Specification as a development task

Given precondition φ and postcondition ψ

develop a program S such that

$$\boxed{\{\varphi\}\, S\, \{\psi\}}$$

For instance

Find S such that

$$\{n \geq 0\}\ S\ \{rt^2 \leq n \wedge n < (rt+1)^2\}$$

One correct solution:

$$\{n \geq 0\}$$
$$rt := 0;\ sqr := 1;$$
$$\mathbf{while}\ sqr \leq n\ \mathbf{do}\ rt := rt + 1;\ sqr := sqr + 2 * rt + 1$$
$$\{rt^2 \leq n \wedge n < (rt+1)^2\}$$

Hoare's logic: trouble #1

Another correct solution:

$$\{n \geq 0\}$$
$$\textbf{while true do skip}$$
$$\{rt^2 \leq n \wedge n < (rt+1)^2\}$$

since $\vdash$

$$\{n \geq 0\}$$
$$\textbf{while } \{\textbf{true}\} \textbf{ true do skip}$$
$$\{rt^2 \leq n \wedge n < (rt+1)^2\}$$

Partial correctness:
termination not guaranteed,
and hence not requested!

Total correctness

Total correctness $=$ partial correctness $+$ successful termination

Total correctness judgements:

$$\boxed{[\varphi]\; S\; [\psi]}$$

Intended meaning:

Whenever the program S starts in a state satisfying the precondition φ
then it terminates successfully in a final state that satisfies the postcondition ψ

Total correctness: semantics

$$\models [\varphi]\, S\, [\psi]$$

iff

$$\{\varphi\} \subseteq [\![S]\!]\, \{\psi\}$$

where for $S \in \mathbf{Stmt}$, $A \subseteq \mathbf{State}$:

$$[\![S]\!]\, A = \{s \in \mathbf{State} \mid \mathcal{S}[\![S]\!]\, s = a, \text{for some } a \in A\}$$

Spelling this out:

The total correctness judgement $[\varphi]\, S\, [\psi]$ holds, written $\models [\varphi]\, S\, [\psi]$, if for all states $s \in \mathbf{State}$

$$\text{if } \mathcal{F}[\![\varphi]\!]\, s = \mathbf{tt} \text{ then } \mathcal{S}[\![S]\!]\, s \in \mathbf{State} \text{ and } \mathcal{F}[\![\psi]\!]\, (\mathcal{S}[\![S]\!]\, s) = \mathbf{tt}$$

Total correctness: proof rules

$$\frac{}{[\varphi[x \mapsto e]]\; x := e\; [\varphi]}$$

$$\frac{}{[\varphi]\; \mathbf{skip}\; [\varphi]}$$

$$\frac{[\varphi]\; S_1\; [\theta] \qquad [\theta]\; S_2\; [\psi]}{[\varphi]\; S_1; S_2\; [\psi]}$$

$$\frac{[\varphi \wedge b]\; S_1\; [\psi] \qquad [\varphi \wedge \neg b]\; S_2\; [\psi]}{[\varphi]\; \mathbf{if}\; b\; \mathbf{then}\; S_1\; \mathbf{else}\; S_2\; [\psi]}$$

$$\frac{???}{[???]\; \mathbf{while}\; b\; \mathbf{do}\; S\; [???]}$$

$$\frac{\varphi' \Rightarrow \varphi \qquad [\varphi]\; S\; [\psi] \qquad \psi \Rightarrow \psi'}{[\varphi']\; S\; [\psi']}$$

Adjustments are necessary if expressions may generate errors!

Total-correctness rule for loops

$$\frac{(nat(l) \wedge \varphi(l+1)) \Rightarrow b \qquad [nat(l) \wedge \varphi(l+1)]\ S\ [\varphi(l)] \qquad \varphi(0) \Rightarrow \neg b}{[\exists l.nat(l) \wedge \varphi(l)]\ \mathbf{while}\ b\ \mathbf{do}\ S\ [\varphi(0)]}$$

where

- $\varphi(l)$ is a formula with a free variable l that does not occur in $\mathbf{while}\ b\ \mathbf{do}\ S$,
- $nat(l)$ stands for $0 \leq l$, and
- $\varphi(l+1)$ and $\varphi(0)$ result by substituting, respectively, $l+1$ and 0 for l in $\varphi(l)$.

