### **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



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

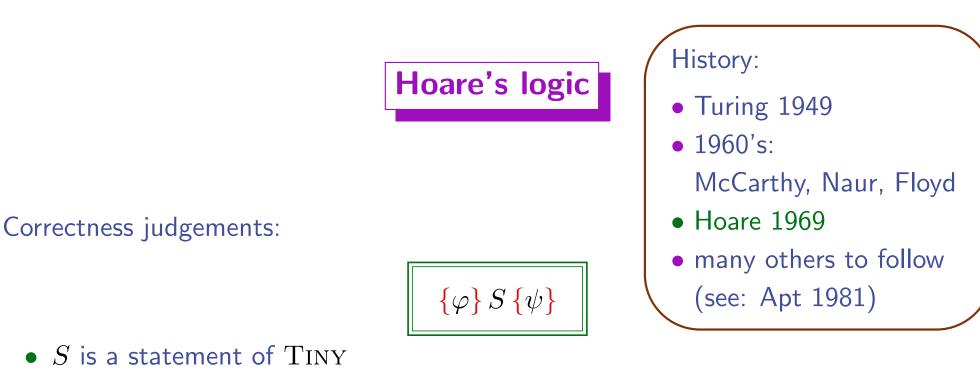
completeness of the logical calculus

under acceptable technical conditions

A specified program

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

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



• the precondition  $\varphi$  and the postcondition  $\psi$  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  $\varphi$ 

and terminates successfully, then the final state satisfies the postcondition  $\psi$ 

### Formal definition

Recall the simplest semantics of  $T\ensuremath{\text{INY}}$  , with

$$\mathcal{S} \colon \mathbf{Stmt} \to \mathbf{State} \rightharpoonup \mathbf{State}$$

We add now a new syntactic category:

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

with the corresponding semantic function:

 $\mathcal{F}\colon \mathbf{\overline{Form}} \to \mathbf{State} \to \mathbf{Bool}$ 

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

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

For  $S \in$ **Stmt**,  $A \subseteq$  **State**:

 $A \llbracket S \rrbracket = \{ s \in \mathbf{State} \mid S \llbracket S \rrbracket a = s, \text{for some } a \in A \}$ 

Hoare's logic: semantics

$$ert arphi \left\{arphi
ight\} S\left\{\psi
ight\}$$
 iff  $\left\{arphi
ight\} \left[\!\left[S
ight]\!\right] \subseteq \left\{\psi
ight\}$ 

Spelling this out:

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

 $\begin{array}{l} \text{if } \mathcal{F}\llbracket \varphi \rrbracket s = \mathbf{tt} \text{ and } \mathcal{S}\llbracket S \rrbracket s \in \mathbf{State} \\ \\ \text{then } \mathcal{F}\llbracket \psi \rrbracket \left( \mathcal{S}\llbracket S \rrbracket s \right) = \mathbf{tt} \end{array} \end{array}$ 

## Hoare's logic: proof rules

## Example of a proof

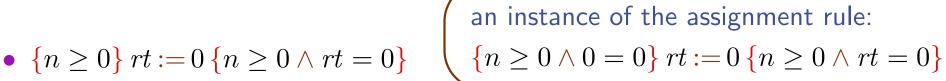
We will prove the following partial correctness judgement:

```
 \begin{split} &\{n \geq 0\} \\ &rt := 0; \\ &sqr := 1; \\ & \textbf{while } sqr \leq n \textbf{ do} \\ &rt := rt + 1; \\ &sqr := sqr + 2 * rt + 1 \\ &\{rt^2 \leq n \wedge n < (rt + 1)^2\} \end{split}
```

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

### Step by step

$$\overline{\{\varphi[x\mapsto e]\}\,x\!:=\!e\,\{\varphi\}}$$



• 
$$\{n \ge 0 \land rt = 0\}$$
  $sqr := 1$   $\{n \ge 0 \land rt = 0 \land sqr = 1\}$ 

• 
$$\{n \ge 0\}$$
  $rt := 0; sqr := 1 \{n \ge 0 \land rt = 0 \land sqr = 1\}$ 

$$\{n \ge 0\} \ rt := 0; \ sqr := 1 \ \{n \ge 0 \land rt = 0 \land sqr = 1\}$$

$$\{\varphi\} \ S_1 \ \{\theta\} \ S_2 \ \{\psi\} \ \{\varphi\} \ S_1; \ S_2 \ \{\psi\} \ S_2 \ S_1; \ S_2 \ \{\psi\} \ S_2 \ S$$

