Dokazovanje pravilnosti programov

Specifikacija: opis, kaj naj program poine → Kaj bi radi?

Implementacija: je konkreten program, ki poine, kar zahteva specifikacija. → Kako se to naredi?

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

Hoarova logika

Hoarova trojica

$\{P\}\ C\ \{Q\}$

- P: predpogoj (precondition)
- Q: konični pogoj (postcondition)

P, Q logični formuli
C ukaz (koda)

Delna pravilnost $\{P\}\ C\ \{Q\}$ ← ne zagotavlja, da se C ustavi!

Če velja P in če se C ustavi, bo veljal Q.

Popolna pravilnost $[P]\ C\ [Q]$ ← zagotavlja, da se C ustavi.

Če velja P, potem se C ustavi in veljal bo Q.

Primer:

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

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

$\{x = m \wedge y = n\}$
$t := x;\ x := y;\ y := t$
$\{x = n \wedge y = m\}$ ✓

✗
$[x = m \wedge y = n]$
while true do skip done
$[x = n \wedge y = m]$

$\{x = m \wedge y = n\}$
while true do skip done
$\{x = n \wedge y = m\}$ ✓

$\{x = m \wedge y = n\}$
$x := 0;\ y := 0;\ m := 0;\ n := 0;$ ✓?!
$\{x = n \wedge y = m\}$

$\{x = m \wedge y = n\}$
$t := x;\ x := y;\ y := t$
$\{x = n \wedge y = m\}$

m in n naj bosta duhova (ghost variable)

To pomeni, da se m in n ne smeta uporabljati v programu.

$\{x = m \wedge y = n\}$
$t := x;\ x := y;\ y := t;$ skip; skip; skip
$\{x = n \wedge y = m\}$

$\{x = m \wedge y = n\}$
$x := x + y;$
$y := x - y;$
$x := x - y;$
$\{x = n \wedge y = m\}$ ✓

Primer: $\{true\}$ ← ni nobene predpostavke

C

$\{x \leq y\}$

$\{true\}$
$x := 0;$
$y := 10;$
$\{x \leq y\}$ ✓

Primer: $\{x = m \wedge y = n\}$

if $x < y$ then
  Skip
else $t := x; x := y; y := t$
end

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

Pravila Hoarove logike

$$\frac{A_1 \quad A_2 \quad \ldots \quad A_n}{B}$$

$A_1, A_2, \ldots, A_n$ ← predpostavke

B ← sklep

"Če veljajo $A_1, \ldots, A_n$, potem velja tudi B."

$P \equiv \forall x . \exists z . y + z = x$

$FV(P) = \{y\}$

x in z sta vezani spremenljivki
(x in z sta lokalni spremenljivki)

$Q \Rightarrow false \quad \neg Q$

Primeri

$\{ true \}\ c\ \{ true \}$ velja vedno

$\{ true \}\ c\ \{ false \}$ c se ne ustavi

$true \wedge P \Leftrightarrow P$

$true \wedge$ "c se ustavi" $\Rightarrow false$

"c se ustavi" $\Rightarrow false$

$\neg$ "c se ustavi"

$\{ false \}\ c\ \{ true \}$ velja vedno

$\{ false \}\ c\ \{ false \}$ velja vedno

$[true]\ c\ [true]$ c se ustavi

$[true]\ c\ [false]$ nikoli ne velja

$[false]\ c\ [true]$ vedno velja

$[false]\ c\ [false]$ vedno velja

Primer:

```
{ x ≤ y }
s := (x + y) / 2
{ x ≤ s ≤ y }
```

$\{ x \le y \}$

$\{ x \le (x+y)/2 \le y \}$ ⟸ ?

$s := (x+y)/2;$

$\{ x \le s \le y \}$

Predpostavimo $x \le y$

Preverimo $x \le (x+y)/2$ /·2

$\Leftrightarrow\ 2x \le x+y$ /−x

$\Leftrightarrow\ x \le y$ ✓

Preverimo $(x+y)/2 \le y$ /·2

$x+y \le 2y$ /−y

$x \le y$ ✓

Primer:

```
{ x ≤ 7 }
x := x + 3
{ x ≤ 10 }
```

$\{ x \le 7 \}$

$\{ x+3 \le 10 \}$ ⟸ ✓

$x := x+3;$

$\{ x \le 10 \}$

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

P je $x \le 10$

e je $x+3$

$P[x \mapsto e]$ je $x+3 \le 10$

$\{ x \leq 7 \}$

$\{ (x+3) - 3 \leq 7 \}$

$x := x + 3;$

$\{ x - 3 \leq 7 \}$

$\{ \text{true} \}$

$\{ 5 = 5 \}$

$x := 5;$

$\{ x = 5 \}$

P ... $x = 5$

e ... 5

$P[x \mapsto e]$ $\quad 5 = 5$

```
while i < b do
   { i < b, ?Q }
   p := p * a ;
   i := i + 1
   { ?Q }
done
```

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