Informally: l is a *counter* that indicates the number of iterations of the loop body

Soundness

(of the proof rules for total correctness for the statements of TINY)

$$\boxed{\text{if } \boxed{\mathcal{TH}(\mathbf{Int}) \vdash [\varphi]\, S\, [\psi]} \text{ then } \boxed{\models [\varphi]\, S\, [\psi]}}$$

Proof: By induction on the structure of the proof tree: all the cases are as for partial correctness, except for the rule for loops.

loop rule: Consider $s \in \{nat(l) \wedge \varphi(l)\}$. By induction on $s(l)$ (which is a natural number) show that $\mathcal{S}[\![\mathbf{while}\ b\ \mathbf{do}\ S]\!]\, s = s'$ for some $s' \in \{\varphi(0)\}$ (easy!). To complete the proof, notice that if a variable x does not occur in a statement $S' \in \mathbf{Stmt}$ and two states differ at most on x, then whenever S' terminates successfully starting in one of them, then so it does starting in the other, and the result states differ at most on x.

Completeness

(of the proof system for total correctness for the statements of $\textsc{Tiny}$**)**

It so happens that:

$$\boxed{\mathcal{TH}(\mathbf{Int}) \vdash [\varphi]\, S\, [\psi]} \quad \text{iff} \quad \boxed{\models [\varphi]\, S\, [\psi]}$$

Proof (idea): Only loops cause extra problems: here, for $\varphi(l)$ take the conjunction of the (partial correctness) loop invariant with the formula

"the loop terminates in exactly l iterations"

It so happens that the latter can indeed be expressed here (since finite tuples of integers and their finite sequences can be coded as natural numbers)!

For example

To prove:

$$[n \geq 0 \wedge rt = 0 \wedge sqr = 1]$$
$$\quad \textbf{while } sqr \leq n \textbf{ do}$$
$$\qquad rt := rt + 1; sqr := sqr + 2 * rt + 1$$
$$[rt^2 \leq n \wedge n < (rt + 1)^2]$$

use the following invariant with the iteration counter l:

$$sqr = (rt + 1)^2 \wedge rt^2 \leq n \wedge l = \lfloor \sqrt{n} \rfloor - rt$$

Cheating here, of course:

"$l = \lfloor \sqrt{n} \rfloor - rt$" has to be captured by a first-order formula in the language of TINY

Luckily: this can be done!

Here, this is quite easy:
$(rt + l)^2 \leq n < (rt + l + 1)^2$

Well-founded relations

A relation $\succ \subseteq W \times W$ is *well-founded* if there is no infinite chain

$$a_0 \succ a_1 \succ \ldots \succ a_i \succ a_{i+1} \succ \ldots$$

Typical example:

$\langle \mathbf{Nat}, > \rangle$

BTW: For well-founded $\succ \subseteq W \times W$, its transitive and reflexive closure $\succ^* \subseteq W \times W$ is a partial order on W.
BUT: subtracting identity from an arbitrary partial order on W need not in general yield a well-founded relation.

Few other examples:

- $\mathbf{Nat}^n$ with component-wise (strict) ordering;
- A^* with proper prefix ordering;
- $\mathbf{Nat}^n$ with lexicographic (strict) ordering generated by the usual ordering on $\mathbf{Nat}$;
- any ordinal with the natural (strict) ordering; etc.

Total correctness $=$ partial correctness $+$ successful termination

Proof method

To prove

$$[\varphi]\ \mathbf{while}\ b\ \mathbf{do}\ S\ [\varphi \wedge \neg b]$$

- show "partial correctness": $[\varphi \wedge b]\ S\ [\varphi]$
- show "termination": find a set W with a well-founded relation $\succ \subseteq W \times W$ and a function $w \colon \mathbf{State} \to W$ such that for all states $s \in \{\varphi \wedge b\}$,

$$w(s) \succ w(\mathcal{S}[\![S]\!]\, s)$$

BTW: $w \colon \mathbf{State} \rightharpoonup W$ may be partial as long as it is defined on $\{\varphi\}$.

