Semantyka i weryfikacja programów

Bartosz Klin

(slajdy Andrzeja Tarleckiego)

Instytut Informatyki
Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski

http://www.mimuw.edu.pl/~klin pok. 5680
klin@mimuw.edu.pl

Strona tego wykładu:

http://www.mimuw.edu.pl/~klin/sem18-19.html

Program Semantics & Verification

Bartosz Klin

(slides courtesy of Andrzej Tarlecki)

Institute of Informatics
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw

http://www.mimuw.edu.pl/~klin office: 5680
klin@mimuw.edu.pl

This course: http://www.mimuw.edu.pl/~klin/sem18-19.html

Program correctness and verification

Programs should be:

- *clear; efficient; robust; reliable; user friendly; well documented; ...*
- *but first of all, CORRECT*
- *don't forget though: also, executable...*

Correctness

Program correctness makes sense only

w.r.t. a precise *specification* of the requirements.

Defining correctness

We need:

- A formal definition of the programs in use

  syntax and semantics of the programming language

- A formal definition of the specifications in use

  syntax and semantics of the specification formalism

- A formal definition of the notion of correctness to be used

  what does it mean for a program to satisfy a specification

Proving correctness

We need:

- A formal system to prove correctness of programs w.r.t. specifications

  a logical calculus to prove judgments of program correctness

- A (meta-)proof that the logic proves only true correctness judgements

  soundness of the logical calculus

- A (meta-)proof that the logic proves all true correctness judgements

  completeness of the logical calculus

  under acceptable technical conditions

A specified program

$$\{n \geq 0\}$$

$$rt := 0;\ sqr := 1;$$

$$\textbf{while}\ sqr \leq n\ \textbf{do}$$

$$\qquad (rt := rt + 1;\ sqr := sqr + 2 * rt + 1)$$

$$\{rt^2 \leq n < (rt + 1)^2\}$$

If we start with a non-negative n, and execute the program successfully, then we end up with rt holding the integer square root of n

Hoare's logic

Correctness judgements:

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

- S is a statement of TINY
- the *precondition* φ and the *postcondition* ψ are first-order formulae with variables in **Var**

Intended meaning:

Partial correctness:
termination not guaranteed!

Whenever the program S starts in a state satisfying the precondtion φ and terminates successfully, then the final state satisfies the postcondition ψ

Formal definition

Recall the simplest semantics of TINY, with $\boxed{\mathcal{S} : \mathbf{Stmt} \to \mathbf{State} \rightharpoonup \mathbf{State}}$

We add now a new syntactic category:

$$\boxed{\varphi \in \mathbf{Form} ::= b \mid \varphi_1 \land \varphi_2 \mid \varphi_1 \Rightarrow \varphi_2 \mid \neg\varphi' \mid \exists x.\varphi' \mid \forall x.\varphi'}$$

with the corresponding semantic function:

$$\boxed{\mathcal{F} : \mathbf{Form} \to \mathbf{State} \to \mathbf{Bool}}$$

and standard semantic clauses.

Also, the usual definitions of *free variables* of a formula and *substitution* of an expression for a variable

More notation

For $\varphi \in \mathbf{Form}$:

$$\boxed{\{\varphi\} = \{s \in \mathbf{State} \mid \mathcal{F}[\![\varphi]\!]\, s = \mathbf{tt}\}}$$

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

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

Hoare's logic: semantics

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

iff

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

Spelling this out:

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

if $\mathcal{F}[\![\varphi]\!]\; s = \mathbf{tt}$ and $\mathcal{S}[\![S]\!]\; s \in \mathbf{State}$

then $\mathcal{F}[\![\psi]\!]\; (\mathcal{S}[\![S]\!]\; s) = \mathbf{tt}$

Hoare's logic: proof rules

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

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

$$\frac{\{\varphi\}\; S_1\; \{\theta\} \quad \{\theta\}\; S_2\; \{\psi\}}{\{\varphi\}\; S_1; S_2\; \{\psi\}}$$

