimo duhove (ghost variable):

m in n se v kodi ne smeta pojaviti

$\{x = m \land y = n\}$ m, n duhova

②

```
z := x;
x := y;
y := z
```

$\{x = n \land y = m\}$

Primer:

$\{n \geq 1\}$

```
s := n·(n+1)/2;
x := 42
```

$\{s = \underbrace{1+2+3+\cdots+n}\}$

Sestavi program, ki sešteje prvih n naravnih števil in vsoto shrani v s.

Pravila sklepanja

hipoteze

$$\frac{H_1 \quad H_2 \quad \cdots \quad H_n}{S}$$

sklep

$$\frac{P' \Rightarrow P \qquad \{P\}\, c\, \{Q\} \qquad Q \Rightarrow Q'}{\{P'\}\, c\, \{Q'\}}$$

pišemo:

$\{P'\}$ ↓ preveri $P' \Rightarrow P$

$\{P\}$

c

$\{Q\}$ ↓ preveri $Q \Rightarrow Q'$

$\{Q'\}$

```
{ $x = 7 \lor x < 5$ }    ✓
y := y+1;                 ker x ne omenimo
z := 13
{ $x = 7 \lor x < 5$ }
```

```
{ $x = 7 \lor x < 5$ }    ✓
y := y+1;                 ker x ne spreminjamo
z := x + 5
{ $x = 7 \lor x < 5$ }
```

```
{ $x = 7 \lor x < 5$ }    ?
y := y+1;
x := x+y
{ $x = 7 \lor x < 5$ }
```

Primer (Java):

```
void f(HashMap x, HashMap y) {

    x.add(7, "foo");
```

Ali se je y spremenil?

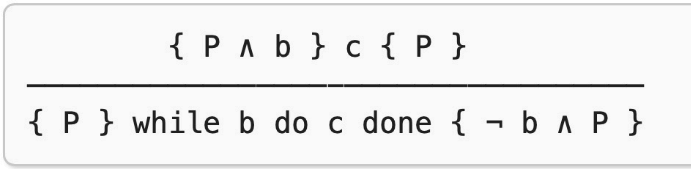
Morda, x in y bi lahko bila isti objekt.

aliasing / prekrivanje

Pogojni stavek:

```
{ true }
if $x < 5$ then
    { $true \land x < 5$ }
    y := x
    { $y < 5$ }
else
    { $x \geq 5$ }
    y := 0
    { $y < 5$ }
end
{ $y < 5$ }
```

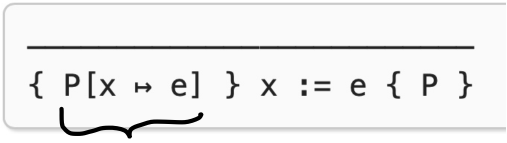
Zanka while



```
         { P ∧ b } c { P }
────────────────────────────────────
{ P } while b do c done { ¬ b ∧ P }
```

P invarianta zanke

Prirejanje



```
─────────────────────────────
{ P[x ↦ e] } x := e { P }
```

$\underbrace{P[x \mapsto e]}$ v P zamenjaj x z e.
(Substitucija / zamenjava)

Primer:

$\{7 < 10\}$

$x := 7$

$\{x < 10\}$

P je "x < 10"

e je 7

$P[x \mapsto 7]$ dobim $7 < 10$

$$\frac{}{\{P(e)\}\ \ x := e\ \ \{P(x)\}}$$

Primer:

$\{i < n\}$ ⇓ ✓

$\{i+1 \le n\}$

$i := i+1$

$\{i \le n\}$

Primer:

$\{i < n\}$ ⇓ ✓

$\{(i+1)-1 < n\}$ ← $P(i+1)$

$i := i+1$

$\{\underbrace{i - 1 < n}_{P(i)}\}$

.... $P(k) = k-1 < n$

$\{2+2=4\}$

$x = 7$

$\{2+2=4\}$

Primer butaste specif.

p	q	$p \Rightarrow q$
⊥	⊥	⊤
⊥	⊤	⊤
⊤	⊥	⊥
⊤	⊤	⊤

```
i = 0
p = 1
while i < b do
   p := p * a ;
   i := i + 1
done
```

i	p	$p = a^i \wedge$
0	1	
1	a	
2	a^2	
3	a^3	
⋮	⋮	