Software Validation and Verification
Section II: Model Checking

Topic 5. Proving Program Correctness

Pedro Cabalar

Department of Computer Science and IT
University of Corunna, SPAIN
cabalar@udc.es

8 de octubre de 2019

Hoare triple

- Hoare triple:

  $\{Q\}\ S\ \{R\}$ = If precondition Q initially true, then program S terminates satisfying postcondition R.

- A program is annotated when all its sentences (compound or atomic) are guarded by a pre and a postcondition.
- Example

$$
\begin{array}{l}
\{x = X \wedge y = Y\} \\
t = x; \\
\{t = X \wedge x = X \wedge y = Y\} \\
x = y; \\
\{t = X \wedge x = Y \wedge y = Y\} \\
y = t \\
\{y = X \wedge x = Y\}
\end{array}
$$

- Two adjacent assertions $\{P\}\{Q\}$ just mean $P \Rightarrow Q$

Weakest precondition *wp*

Definición 1 (Weakest Precondition *wp*)

$wp(S, R)$ represents all the initial states for which S terminates satisfying R

- Example: $wp(\text{“}i = i + 1\text{”}, i \leq 1) =$

$$\{\{(i, 0)\}, \{(i, -1)\}, \{(i, -2)\}, \dots\}$$

- We can represent $wp(S, R)$ using a formula. In our example:

$$wp(\text{“}i = i + 1\text{”}, i \leq 1) = (i \leq 0)$$

Weakest precondition *wp*

- *wp* is the weakest formula. For instance, $i \leq -10$ works as a precondition, but is not the weakest condition: ($i \leq -10 \Rightarrow i \leq 0$).
- Example: assume that S is

$$
\begin{array}{l}
\textbf{if} \\
:: \quad x \geq y \quad \rightarrow \quad z = x \\
:: \quad x < y \quad \rightarrow \quad z = y \\
\textbf{fi}
\end{array}
$$

$$
\begin{aligned}
wp(S, z = max(x, y)) &= \texttt{true} \\
wp(S, z = y) &= x \leq y \\
wp(S, z = y - 1) &= \texttt{false} \\
wp(S, z = y + 1) &= (x = y + 1)
\end{aligned}
$$

Weakest precondition wp

- Which is the meaning of $wp(S, \texttt{true})$?

  It represents the initial conditions that guarantee program termination. Example:

  $$wp(\text{“}\textbf{do} :: \texttt{true} \rightarrow \texttt{skip}\ \textbf{od}\text{”}, \texttt{true}) = \texttt{false}$$

- **Important**: $\{Q\}\ S\ \{R\}$ holds if and only if $Q \Rightarrow wp(S, R)$ is a tautology.
- This is called the total correction of S with respect to Q and R.

Partial correction

Definición 2 (Partial correction)

The notation $Q\{S\}R$ stands by the **partial correction** *of S with respect to Q y R and it means that, if S starts in a state where Q and terminates, then it satisfies R in the final state.*

- Note that it does not guarantee termination. $Q\{S\}R$ is trivially true if S does not terminate!!!
  Example:

  $$\mathtt{true}\ \{\mathbf{do} :: \mathtt{true} \rightarrow \mathtt{skip}\ \mathbf{od}\}\mathtt{true}$$

  is a tautology.

Properties of *wp*

- Properties:
  1. $wp(S, \texttt{false}) = \texttt{false}$
  2. $wp(S, \alpha) \wedge wp(S, \beta) = wp(S, \alpha \wedge \beta)$
  3. If $\alpha \Rightarrow \beta$ then $wp(S, \alpha) \Rightarrow wp(S, \beta)$
  4. $wp(S, \alpha) \vee wp(S, \beta) \Rightarrow wp(S, \alpha \vee \beta)$
  5. $wp(S, \alpha) \vee wp(S, \beta) \Leftarrow wp(S, \alpha \vee \beta)$,
     if S is deterministic.

