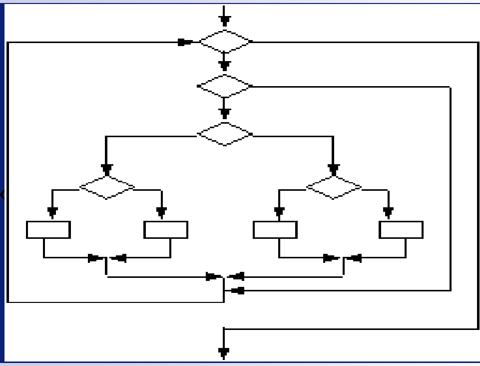
Spring 2006



Program correctness

Weakest preconditions

Marcello Bonsangue

Axiomatic semantics

- We have a language for asserting properties of programs (syntax).
- We know when an assertion is true (validity).
- We have a symbolic way for deriving assertions (proof system).
- What is the relation between validity and provability?

Hoare Logic soundness and completeness

- Soundness (what can be proved is valid):

$$\vdash_{par} \{\phi\}\ c\ \{\psi\} \quad \text{implies} \quad \models_{par} \{\phi\}\ c\ \{\psi\}$$

- Completeness (what is valid can be proved):

$$\models_{par} \{\phi\}\ c\ \{\psi\} \quad \text{implies} \quad \vdash_{par} \{\phi\}\ c\ \{\psi\}$$

Soundness

- Theorem: The proof system for partial correctness is sound

  equivalently, if $\vdash_{par} \{\phi\}\ c\ \{\psi\}$ then

  $\forall\sigma, I\ \ (\sigma, I \ \vDash_{par} \phi \text{ and } \langle c, \sigma\rangle \to \sigma'\ ) \Rightarrow \sigma', I \vDash_{par} \psi$

  Proof by induction on the length of the derivation of the Hoare triples, reasoning about each axiom and rule separately. (why?)

Soundness of skip

Case: last rule used in the derivation is

$$\{\phi\}\ \underline{\text{skip}}\ \{\phi\}.$$

We have to prove

$$\forall \sigma, I\ (\sigma, I \vDash_{par} \phi \text{ and } <\underline{\text{skip}}, \sigma> \rightarrow \sigma') \Rightarrow \sigma', I \vDash_{par} \phi$$

Which follows because $\sigma' = \sigma$.

Soundness of assignment

Case last rule in the derivation is $\{\phi[a/x]\}\ x := a\ \{\phi\}$

Take σ and I such that $\sigma, I \models \phi[a/x]$. Then

$$< x := a, \sigma > \to \sigma[a/x]$$

We need to prove $\sigma[a/x], I \models \phi$, which follows from the substitution lemma

LEMMA: $\sigma, I \models \phi[a/x]$ implies $\sigma[a/x], I \models \phi$

Proof: by induction on the structure of ϕ

Soundness of consequence rule

- Case last rule in the derivation is

$$\frac{\vdash \phi \Rightarrow \phi' \quad \{\phi'\}\ c\ \{\psi'\} \quad \vdash \psi' \Rightarrow \psi}{\{\phi\}\ c\ \{\psi\}}$$

- From soundness of first order logic we have

$$\sigma, I \models \phi \Rightarrow \phi'.$$

  Hence $\sigma, I \models \phi'$.

- From induction hypothesis we get $\sigma', I \models \psi'$.

- From soundness of first order logic we finally obtain

$$\sigma', I \models \psi' \Rightarrow \psi\ .$$

  Therefore $\sigma', I \models \psi$

Soundness of while

- Case last rule in the derivation is

$$\frac{\{\phi \wedge b\}\ c\ \{\phi\}}{\{\phi\}\ \underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}\ \{\phi \wedge \neg b\}}$$

- Assume $\sigma, I \vDash \phi$. We proceed by induction on the derivation of $\langle \underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \sigma\rangle \to \sigma'$
  - There are two cases (we treat only one):

$$\frac{\langle b, \sigma\rangle \to T \quad \langle c, \sigma\rangle \to \sigma' \quad \langle \underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \sigma'\rangle \to \sigma''}{\langle \underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \sigma\rangle \to \sigma''}$$

  - We need to prove $\sigma'', I \vDash \phi \wedge \neg b$

Soundness of while (II)

- By definition of derivation of $<b, \sigma> \to T$ we obtain

$$\sigma, I \models b$$

  Hence $\sigma, I \models \phi \wedge b$

- By induction hypothesis on derivation of $\{\phi \wedge b\}$ c $\{\phi\}$ we have

$$\sigma', I \models \phi$$

- By induction hyp. on derivation of $<\underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \sigma'> \to \sigma''$ we finally obtain

$$\sigma'', I \models \phi \wedge \neg b$$

Hoare Logic

- We have seen that if we can derive an assertion in the Hoare logic then this assertion is true (soundness).