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

$$\frac{\{\varphi \wedge b\}\; S\; \{\varphi\}}{\{\varphi\}\; \mathbf{while}\ b\ \mathbf{do}\ S\; \{\varphi \wedge \neg b\}}$$

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

Example of a proof

We will prove the following partial correctness judgement:

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

Consequence rule will be used implicitly
to replace assertions by equivalent ones of a simpler form

Step by step

- $\{n \geq 0\}\; rt := 0\; \{n \geq 0 \wedge rt = 0\}$
- $\{n \geq 0 \wedge rt = 0\}\; sqr := 1\; \{n \geq 0 \wedge rt = 0 \wedge sqr = 1\}$
- $\{n \geq 0\}\; rt := 0;\, sqr := 1\; \{n \geq 0 \wedge rt = 0 \wedge sqr = 1\}$
- $\boxed{\{n \geq 0\}\; rt := 0;\, sqr := 1\; \{sqr = (rt+1)^2 \wedge rt^2 \leq n\}}$

EUREKA!!!
We have just invented
the *loop invariant*

Loop invariant

- $\{(sqr = (rt+1)^2 \wedge rt^2 \leq n) \wedge sqr \leq n\}\ rt := rt + 1\ \{sqr = rt^2 \wedge sqr \leq n\}$
- $\{sqr = rt^2 \wedge sqr \leq n\}\ sqr := sqr + 2 * rt + 1\ \{sqr = (rt+1)^2 \wedge rt^2 \leq n\}$
- $\{(sqr = (rt+1)^2 \wedge rt^2 \leq n) \wedge sqr \leq n\}$
  $rt := rt + 1; sqr := sqr + 2 * rt + 1$
  $\{sqr = (rt+1)^2 \wedge rt^2 \leq n\}$
- $\{sqr = (rt+1)^2 \wedge rt^2 \leq n\}$
  while $sqr \leq n$ **do**
  $rt := rt + 1; sqr := sqr + 2 * rt + 1$
  $\{(sqr = (rt+1)^2 \wedge rt^2 \leq n) \wedge \neg(sqr \leq n)\}$

Finishing up

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

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

QED

A fully specified program

$\{n \geq 0\}$

$rt := 0;$

$\{n \geq 0 \wedge rt = 0\}$

$sqr := 1;$

$\{n \geq 0 \wedge rt = 0 \wedge sqr = 1\}$

$\textbf{while}\ \{sqr = (rt+1)^2 \wedge rt^2 \leq n\}\ sqr \leq n\ \textbf{do}$

$\quad rt := rt + 1;$

$\quad \{sqr = rt^2 \wedge sqr \leq n\}$

$\quad sqr := sqr + 2 * rt + 1$

$\{rt^2 \leq n < (rt+1)^2\}$

The first-order theory in use

In the proof above, we have used quite a number of facts concerning the underlying data type, that is, **Int** with the operations and relations built into the syntax of TINY. Indeed, each use of the consequence rule requires such facts.

Define the *theory* of **Int**

$$\boxed{\mathcal{TH}(\mathbf{Int})}$$

to be the set of all formulae that hold in all states.

The above proof shows:

$$\mathcal{TH}(\mathbf{Int}) \vdash \begin{array}{l} \{n \geq 0\} \\ \quad rt := 0; sqr := 1; \\ \quad \mathbf{while}\ sqr \leq n\ \mathbf{do}\ rt := rt + 1; sqr := sqr + 2 * rt + 1 \\ \{rt^2 \leq n \wedge n < (rt+1)^2\} \end{array}$$

Soundness

Fact: *Hoare's proof calculus (given by the above rules) is sound, that is:*

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

So, the above proof of a correctness judgement validates the following semantic fact:

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

Proof

(of soundness of Hoare's proof calculus)

By induction on the structure of the proof in Hoare's logic:

assignment rule: Easy, but we need a lemma (to be proved by induction on the structure of formulae):

$$\mathcal{F}[\![\varphi[x \mapsto e]]\!]\ s = \mathcal{F}[\![\varphi]\!]\ s[x \mapsto \mathcal{E}[\![e]\!]\ s]$$

