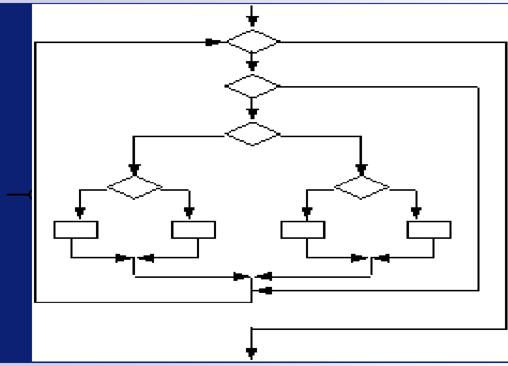
Spring 2008



Program correctness

Total correctness

Marcello Bonsangue

Leiden Institute of Advanced Computer Science
Research & Education

Total correctness

- Hoare triple for total correctness

precondition

postcondition

$$\models_{tot} \{\phi\}\ c\ \{\psi\}$$

command

If the command c is executed in a state that satisfies ϕ then c is guaranteed to terminate and the resulting state will satisfy ψ

program termination is required

Example

- $\models_{tot}$ { $y \leq x$ } z := x; z := z +1 { $y < z$ } is valid
- $\models_{tot}$ { true } <u>while</u> true <u>do</u> skip <u>od</u> { false } is not valid
- $\models_{tot}$ { false } <u>while</u> true <u>do</u> skip <u>od</u> { true } is valid
- Let Fact = y := 1; z := 0;

```
while z ≠ x do
        z := z + 1;
        y := y*z
od
```

Is $\models_{tot}$ { $x \geq 0$ } Fact { $y = x!$ } valid?

Total correctness

- Total correctness: $I \vDash_{tot} \{\phi\}\ c\ \{\psi\}$

  $\forall\sigma.\ \sigma, I \vDash \phi \Rightarrow \exists\ \sigma'.(<c, \sigma> \rightarrow \sigma' \text{ and } \sigma', I \vDash \psi)$

  where ϕ and ψ are assertions and c is a command

Validity

- To give an absolute meaning to

$$\{i < x\}\ x := x+3\ \{i < x\}$$

  we have to quantify over all interpretations I

- Total correctness:

$$\models_{tot} \{\phi\}\ c\ \{\psi\} \quad \equiv \quad \forall I.\ I \models_{tot} \{\phi\}\ c\ \{\psi\}$$

Towards a calculus

- Partial correctness does not tell anything about termination

- Only while b do c od introduces the possibility of non-termination

  ⇓

  a proof calculus for total correctness is the same as that for partial correctness except for the while-rule

Intuition

- To prove total correctness we need
  - a proof of partial correctness
  - a proof that the while statement terminates

- Termination can be proved by finding an integer expression E (the variant) that
  - is always non-negative
  - decreases every time we execute the body of the while statement

Proof rules total and partial correctness (I)

- $\{\phi\}$ skip $\{\phi\}$ skip
- $\{\phi[a/x] \wedge \mathrm{def}(a)\}$ x := a $\{\phi\}$ ass
- $$\frac{\{\phi\}\ c_1\ \{\psi\} \qquad \{\psi\}\ c_2\ \{\varphi\}}{\{\phi\}\ c_1;\ c_2\ \{\varphi\}}$$ seq

Proof rules
total and partial correctness (II)

- $$\frac{\{\phi \wedge b\}\ c_1\ \{\psi\} \qquad \{\phi \wedge \neg b\}\ c_2\ \{\psi\}}{\{\phi\}\ \underline{\text{if}}\ b\ \underline{\text{then}}\ c_1\ \underline{\text{else}}\ c_2\ \underline{\text{fi}}\ \{\psi\}}\quad \text{if}$$

- $$\frac{\vdash \phi \Rightarrow \phi' \qquad \{\phi'\}\ c\ \{\psi'\} \qquad \vdash \psi' \Rightarrow \psi}{\{\phi\}\ c\ \{\psi\}}\quad \text{cons}$$

Proof rule total correctness (III)

$$\{\phi \wedge b \wedge 0 \leq E = E_0 \}\ c\ \{\phi \wedge 0 \leq E < E_0 \}$$

$$\{\phi \wedge 0 \leq E \}\ \underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}\ \{\phi \wedge \neg b\}$$

where E_0 is a logical variable for retaining the initial value of E

Finding E cannot be mechanized !!!

