Program Specifications and Proofs

Syntax:

$$\begin{aligned} \mathit{spec} ::= &\ [\mathit{assert}]\ \mathit{comm}\ [\mathit{assert}] && \text{total correctness} \\ \mid &\ \{\mathit{assert}\}\ \mathit{comm}\ \{\mathit{assert}\} && \text{partial correctness} \end{aligned}$$

Semantics:

$$[\![-]\!]_{\mathit{spec}} \in \mathit{spec} \to \mathbf{B} \qquad s \text{ is valid when } [\![s]\!]_{\mathit{spec}} = \mathbf{true}$$

$$\begin{aligned} [\![\,[p]\ c\ [q]\,]\!]_{\mathit{spec}} &= \forall \sigma \in \Sigma.\ [\![p]\!]_{\mathit{assert}}\sigma \Rightarrow \\ &\qquad ([\![c]\!]_{\mathit{comm}}\sigma \neq \bot \text{ and } [\![q]\!]_{\mathit{assert}}([\![c]\!]_{\mathit{comm}}\sigma)) \\ [\![\{p\}\ c\ \{q\}]\!]_{\mathit{spec}} &= \forall \sigma \in \Sigma.\ [\![p]\!]_{\mathit{assert}}\sigma \Rightarrow \\ &\qquad ([\![c]\!]_{\mathit{comm}}\sigma = \bot \text{ or } [\![q]\!]_{\mathit{assert}}([\![c]\!]_{\mathit{comm}}\sigma)) \end{aligned}$$

or, when $\mathbf{B}$ is ordered by $\mathbf{false} \sqsubseteq \mathbf{true}$,

$$\begin{aligned} [\![\,[p]\ c\ [q]\,]\!]_{\mathit{spec}} &= [\![p]\!]_{\mathit{assert}} \sqsubseteq ([\![q]\!]_{\mathit{assert}})_{\perp\!\!\!\perp} \cdot [\![c]\!]_{\mathit{comm}} \\ [\![\{p\}\ c\ \{q\}]\!]_{\mathit{spec}} &= (\mathit{not} \cdot [\![q]\!]_{\mathit{assert}})_{\perp\!\!\!\perp} \cdot [\![c]\!]_{\mathit{comm}} \sqsubseteq \mathit{not} \cdot [\![p]\!]_{\mathit{assert}} \end{aligned}$$

Properties of Program Specs

If $[\![\, [p] \; c \; [q] \,]\!]_{spec}$ and $[\![c]\!]_{comm} \sqsubseteq [\![c']\!]_{comm}$, then $[\![\, [p] \; c' \; [q] \,]\!]_{spec}$.

If $[\![\{p\} \; c \; \{q\}]\!]_{spec}$ and $[\![c']\!]_{comm} \sqsubseteq [\![c]\!]_{comm}$, then $[\![\{p\} \; c' \; \{q\}]\!]_{spec}$.

If $[\![\{p\} \; c_i \; \{q\}]\!]_{spec}$ for all c_i,

and $[\![c_0]\!]_{comm} \sqsubseteq [\![c_1]\!]_{comm} \sqsubseteq \ldots$,

and $[\![c]\!]_{comm} = \sqcup_i [\![c_i]\!]_{comm}$,

then $[\![\{p\} \; c \; \{q\}]\!]_{spec}$.

Examples of Program Specs

$\{x-y > 3\}$	$x := x-y$	$\{x > 2\}$	valid
$[x-y > 3]$	$x := x-y$	$[x > 2]$	valid
$\{x \leq 10\}$	**while** $x \neq 10$ **do** $x := x+1$	$\{x=10\}$	valid
$[x \leq 10]$	**while** $x \neq 10$ **do** $x := x+1$	$[x=10]$	valid
$\{\mathbf{true}\}$	**while** $x \neq 10$ **do** $x := x+1$	$\{x=10\}$	valid
$[\mathbf{true}]$	**while** $x \neq 10$ **do** $x := x+1$	$[x=10]$	invalid