- Next we concentrate on the opposite direction (completeness).

Completeness of Hoare Logic

- Can we prove that if an assertion is true then it is derivable?

- More formally, can we prove

$$\models_{par} \{\phi\}\ c\ \{\psi\} \quad \text{implies} \vdash_{par} \{\phi\}\ c\ \{\psi\}?$$

- The answer is yes, but only if the underlying logic is complete ($\models \phi$ implies $\vdash \phi$) and expressive enough
  - This is called relative completeness.

Idea for proving completeness

- To prove $\models_{tot} \{\phi\}\ c\ \{\psi\}$ implies $\vdash_{tot} \{\phi\}\ c\ \{\psi\}$

1. Assume we can compute $wp(c,\psi)$ such that
   - $wp(c,\psi)$ is a precondition of ψ, i.e.

     $$\vdash_{tot} \{wp(c,\psi)\}\ c\ \{\psi\}$$

   - $wp(c,\psi)$ is the weakest precondition of ψ, i.e.

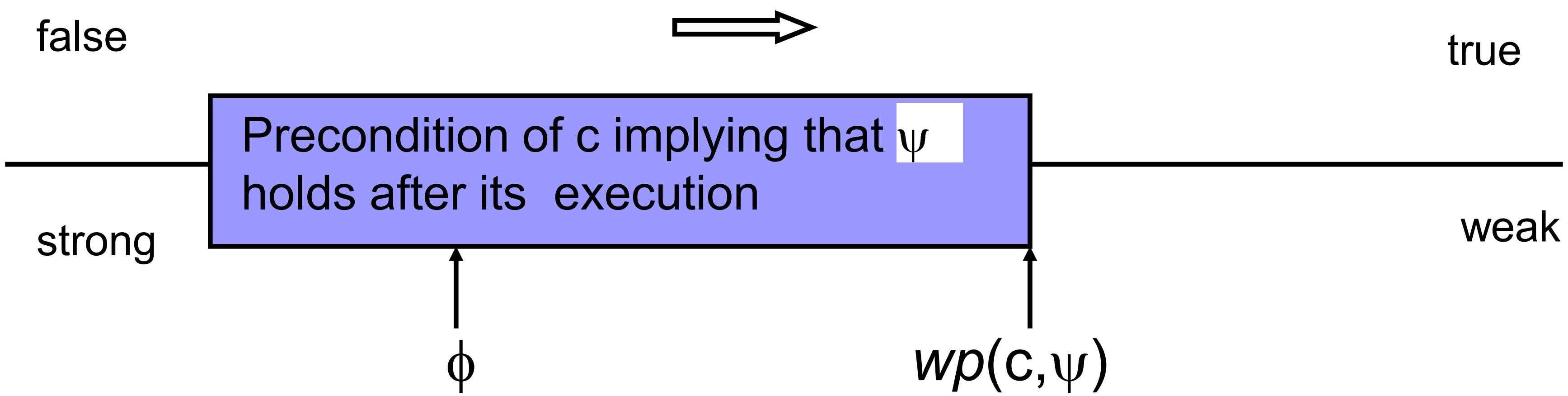
     $$\models_{tot} \{\varphi\}\ c\ \{\psi\} \text{ implies } \models \varphi \Rightarrow wp(c,\psi)$$

2. By completeness of the underlying logic and the consequence rule we obtain

$$\frac{\vdash \phi \Rightarrow wp(c,\psi) \qquad \vdash_{tot} \{wp(c,\psi)\}\ c\ \{\psi\}}{\vdash_{tot} \{\phi\}\ c\ \{\psi\}}$$

Weakest precondition (Dijkstra)