Proof outline

- Proof outline for total correctness are similar to those for partial correctness except for
  - the precondition of the while which now writes

$$\{ \phi \wedge 0 \leq E \}$$

  - the body of the while which now writes

$$\{ \phi \wedge b \wedge 0 \leq E = E_0 \} \; c \; \{ \phi \wedge 0 \leq E < E_0 \}$$

An example

DIV ≡

```
q := 0;
r := x;
while r ≥ y do
        r := r-y;
        q := q+1
od
```

We wish to prove

$\{x \geq 0 \wedge y > 0\}$ DIV $\{q*y+r=x \wedge 0 \leq r < y\}$

An example (II)

$\{x \geq 0 \wedge y > 0\}$	
$\{0*y+x=x \wedge 0 \leq x\}$	implied
q := 0;	
$\{q*y+x=x \wedge 0 \leq x\}$	ass.
r := x;	
$\{ I \}$	ass.
while $r \geq y$ do	
$\{ I \wedge r \geq y \}$	Inv $\wedge$ guard
$\{ (q+1)*y+ r-y = x \wedge 0 \leq r-y \}$	implied
r := r-y;	
$\{ (q+1)*y+r=x \wedge 0 \leq r \}$	ass.
q := q+1	
$\{ I \}$	ass.
od	
$\{ I \wedge r<y \}$	while
$\{ q*y+r=x \wedge 0 \leq r<y \}$	implied

where $I \equiv q*y+r=x \wedge 0 \leq r$ is the invariant

An example (III)

$\{x \geq 0 \wedge y > 0\}$	
$\{0*y+x=x \wedge 0 \leq x\}$	implied
q := 0;	
$\{q*y+x=x \wedge 0 \leq x\}$	ass.
r := x;	
$\{I \wedge 0 \leq r\}$	ass.
<u>while</u> $r \geq y$ <u>do</u>	
$\{I \wedge r \geq y \wedge 0 \leq r = z\}$	Inv $\wedge$ guard
$\{(q+1)*y + r-y = x \wedge 0 \leq r-y < z\}$	implied?????
r := r-y;	
$\{(q+1)*y+r=x \wedge 0 \leq r < z\}$	ass.
q := q+1	
$\{I \wedge 0 \leq r < z\}$	ass.
<u>od</u>	
$\{I \wedge r < y\}$	while
$\{q*y+r=x \wedge 0 \leq r < y\}$	implied

where $I \equiv q*y+r=x \wedge 0 \leq r$ is the invariant and r is the variant

An example (IV)

$\{x \geq 0 \wedge y > 0\}$	
$\{0*y+x=x \wedge 0 \leq x \wedge y > 0\}$	implied
$q := 0;$	
$\{q*y+x=x \wedge 0 \leq x \wedge y > 0\}$	ass.
$r := x;$	
$\{I \wedge 0 \leq r\}$	ass.
<u>while</u> $r \geq y$ <u>do</u>	
$\{I \wedge r \geq y \wedge 0 \leq r = z\}$	Inv ∧ guard
$\{(q+1)*y + r-y = x \wedge y > 0 \wedge 0 \leq r-y < z\}$	implied
$r := r-y;$	
$\{(q+1)*y+r=x \wedge y > 0 \wedge 0 \leq r < z\}$	ass.
$q := q+1$	
$\{I \wedge 0 \leq r < z\}$	ass.
<u>od</u>	
$\{I \wedge r < y\}$	while
$\{q*y+r=x \wedge 0 \leq r < y\}$	implied

where $I \equiv q*y+r=x \wedge 0 \leq r \wedge y > 0$ is the invariant and r is the variant