Basic Inference Rules for Specifications

assignment (AS)

$$\frac{}{[\![p/v \rightarrow e]\!] \; v := e \; [\![p]\!]}$$

sequential composition (SQ)

$$\frac{[\![p]\!] \; c_1 \; [\![q]\!] \qquad [\![q]\!] \; c_2 \; [\![r]\!]}{[\![p]\!] \; c_1 \; ; \; c_2 \; [\![r]\!]}$$

strengthening precedent (SP)

$$\frac{p \Rightarrow q \qquad [\![q]\!] \; s \; [\![r]\!]}{[\![p]\!] \; s \; [\![r]\!]}$$

weakening consequent (WC)

$$\frac{[\![p]\!] \; s \; [\![q]\!] \qquad q \Rightarrow r}{[\![p]\!] \; s \; [\![r]\!]}$$

(SP) and (WC) are noncompositional (not syntax-directed).

Soundness of the Assignment Rule

Recall the Substitution Theorem for predicate logic:

if $[\![\delta-]\!]\sigma' = \sigma$ (on $FV(p)$), then $[\![p/\delta]\!]\sigma' = [\![p]\!]\sigma$

Let σ' be a state satisfying $p/v \rightarrow e$: $[\![p/v \rightarrow e]\!]_{assert}\sigma' = \mathbf{true}$.

Let $\sigma = [\![v := e]\!]_{comm}\sigma' = [\sigma' | v : [\![e]\!]_{intexp}\sigma']$.

Let $\delta = (v \rightarrow e) = [c_{\text{var}} | v : e]$.

Then $[\![\delta v]\!]_{intexp}\sigma' = [\![e]\!]_{intexp}\sigma' = \sigma v$, and
$[\![\delta u]\!]_{intexp}\sigma' = [\![u]\!]_{intexp}\sigma' = [\![u]\!]_{intexp}\sigma$ for $u \neq v$.

By the Substitution Theorem, $[\![p/v \rightarrow e]\!]\sigma' = [\![p]\!]\sigma$.

Derived Rule for Multiple Sequential Composition

$$(\text{MSQ}_n) \qquad \frac{\begin{array}{c} p_0 \Rightarrow q_0 \\ [\![q_0]\!]\ c_0\ [\![p_1]\!] \quad p_1 \Rightarrow q_1 \\ \ldots \\ [\![q_{n-1}]\!]\ c_{n-1}\ [\![p_n]\!] \quad p_n \Rightarrow q_n \end{array}}{[\![p_0]\!]\ c_0\ ;\ \ldots\ ;\ c_{n-1}\ [\![q_n]\!]}$$

Derivation of (MSQ_1):

1.	$p_0 \Rightarrow q_0$	assumption
2.	$[\![q_0]\!]\ c_0\ [\![p_1]\!]$	assumption
3.	$[\![p_0]\!]\ c_0\ [\![p_1]\!]$	SP $(1, 2)$
4.	$p_1 \Rightarrow q_1$	assumption
5.	$[\![p_0]\!]\ c_0\ [\![q_1]\!]$	WC $(3, 4)$

(MSQ_n) derived from (MSQ_{n-1}) and (SQ).

A Simple Derivation

Prove validity of

$$[y > 3]\ x := 2*y\ ;\ x := x-y\ [x \geq 4]$$

1.	$y > 3 \Rightarrow (2*y)-y > 3$	(predicate logic)
2.	$[(2*y)-y > 3]\ x := 2*y\ [x-y > 3]$	AS
3.	$[x-y > 3]\ x := x-y\ [x > 3]$	AS
4.	$x > 3 \Rightarrow x \geq 4$	(predicate logic)
5.	$[y > 3]\ x := 2*y\ ;\ x := x-y\ [x \geq 4]$	$MSQ_2\ (1, 2, *, 3, 4)$