Axiomatic semantics

- The meaning of a sentence is characterized by its *wp*

$$
\begin{aligned}
wp(\text{“}skip\text{”}, R) &= R \\
wp(\text{“}abort\text{”}, R) &= \texttt{false} \\
wp(\text{“}S1; S2\text{”}, R) &= wp(S1, wp(S2, R))
\end{aligned}
$$

- Examples:

$$
\begin{aligned}
wp(\text{“}skip; skip\text{”}, R) &= R \\
wp(\text{“}S1; (S2; S3)\text{”}, R) &= wp(\text{“}(S1; S2); S3\text{”}, R) \\
wp(\text{“}S; abort\text{”}, R) &= F
\end{aligned}
$$

Assignments

- Assignment "$x = e$".
  Variable identifier x gets the value of expression e.

$$wp(\text{“}x := e\text{”}, R) = domain(e)\ \textbf{cand}\ R^x_e$$

  where R^x_e is the textual substitution of x by e in R.
  where $domain(e)$ checks that e is evaluable.
- Examples:

$$\begin{aligned}
wp(\text{“}x = 5\text{”}, x = 5) &= \texttt{true} \\
wp(\text{“}x = 5\text{”}, x \neq 5) &= \texttt{false} \\
wp(\text{“}x = x + 1\text{”}, x = 5) &= (x = 4) \\
wp(\text{“}x = x + 1\text{”}, x < 0) &= (x < -1) \\
wp(\text{“}x = a/b\text{”}, x \neq 1) &= (b \neq 0)\ \textbf{cand}\ a/b \neq 1 \\
wp(\text{“}x = b[i]\text{”}, x = b[i]) &= inrange(i, b)\ \textbf{cand}\ \texttt{true}
\end{aligned}$$

Conditional instructions

- *wp* for an **IF** is defined as:

$$\begin{aligned} wp(\mathbf{IF}, R) &= domain(BB) \wedge BB \wedge \\ &\quad \wedge (B_1 \Rightarrow wp(S_1, R)) \wedge \ldots \\ &\quad \wedge (B_n \Rightarrow wp(S_n, R)) \end{aligned}$$

- Proving $Q \Rightarrow wp(\mathbf{IF}, R)$ is equivalent to prove:
  - i) $Q \Rightarrow domain(BB)$
  - ii) $Q \Rightarrow BB$
  - iii) $Q \wedge B_i \Rightarrow wp(S_i, R)$

Example

- Prove that $wp(IF, abs(x) = z)$ is `true`, where:
  $\mathbf{IF}: \quad \mathbf{if} :: \mathbf{x} \geq \mathbf{0} \rightarrow \mathbf{z} = \mathbf{x} \mid \mathbf{x} \leq \mathbf{0} \rightarrow \mathbf{z} = -\mathbf{x}\ \mathbf{fi}$
- $wp(\text{“}\mathbf{if}\ \ \mathbf{fi}\text{”}, \mathtt{true})$
- $wp(\text{“}\mathbf{if} :: \mathbf{x} = \mathbf{0} \rightarrow \mathbf{z} = \mathbf{x} + \mathbf{1} \ :: \ \mathbf{x} = \mathbf{0} \rightarrow \mathbf{z} = \mathbf{2} * \mathbf{x} + \mathbf{1}\ \mathbf{fi}\text{”}, \mathbf{z} = \mathbf{1})$

More examples

- Quotient and reminder

$$
\begin{array}{l}
\{q * d + r = n \wedge r \geq 0\} \\
\textbf{if} \quad :: d \leq r \quad \rightarrow \quad r = r - d; q = q + 1 \\
\qquad :: d > r \quad \rightarrow \quad \texttt{skip} \\
\textbf{fi} \\
\{q * d + r = n \wedge r \geq 0\}
\end{array}
$$

- $wp(\text{“}\textbf{if} :: \mathbf{a} = \mathbf{0} \rightarrow \texttt{skip}\ \textbf{fi}\text{”}, \texttt{true})$
- Obtain the *wp* for

$$
\begin{array}{l}
\textbf{if} \quad :: a > b \quad \rightarrow \quad a = a - b \\
\qquad :: b > a \quad \rightarrow \quad b = b - a \\
\textbf{fi} \\
\{a > 0 \wedge b > 0\}
\end{array}
$$