Example

Prove:

$$
\begin{array}{l}
[x \geq 0 \wedge y \geq 0] \\
\quad \textbf{while } x > 0 \textbf{ do} \\
\qquad\quad \textbf{if } y > 0 \textbf{ then } y := y - 1 \textbf{ else } (x := x - 1; y := f(x)) \\
[\textbf{true}]
\end{array}
$$

where f yields a natural number for any natural argument.

- If one knows nothing more about f, then the previous proof rule for the total correctness of loops is useless here.
- BUT: termination can be proved easily using the function $w \colon \textbf{State} \rightarrow \textbf{Nat} \times \textbf{Nat}$, where $w(s) = \langle s\, x, s\, y \rangle$:
  after each iteration of the loop body the value of w decreases w.r.t. the (well-founded) lexicographic order on pairs of natural numbers.

A fully specified program

$$[x \geq 0 \wedge y \geq 0]$$

$$\textbf{while } [x \geq 0 \wedge y \geq 0]\ x > 0 \textbf{ do } \textbf{decr } \langle x, y \rangle \textbf{ in } \mathbf{Nat} \times \mathbf{Nat} \textbf{ wrt } \succ$$

$$\quad \textbf{if } y > 0 \textbf{ then } y := y - 1 \textbf{ else } (x := x - 1; y := f(x))$$

$$[\mathbf{true}]$$

. . . with various notational variants
assuming some external definitions for
the well-founded set and function into it

Hoare's logic: trouble #2

Find S such that

$$\{n \geq 0\}\, S\, \{rt^2 \leq n \wedge n < (rt+1)^2\}$$

Another correct solution:

$$\begin{array}{l} \{n \geq 0\} \\ \quad rt := 0;\, n := 0 \\ \{rt^2 \leq n \wedge n < (rt+1)^2\} \end{array}$$

OOOOPS?!

A number of techniques to avoid this:

- variables that are required not to be used in the program;
- binary postconditions;
- various forms of algorithmic/dynamic logic, with program modalities.

Binary postconditions

Sketch

- New syntactic category $\boxed{\mathbf{BForm}}$ of *binary formulae*, which are like the usual formulae, except they can use both the usual variables $x \in \mathbf{Var}$ and their "past" copies $\widehat{x} \in \widehat{\mathbf{Var}}$.
  For any syntactic item ω, we write $\widehat{\omega}$ for ω with each variable x replaced by $\widehat{x}$.
- Semantic function: $\boxed{\mathcal{BF} : \mathbf{BForm} \to \mathbf{State} \times \mathbf{State} \to \mathbf{Bool}}$
  $\mathcal{BF}[\![\psi]\!]\,\langle s_0, s\rangle$ is defined as usual, except that the state s_0 is used to evaluate "past" variables $\widehat{x} \in \widehat{\mathbf{Var}}$ and s is used to evaluate the usual variables $x \in \mathbf{Var}$.

Correctness judgements

$$pre\,\varphi;\ S\ post\,\psi$$

where $\varphi \in \mathbf{Form}$ is a (unary) precondition; $S \in \mathbf{Stmt}$ is a statement (as usual); and $\psi \in \mathbf{BForm}$ is a binary postcondition.

Semantics:

The judgement $pre\,\varphi;\ S\ post\,\psi$ holds, written $\models pre\,\varphi;\ S\ post\,\psi$, if for all states $s \in \mathbf{State}$

$$\text{if } \mathcal{F}[\![\varphi]\!]\,s = \mathbf{tt} \text{ then } \mathcal{S}[\![S]\!]\,s \in \mathbf{State} \text{ and } \mathcal{BF}[\![\psi]\!]\,\langle s, \mathcal{S}[\![S]\!]\,s\rangle = \mathbf{tt}$$

Proof rules

$$\frac{}{pre\,\varphi;\ x := e\ post\,(\widehat{\varphi} \wedge x = \widehat{e} \wedge \vec{y} = \widehat{\vec{y}})}$$

where $\vec{y}$ are variables other than x.

$$\frac{}{pre\,\varphi;\ \mathbf{skip}\ post\,(\varphi \wedge \vec{y} = \widehat{\vec{y}})}$$

$$\frac{pre\,\varphi_1;\ S_1\ post\,(\psi_1 \wedge \varphi_2) \qquad pre\,\varphi_2;\ S_2\ post\,\psi_2}{pre\,\varphi_1;\ S_1; S_2\ post\,\psi_1 * \psi_2}$$

where $\psi_1 * \psi_2$ is $\exists \vec{z}.(\psi_1[\vec{x} \mapsto \vec{z}] \wedge \psi_2[\widehat{\vec{x}} \mapsto \vec{z}])$, with all the variables free in ψ_1 or ψ_2 are among $\vec{x}$ or $\widehat{\vec{x}}$, and $\vec{z}$ are new variables.