Derived Rule for Repeated Assignment

Derived from MSQ_n:

$$\mathrm{RAS}_n \ \frac{p \Rightarrow (\ldots(q/v_{n-1} \rightarrow e_{n-1})\ldots/v_0 \rightarrow e_0)}{[\![p]\!]\ v_0 \texttt{:=} e_0\ \texttt{;}\ \ldots v_{n-1} \texttt{:=} e_{n-1}\ [\![q]\!]}$$

The previous example can now be proved by

1.	$\texttt{y} > 3 \Rightarrow \texttt{(2*y)-y} \geq 4$	(predicate logic)
2.	$\texttt{[y > 3] x:=2*y ; x:=x-y [x} \geq \texttt{4]}$	RAS_2 (1)

Rule (RAS_n) is sound and complete:

if its conclusion is valid, its premiss is valid

Rules for the while Construct

Partial correctness of **while**:

$$\text{(WHP)}\ \frac{\{i \wedge b\}\ c\ \{i\}}{\{i\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{i \wedge \neg b\}}$$

i is the loop invariant.

Total correctness of **while**:

$$\text{(WHT)}\ \frac{[i \wedge b \wedge (e = v_0)]\ c\ [i \wedge (e < v_0)] \quad i \wedge b \Rightarrow e \geq 0}{[i]\ \textbf{while}\ b\ \textbf{do}\ c\ [i \wedge \neg b]}$$

where $v_0 \notin FV(i) \cup FV(b) \cup FV(e) \cup FV(c)$.

Proof of Soundness for (WHP)

$$\text{(WHP)}\ \ \frac{\{i \wedge b\}\ c\ \{i\}}{\{i\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{i \wedge \neg b\}}$$

Recall the approximation to **while** b **do** c:

$$\begin{aligned} w_0 &\stackrel{\text{def}}{=} \textbf{while true do skip} \\ \Rightarrow m_0\sigma &= \bot \qquad\qquad (\text{where } m_i = [\![w_i]\!]_{comm}) \end{aligned}$$

$$\begin{aligned} w_{i+1} &\stackrel{\text{def}}{=} \textbf{if}\ b\ \textbf{then}\ (c\ ;\ w_i)\ \textbf{else skip} \\ \Rightarrow m_{i+1}\sigma &= \textbf{if}\ [\![b]\!]\sigma\ \textbf{then}\ (m_i)_{\perp\!\!\perp}([\![c]\!]\sigma)\ \textbf{else}\ \sigma \end{aligned}$$

Recall meaning of partial correctness specs:

$$[\![\{p\}\ c\ \{q\}]\!]_{spec} = (not \cdot [\![q]\!]_{assert})_{\perp\!\!\perp} \cdot [\![c]\!]_{comm} \ \sqsubseteq\ not \cdot [\![p]\!]_{assert}$$

Goal: prove that if $\quad (not \cdot [\![i]\!])_{\perp\!\!\perp} \cdot [\![c]\!] \ \sqsubseteq\ not \cdot [\![i \wedge b]\!]$

then $\quad (not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp} \cdot (\sqcup_i\, m_i) \ \sqsubseteq\ not \cdot [\![i]\!]$

Proof of Soundness for (WHP), cont'd

Goal: prove that if $\quad (not \cdot [\![i]\!])_{\perp\!\!\perp} \cdot [\![c]\!] \;\sqsubseteq\; not \cdot [\![i \wedge b]\!]$

then $\quad (not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp} \cdot (\bigsqcup_n m_n) \;\sqsubseteq\; not \cdot [\![i]\!]$

$$(not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp} \cdot (\textstyle\bigsqcup_n m_n) \;\sqsubseteq\; not \cdot [\![i]\!]$$

$$\text{if} \quad \bigsqcup_n ((not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp} \cdot m_n) \;\sqsubseteq\; not \cdot [\![i]\!]$$

$$\text{if } \forall n. \quad (not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp} \cdot m_n \;\sqsubseteq\; not \cdot [\![i]\!]$$