Iterative instructions

- *wp* for a **do** knowing the number k of iterations $wp_k(\mathbf{DO}, \mathbf{R})$.
- Example with $k = 3$ iterations:

  $\{wp_3(\mathbf{DO}, \mathbf{R})\ ?\ \}$
  do $::\ \mathbf{B_1} \rightarrow \mathbf{S_1}\ ::\ \mathbf{B_2} \rightarrow \mathbf{S_2}\ ::$ **else** $\rightarrow$ **break od**
  $\{\ R\ \}$

  is equivalent to:

  if $::\ \mathbf{B_1} \rightarrow \mathbf{S_1}\ ::\ \mathbf{B_2} \rightarrow \mathbf{S_2}$ **fi**;
  if $::\ \mathbf{B_1} \rightarrow \mathbf{S_1}\ ::\ \mathbf{B_2} \rightarrow \mathbf{S_2}$ **fi**;
  if $::\ \mathbf{B_1} \rightarrow \mathbf{S_1}\ ::\ \mathbf{B_2} \rightarrow \mathbf{S_2}$ **fi**
  $\{\ R\ \}$

Iterative instructions

- *wp* for a **do**:
  i) To stop in 0 iterations:
  $$wp_0(\mathbf{DO},\mathbf{R}) \quad = \quad \neg BB \wedge R$$
  ii) For stopping in $k > 0$:
  $$wp_k(\mathbf{DO},\mathbf{R}) \quad = \quad wp(\mathbf{IF},\mathbf{wp_{k-1}}(\mathbf{DO},\mathbf{R}))$$
  iii) For any number of iterations:
  $$wp(\mathbf{DO},\mathbf{R}) \quad = \quad (\exists K : 0 \leq K : wp_K(\mathbf{DO},\mathbf{R})))$$
- Unfeasible. The expansion of $wp(\mathbf{DO},\mathbf{R})$ is recursive and it would never stop.

Iterative instructions

- We use the following technique:
  1. To prove partial correctness: we try to obtain a formula called invariant, true before and after each loop iteration.
  2. To prove termination: we will look for a function t, called bound function, that in each iteration is > 0 and strictly decreasing.decreciente.

Instrucción iterativa

- We will usually write:

$$\{Q : \ldots\}$$
$$\langle \textit{loop initialization} \rangle$$
$$\{inv\ P : \ldots\}$$
$$\{bound\ t : \ldots\}$$
$$\mathbf{do}\ :: \mathbf{B_1} \rightarrow \mathbf{S_1} \cdots :: \mathbf{B_n} \rightarrow \mathbf{S_n} :: \mathbf{else} \rightarrow \mathbf{break\ od}$$
$$\{R : \ldots\}$$

- BB stands for $B_1 \vee \cdots \vee B_n$ and so $else = \neg BB$
- Proof method: proving five steps:
  1. P true before starting the loop,
  2. $\{P \wedge B_i\}S_i\{P\}$ in each branch,
  3. $P \wedge \neg BB \Rightarrow R$,
     (up to this point we get partial correctness)
  4. $P \wedge BB \Rightarrow (t > 0)$,
  5. $\{P \wedge B_i\}t1 = t; S_i\{t < t1\}$ for each branch.
     (now: total correctness)

Iterative instruction

- Example: Euclid's greatest common divisor

$$
\begin{array}{l}
\{Q : X > 0 \wedge Y > 0\} \\
a = X; \\
b = Y; \\
\{P : a > 0 \wedge b > 0 \wedge gcd(a,b) = gcd(X,Y)\} \\
\{t : a + b\} \\
\quad \textbf{do} \\
\quad\quad :: \quad a > b \quad \rightarrow a = a - b \\
\quad\quad :: \quad a < b \quad \rightarrow b = b - a \\
\quad\quad :: \quad else \quad\;\; \rightarrow break \\
\quad \textbf{od} \\
\{R : a = gcd(X,Y)\}
\end{array}
$$