- Assertions can be ordered



- Thus to verify $\{\phi\}$ c $\{\psi\}$ we compute *wp*(c,ψ) and prove $\phi \Rightarrow$ *wp*(c,ψ)

Weakest precondition

- The definition of the weakest precondition follows the rules of the Hoare logic

- SKIP

$$\frac{}{\{\phi\}\ \text{skip}\ \{\phi\}}$$

$$wp(\text{skip},\phi) = \phi$$

Weakest precondition

- ASSIGNMENT

$$\frac{}{\{\phi[a/x]\}\ x := a\ \{\phi\}}$$

$$wp(x{:=}a,\phi) = \phi[a/x]$$

- SEQUENTIAL COMPOSITION

$$\frac{\{\phi\}\ c_1\ \{\psi\} \qquad \{\psi\}\ c_2\ \{\varphi\}}{\{\phi\}\ c_1;\ c_2\ \{\varphi\}}$$

$$wp(c_1;\ c_2,\ \varphi) = wp(c_1, wp(c_2,\ \varphi))$$

Weakest precondition

- CONDITIONAL

$$\frac{\{\phi_1\}\ c_1\ \{\psi\} \qquad \{\phi_2\}\ c_2\ \{\psi\}}{\{b \Rightarrow \phi_1 \wedge \neg b \Rightarrow \phi_2\}\ \underline{\text{if}}\ b\ \underline{\text{then}}\ c_1\ \underline{\text{else}}\ c_2\ \underline{\text{fi}}\ \{\psi\}}$$

$$wp(\underline{\text{if}}\ b\ \underline{\text{then}}\ c_1\ \underline{\text{else}}\ c_2\ \underline{\text{fi}},\ \psi) = b \Rightarrow wp(c_1,\psi) \wedge \neg b \Rightarrow wp(c_2,\psi)$$

Weakest precondition

- LOOP
  1. We already know that

     while b do c od $\equiv$ if b then (c;while b do c od) else skip fi

  2. Let w = while b do c od and W = wp(w, ψ). We have

$$W = b \Rightarrow wp(c,W) \wedge \neg b \Rightarrow \psi$$

  3. This is a recursive equation
     - We know how to solve it
     - We need a complete partial order (cpo) of assertions

A CPO of assertions

- Refinement order:

$$\phi \leq \psi \text{ iff } \quad \models \psi \Rightarrow \phi$$

  True is the bottom: it does not says much about a state.

- It forms a complete partial order: the least upper bound of every chain $\phi_1 \leq \phi_2 \leq \ldots \leq \phi_n \leq$ is the infinite conjunction $\bigwedge \phi_i$

  where $\sigma, I \models \bigwedge \phi_i$ iff $\sigma, I \models \phi_i$ for all i

Weakest precondition (LOOP)

- Let $F(X) = b \Rightarrow wp(c, X) \wedge \neg b \Rightarrow \psi$.

- Then F is monotone and continuous. Thus it has a least fixed point (the weakest fixed point) and

$$wp(\underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \psi) = \bigwedge F^i(\text{true})$$

- We need an assertion language expressive enough to be able to write $\bigwedge F^i(\text{true})$.

Weakest precondition (LOOP)

- Define a family of preconditions $wp(\underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \psi)_k$ as follows:

  $wp(\underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \psi)_0 \quad = \neg b \Rightarrow \psi$

  $wp(\underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \psi)_{n+1} =$

  $\qquad b \Rightarrow wp(c, wp(\underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \psi)_n) \wedge \neg b \Rightarrow \psi$

  Then $wp(\underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \psi) = \bigwedge wp(\underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \psi)_k$

- Here $wp(\underline{\text{while}}\ b\ \underline{\text{do}}\ c\ \underline{\text{od}}, \psi)_k$ is the weakest precondition on which the loop - if terminated in k or less iterations - terminates in ψ.

Weakest precondition: properties

- For each command c in our language we have
  - wp(c,true) = true
  - if $\psi \Rightarrow \psi'$ then $wp(c, \psi) \Rightarrow wp(c, \psi')$
  - $wp(c, \psi \wedge \psi') = wp(c, \psi) \wedge wp(c, \psi')$
  - $wp(c, \psi \vee \psi') = wp(c, \psi) \vee wp(c, \psi')$

- wp(c,false) characterizes all states in which c does not terminate