Then, for $s \in \mathbf{State}$, if $s \in \{\varphi[x \mapsto e]\}$ then $\mathcal{S}[\![x := e]\!]\ s = s[x \mapsto \mathcal{E}[\![e]\!]\ s] \in \{\varphi\}$.

skip rule: Trivial.

composition rule: Assume $\{\varphi\}\ [\![S_1]\!] \subseteq \{\theta\}$ and $\{\theta\}\ [\![S_2]\!] \subseteq \{\psi\}$. Then $\{\varphi\}\ [\![S_1; S_2]\!] = (\{\varphi\}\ [\![S_1]\!])\ [\![S_2]\!] \subseteq \{\theta\}\ [\![S_2]\!] \subseteq \{\psi\}$.

if-then-else rule: Easy.

consequence rule: Again the same, given the obvious observation that $\{\varphi_1\} \subseteq \{\varphi_2\}$ iff $\varphi_1 \Rightarrow \varphi_2 \in \mathcal{TH}(\mathbf{Int})$.

Soundness of the loop rule

loop rule: We need to show that the least fixed point of the operator

$$\Phi(F) = cond(\mathcal{B}[\![b]\!], \mathcal{S}[\![S]\!];F, id_{\mathbf{State}})$$

satisfies

$$fix(\Phi)(\{\varphi\}) \subseteq \{\varphi \wedge \neg b\}$$

Proceed by *fixed point induction*. Suppose that $F(\{\varphi\}) \subseteq \{\varphi \wedge \neg b\}$ for some $F\colon \mathbf{State} \rightharpoonup \mathbf{State}$, and consider $s \in \{\varphi\}$ with $s' = \Phi(F)(s) \in \mathbf{State}$. Two cases are possible:

- If $\mathcal{B}[\![b]\!]\ s = \mathbf{ff}$ then $s' = s \in \{\varphi \wedge \neg b\}$.
- If $\mathcal{B}[\![b]\!]\ s = \mathbf{tt}$ then $s' = F(\mathcal{S}[\![S]\!]\ s)$. We get $s' \in \{\varphi \wedge \neg b\}$ by the assumption on F, since $\{\varphi \wedge b\}\ [\![S]\!] \subseteq \{\varphi\}$ by the assumption on S, which implies $\mathcal{S}[\![S]\!]\ s \in \{\varphi\}$.

So, $\Phi(F)(\{\varphi\}) \subseteq \{\varphi \wedge \neg b\}$, and the proof is completed.

Problems with completeness

- If $\mathcal{T} \subseteq \mathbf{Form}$ is r.e. then the set of all Hoare's triples derivable from $\mathcal{T}$ is r.e. as well.
- $\models \{\mathbf{true}\}\, S\, \{\mathbf{false}\}$ iff S loops for all initial states.
- Since the halting problem is not decidable for TINY, the set of all judgements of the form $\{\mathbf{true}\}\, S\, \{\mathbf{false}\}$ such that $\models \{\mathbf{true}\}\, S\, \{\mathbf{false}\}$ is not r.e.

Nevertheless:

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

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:

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

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

$$\text{Total correctness} = \text{partial correctness} + \text{successful termination}$$

Total correctness judgements:

$$[\varphi]\ S\ [\psi]$$

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)

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

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:

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

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:

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

use the following invariant with the iteration counter l:

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

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:

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

A 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

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

- 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 $s \in \{\varphi \wedge b\}$,

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

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 $w \colon \textbf{State} \to \textbf{Nat} \times \textbf{Nat}$, 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

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

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

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

$[\mathbf{true}]$

. . . 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

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

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

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

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

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

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

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

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

$$\frac{\varphi' \Rightarrow \varphi \qquad \mathit{pre}\ \varphi;\ S\ \mathit{post}\ \psi \qquad \psi \Rightarrow \psi'}{\mathit{pre}\ \varphi';\ S\ \mathit{post}\ \psi'}$$

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

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

Example

We have now:

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

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