Euclid's greatest common divisor

1. P true before starting the loop

$$
\begin{array}{l}
\{Q : X > 0 \land Y > 0\} \\
\{(P^b_Y)^a_X\} \\
a = X; \\
\{P^b_Y\} \\
b = Y; \\
\{P : a > 0 \land b > 0 \land gcd(a,b) = gcd(X,Y)\}
\end{array}
$$

Prove that $Q \rightarrow (P^b_Y)^a_X$

$$(P^b_Y)^a_X = \underbrace{X > 0 \land X > 0}_{Q} \land \underbrace{gcd(X,Y) = gcd(X,Y)}_{\texttt{true}}$$

Euclid's greatest common divisor

2.1 P is invariant for loop branch 1: $a > b \wedge P \stackrel{?}{\rightarrow} wp(\text{“}a = a - b\text{''}, P)$

$$P^a_{a-b} = \underbrace{a - b > 0}_{\substack{\uparrow \\ a>b}} \wedge \underbrace{b > 0}_{\substack{\uparrow \\ P}} \wedge gcd(a-b, b) = \underbrace{gcd(X, Y)}_{=gcd(a,b) \text{ by } P}$$

We prove that $gcd(a-b, b) = gcd(a, b)$ as follows:

1. Any common divisor k of $a - b, b$ is a common divisor of a, b.

$$\begin{array}{rl} b = k \cdot n, \quad a - b = k \cdot m & k \text{ common divisor of } b, a-b \\ a - (k \cdot n) = k \cdot m & \text{replace } b \text{ by } k \cdot n \\ a = k \cdot m + k \cdot n = k \cdot (m+n) & k \text{ is also a divisor of } a \end{array}$$

2. Any common divisor k of a, b is also a common divisor of $a - b, b$.

$$\begin{array}{rl} a = k \cdot h, \quad b = k \cdot n & k \text{ common divisor of } a, b \\ a - b = k \cdot h - k \cdot n = k \cdot (h - n) & k \text{ is also a divisor of } a - b \end{array}$$

Euclid's greatest common divisor

2.1 P is invariant for loop branch 2: $a < b \wedge P \stackrel{?}{\rightarrow} wp(\text{“}b = b - a\text{''}, P)$

We don't need to prove it: analogous to branch 1

We can switch roles of a and b!

Euclid's greatest common divisor

3 P and exiting the loop implies R: $P \wedge a = b \stackrel{?}{\rightarrow} a = gcd(X, Y)$

$$\underbrace{a > 0 \wedge b > 0 \wedge gcd(a, b) = gcd(X, Y)}_{P} \wedge a = b$$
$$\rightarrow \quad \underbrace{gcd(a, a)}_{a} = gcd(X, Y)$$

We have proved partial correctness

Euclid's greatest common divisor

4 $t > 0$ each time we enter the loop: $P \wedge a \neq b \stackrel{?}{\rightarrow} a + b > 0$
Obvious because $P \rightarrow a > 0 \wedge b > 0$ and so $a + b > 0$

5.1 Branch 1 decreases the bound function t:
$P \wedge a > b \stackrel{?}{\rightarrow} wp(\text{``}t1 = a + b; a = a - b'', a + b < t1)$

$$\begin{array}{cl} & wp(\text{``}t1 = a + b; a = a - b'', a + b < t1) \\ = & ((a + b < t1)^{a}_{a-b})^{t}1_{a+b} \\ = & a - b + b < a + b \\ \equiv & -b < 0 \\ \equiv & b > 0 \end{array}$$

which holds because of P

5.2 Branch 2 decreases the bound function t: analogous to previous step. Switch roles of a and b

Iterative instruction

- Another example:

$$
\begin{array}{l}
\{b \geq 0\} \\
x = a;\ y = b;\ z = 0; \\
\{P : y \geq 0 \land z + x * y = a * b\} \\
\{t : y\} \\
\quad \textbf{\textit{do}} \\
\qquad :: \quad y > 0 \land even(y) \quad \rightarrow y = y/2;\ x = x + x \\
\qquad :: \quad \neg even(y) \quad \rightarrow y = y - 1;\ z = z + x \\
\qquad :: \quad else \quad \rightarrow break \\
\quad \textbf{\textit{od}} \\
\{R : z = a * b\}
\end{array}
$$