By induction on n: easy for $n = 0$; if true for n then for any σ

$$\begin{aligned}
&(not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp}(m_{n+1}\sigma) \\
&\quad = \textbf{if } [\![b]\!]\sigma \textbf{ then } (not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp}((m_n)_{\perp\!\!\perp}([\![c]\!]\sigma)) \\
&\quad\ \textbf{else } (not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp}\sigma
\end{aligned}$$

Proof of Soundness for (WHP), cont'd

Goal: prove that if

$$(not \cdot [\![i]\!])_{\perp\!\!\perp} \cdot [\![c]\!] \sqsubseteq not \cdot [\![i \wedge b]\!]$$

and

$$(not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp} \cdot m_n \sqsubseteq not \cdot [\![i]\!]$$

then

$$(not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp} \cdot m_{n+1} \sqsubseteq not \cdot [\![i]\!]$$

$$\begin{aligned}
&(not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp}(m_{n+1}\sigma) \\
&\quad = \textbf{if } [\![b]\!]\sigma \textbf{ then } (not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp}((m_n)_{\perp\!\!\perp}([\![c]\!]\sigma)) \\
&\qquad\; \textbf{else } (not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp}\sigma
\end{aligned}$$

$$\begin{aligned}
(not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp}((m_n)_{\perp\!\!\perp}([\![c]\!]\sigma)) &= ((not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\perp} \cdot m_n)_{\perp\!\!\perp}([\![c]\!]\sigma) \\
&\sqsubseteq (not \cdot [\![i]\!])_{\perp\!\!\perp}([\![c]\!]\sigma) \\
&\sqsubseteq (not \cdot [\![i \wedge b]\!])_{\perp\!\!\perp}\sigma
\end{aligned}$$

Proof of Soundness for (WHP), cont'd

Goal: prove that if $(not \cdot [\![i]\!])_{\perp\!\!\!\perp} \cdot [\![c]\!] \sqsubseteq not \cdot [\![i \wedge b]\!]$

and $(not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\!\perp} \cdot m_n \sqsubseteq not \cdot [\![i]\!]$

then $(not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\!\perp} \cdot m_{n+1} \sqsubseteq not \cdot [\![i]\!]$

$$
\begin{aligned}
&(not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\!\perp}(m_{n+1}\sigma) \\
&\qquad \sqsubseteq\ \mathbf{if}\ [\![b]\!]\sigma\ \mathbf{then}\ (not \cdot [\![i \wedge b]\!])_{\perp\!\!\!\perp}\sigma\ \mathbf{else}\ (not \cdot [\![i \wedge \neg b]\!])_{\perp\!\!\!\perp}\sigma \\
&\qquad =\ (not \cdot [\![i]\!])\sigma
\end{aligned}
$$

QED

Soundness of (WHT)

$$\text{(WHT)} \quad \frac{[i \wedge b \wedge (e = v_0)]\ c\ [i \wedge (e < v_0)] \qquad i \wedge b \Rightarrow e \geq 0}{[i]\ \textbf{while}\ b\ \textbf{do}\ c\ [i \wedge \neg b]}$$

where the ghost variable v_0 is not in $FV(i) \cup FV(b) \cup FV(e) \cup FV(c)$.

Idea: e serves as a loop counter with initial value v_0.

The first premiss: the counter is decreased by execution of c.

The second premiss: when the counter becomes negative,

b is false, and the loop terminates (invariant i is always satisfied).

Rules for the while Construct

Partial correctness of **while**:

$$\text{(WHP)}\quad \frac{\{i \wedge b\}\ c\ \{i\}}{\{i\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{i \wedge \neg b\}}$$

where i is the loop invariant.

Total correctness of **while**:

$$\text{(WHT)}\quad \frac{[i \wedge b \wedge (e = v_0)]\ c\ [i \wedge (e < v_0)] \qquad i \wedge b \Rightarrow e \geq 0}{[i]\ \textbf{while}\ b\ \textbf{do}\ c\ [i \wedge \neg b]}$$

where $v_0 \notin FV(i) \cup FV(b) \cup FV(e) \cup FV(c)$
e is the loop variant.