Further rules

$$\frac{pre\, \varphi \wedge b;\ S_1\ post\, \psi \qquad pre\, \varphi \wedge \neg b;\ S_2\ post\, \psi}{pre\, \varphi;\ \mathbf{if}\ b\ \mathbf{then}\ S_1\ \mathbf{else}\ S_2\ post\, \psi}$$

$$\frac{pre\, \varphi \wedge b;\ S\ post\, (\psi \wedge \widehat{e} \succ e) \qquad \psi \Rightarrow \varphi \qquad (\psi * \psi) \Rightarrow \psi}{pre\, \varphi;\ \mathbf{while}\ b\ \mathbf{do}\ S\ post\, ((\psi \vee (\varphi \wedge \vec{y} = \widehat{\vec{y}})) \wedge \neg b)}$$

where $\succ$ is well-founded, and all the free variables are among $\vec{y}$ or $\widehat{\vec{y}}$.

$$\frac{\varphi' \Rightarrow \varphi \quad pre\, \varphi;\ S\ post\, \psi \quad \psi \Rightarrow \psi'}{pre\, \varphi';\ S\ post\, \psi'}$$

$$\frac{pre\, \varphi;\ S\ post\, \psi}{pre\, \varphi;\ S\ post\, (\widehat{\varphi} \wedge \psi)}$$

The rules can (have to?) be polished...

Example

We have now:

$$\models \begin{array}{l} \textit{pre } n \geq 0; \\ \quad rt := 0; sqr := 1; \\ \quad \textbf{while } sqr \leq n \textbf{ do } rt := rt + 1; sqr := sqr + 2 * rt + 1 \\ \textit{post } rt^2 \leq \widehat{n} \wedge \widehat{n} < (rt+1)^2 \end{array}$$

$$BUT: \quad \not\models \begin{array}{l} \{n \geq 0\} \\ \quad rt := 0; n := 0 \\ \{rt^2 \leq \widehat{n} \wedge \widehat{n} < (rt+1)^2\} \end{array}$$

Algorithmic/dynamic logic

Sketch

- Salwicki 1970
- Pratt 1974, Harel 1976
- many others to follow (see Harel, Kozen & Tiuryn 2000)

Overall idea:

Extend the logical formulae so that they are closed under the usual logical connectives and quantification, as well as under program modalities

Syntax: For any formula φ and a statement $S \in \mathbf{Stmt}$, build a new formula:

$$\boxed{\langle S \rangle \varphi}$$

Semantics: $\mathcal{F}[\![\langle S \rangle \varphi]\!]\, s = \begin{cases} \mathcal{F}[\![\varphi]\!]\, s' & \text{if } \mathcal{S}[\![S]\!]\, s = s' \in \mathbf{State} \\ \mathbf{ff} & \text{if } \mathcal{S}[\![S]\!]\, s \notin \mathbf{State} \end{cases}$

Proof system

. . . axioms and rules to handle the standard connectives and quantification . . .

Plus axioms and rules to deal with program modalities — interaction between modalities and propositional connectives; (de)composition of modalities — for instance:

$$\langle S\rangle(\varphi \wedge \psi) \iff (\langle S\rangle\varphi \wedge \langle S\rangle\psi)$$

$$\langle S\rangle\neg\varphi \implies \neg\langle S\rangle\varphi$$

$$\langle S\rangle\mathbf{true} \implies (\neg\langle S\rangle\varphi \implies \langle S\rangle\neg\varphi)$$

$$\langle S_1; S_2\rangle\varphi \iff \langle S_1\rangle(\langle S_2\rangle\varphi)$$

etc.

Key to the completeness results here: *infinitary rules for loops*