More Compositional Rules

skip (SK)

$$\frac{}{[\![p]\!]\ \mathbf{skip}\ [\![p]\!]}$$

implication and **skip** (ISK)
(derived from (SK),(SP))

$$\frac{p \Rightarrow q}{[\![p]\!]\ \mathbf{skip}\ [\![q]\!]}$$

conditional (CD)

$$\frac{[\![p \wedge b]\!]\ c\ [\![q]\!] \qquad [\![p \wedge \neg b]\!]\ c'\ [\![q]\!]}{[\![p]\!]\ \mathbf{if}\ b\ \mathbf{then}\ c\ \mathbf{else}\ c'\ [\![q]\!]}$$

variable declaration (DC′)

$$\frac{[\![p]\!]\ v := e\ ;\ c\ [\![q]\!]}{[\![p]\!]\ \mathbf{newvar}\ v := e\ \mathbf{in}\ c\ [\![q]\!]} \quad v \notin FV(q)$$

More Non-Compositional Rules

renaming (RN)

$$\frac{[\![p]\!]\ c\ [\![q]\!]}{[\![p']\!]\ c'\ [\![q']\!]}$$

where p', c', q' are obtained by renaming bound variables in p, c, q

conjunction (CA)

$$\frac{[\![p]\!]\ c\ [\![q]\!] \qquad [\![p']\!]\ c\ [\![q']\!]}{[\![p \wedge p']\!]\ c\ [\![q \wedge q']\!]}$$

disjunction (DA)

$$\frac{[\![p]\!]\ c\ [\![q]\!] \qquad [\![p']\!]\ c\ [\![q']\!]}{[\![p \vee p']\!]\ c\ [\![q \vee q']\!]}$$

constancy for partial correctness (CSP)

$$\frac{}{\{p\}\ c\ \{p\}} \quad FV(p) \cap FA(c) = \{\}$$

constancy for total correctness (CST)

$$\frac{[q]\ c\ [r]}{[p \wedge q]\ c\ [p \wedge r]} \quad FV(p) \cap FA(c) = \{\}$$

Specification of a Factorial Computation

`f:=1` ; **while** `n>0` **do** (`f:=f*n` ; `n:=n-1`)

Specification of a Factorial Computation

$$[\texttt{n=m}]\ \texttt{f:=1}\ ;\ \textbf{while}\ \texttt{n>0}\ \textbf{do}\ (\texttt{f:=f*n}\ ;\ \texttt{n:=n-1})\ [\texttt{n<0} \vee \texttt{f=m!}]$$

Proof: by (SP) and (DA) from

$$(G1)\ [\texttt{n<0}]\ \texttt{f:=1}\ ;\ \textbf{while}\ \texttt{n>0}\ \textbf{do}\ (\texttt{f:=f*n}\ ;\ \texttt{n:=n-1})\ [\texttt{n<0}]$$

$$(G2)\ [\texttt{n=m} \wedge \texttt{n} \geq \texttt{0}]\ \texttt{f:=1}\ ;\ \textbf{while}\ \texttt{n>0}\ \textbf{do}\ (\texttt{f:=f*n}\ ;\ \texttt{n:=n-1})\ [\texttt{f=m!}]$$

Correctness of the Factorial Specification

To prove $(G1)$:

1 $[\texttt{n-1<0} \wedge \texttt{n-1<count}]\ (\texttt{f:=f*n ; n:=n-1})\ [\texttt{n<0} \wedge \texttt{n<count}]$ by (RAS_2)

2 $\texttt{n<0} \wedge \texttt{n>0} \wedge \texttt{n=count} \Rightarrow \texttt{n-1<0} \wedge \texttt{n-1<count}$

3 $[\texttt{n<0} \wedge \texttt{n>0} \wedge \texttt{n=count}]\ (\texttt{f:=f*n ; n:=n-1})\ [\texttt{n<0} \wedge \texttt{n<count}]$ by (SP 1,2)

4 $\texttt{n<0} \wedge \texttt{n>0} \Rightarrow \texttt{n} \geq 0$

5 $[\texttt{n<0}]$ **while** $\texttt{n>0}$ **do** $(\texttt{f:=f*n ; n:=n-1})\ [\texttt{n<0} \wedge \neg(\texttt{n>0})]$ by (WHT 3,4)

6 $[\texttt{n<0}]$ **while** $\texttt{n>0}$ **do** $(\texttt{f:=f*n ; n:=n-1})\ [\texttt{n<0}]$ by (WC 5)

7 $[\texttt{n<0}]\ \texttt{f:=1}\ [\texttt{n<0}]$ by (AS)

8 $[\texttt{n<0}]\ \texttt{f:=1 ;}$ **while** $\texttt{n>0}$ **do** $(\texttt{f:=f*n ; n:=n-1})\ [\texttt{n<0}]$ by (SQ 7,6)

Correctness of the Factorial Specification

To prove $(G2)$:

1 $\mathsf{f{=}m!/n!} \wedge \mathsf{n} \geq 0 \wedge \mathsf{n{>}0} \wedge (\mathsf{n{=}cnt}) \Rightarrow \mathsf{f{*}n{=}m!/(n{-}1)!} \wedge \mathsf{n{-}1} \geq 0 \wedge (\mathsf{n{-}1{<}cnt})$

2 $[\mathsf{f{=}m!/n!} \wedge \mathsf{n} \geq 0 \wedge \mathsf{n{>}0} \wedge (\mathsf{n{=}cnt})]\ (\mathsf{f{:=}f{*}n}\ ;\ \mathsf{n{:=}n{-}1})\ [\mathsf{f{=}m!/n!} \wedge \mathsf{n} \geq 0 \wedge (\mathsf{n{<}cnt})]$

(RAS$_2$ 1)

3 $\mathsf{f{=}m!/n!} \wedge \mathsf{n} \geq 0 \wedge \mathsf{n{>}0} \Rightarrow \mathsf{n} \geq 0$

4 $[\mathsf{f{=}m!/n!} \wedge \mathsf{n} \geq 0]$ **while** $\mathsf{n{>}0}$ **do** $(\mathsf{f{:=}f{*}n}\ ;\ \mathsf{n{:=}n{-}1})\ [\mathsf{f{=}m!/n!} \wedge \mathsf{n} \geq 0 \wedge \neg(\mathsf{n{>}0})]$

(WHT 2,3)

5 $\mathsf{f{=}1} \wedge \mathsf{n{=}m} \wedge \mathsf{n} \geq 0 \Rightarrow \mathsf{f{=}m!/n!} \wedge \mathsf{n} \geq 0$

6 $\mathsf{f{=}m!/n!} \wedge \mathsf{n} \geq 0 \wedge \neg(\mathsf{n{>}0}) \Rightarrow \mathsf{f} = \mathsf{m!}$

7 $[\mathsf{f{=}1} \wedge \mathsf{n{=}m} \wedge \mathsf{n} \geq 0]$ **while** $\mathsf{n{>}0}$ **do** $(\mathsf{f{:=}f{*}n}\ ;\ \mathsf{n{:=}n{-}1})\ [\mathsf{f} = \mathsf{m!}]$

(SP,WC 5,4,6)

8 $[\mathsf{n{=}m} \wedge \mathsf{n} \geq 0]\ \mathsf{f{:=}1}\ [\mathsf{f{=}1} \wedge \mathsf{n{=}m} \wedge \mathsf{n} \geq 0]$

(AS)

9 $[\mathsf{n{=}m} \wedge \mathsf{n} \geq 0]\ \mathsf{f{:=}1}\ ;$ **while** $\mathsf{n{>}0}$ **do** $(\mathsf{f{:=}f{*}n}\ ;\ \mathsf{n{:=}n{-}1})\ [\mathsf{f} = \mathsf{m!}]$

(SQ 8,7)