We have just invented the loop invariant

BTW: another version of the assignment rule:

$$\{\varphi\} x := e \{\exists x' . (\varphi[x \mapsto x'] \land x = e[x \mapsto x'])\}$$

### Loop invariant

an instance of the assignment rule:

$$\{sqr = (rt+1)^2 \land sqr \le n \} rt := rt + 1 \{sqr = rt^2 \land sqr \le n \}$$

• 
$$\{(sqr = (rt+1)^2 \land rt^2 \le n) \land sqr \le n\} rt := rt + 1 \{sqr = rt^2 \land sqr \le n\}$$

• 
$$\{sqr = rt^2 \land sqr \le n\} \ sqr := sqr + 2 * rt + 1 \ \{sqr = (rt+1)^2 \land rt^2 \le n\}$$

• 
$$\{(sqr = (rt + 1)^2 \land rt^2 \le n) \land sqr \le n\}$$
  
 $rt := rt + 1; sqr := sqr + 2 * rt + 1$   
 $\{sqr = (rt + 1)^2 \land rt^2 \le n\}$ 

• 
$$\begin{cases} sqr = (rt+1)^2 \wedge rt^2 \leq n \\ \text{while } sqr \leq n \text{ do} \\ rt := rt+1; sqr := sqr+2 * rt+1 \\ \{(sqr = (rt+1)^2 \wedge rt^2 \leq n) \wedge \neg (sqr \leq n)\} \end{cases}$$

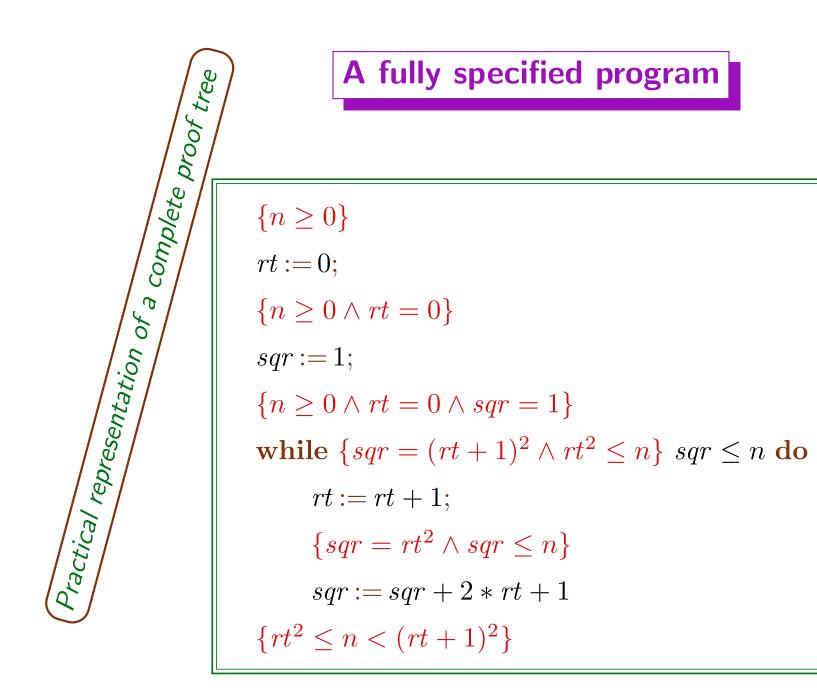
$$\frac{\left\{\varphi \wedge b\right\}S\left\{\varphi\right\}}{\left\{\varphi\right\} \textbf{while } b \textbf{ do } S\left\{\varphi \wedge \neg b\right\}}$$

# Finishing up

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

$$\begin{cases} n \ge 0 \\ rt := 0; sqr := 1; \\ \text{while } sqr \le n \text{ do} \\ rt := rt + 1; sqr := sqr + 2 * rt + 1 \\ \{rt^2 \le n \land n < (rt + 1)^2\} \end{cases}$$

QED



### 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

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

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

The above proof shows:

 $\mathcal{TH}(\mathbf{Int}) \vdash \begin{cases} n \ge 0 \\ rt := 0; sqr := 1; \\ \mathbf{while} \ sqr \le n \ \mathbf{do} \ rt := rt + 1; sqr := sqr + 2 * rt + 1 \\ \{rt^2 \le n \land n < (rt + 1)^2\} \end{cases}$ 

## Soundness

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

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

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

$$\models \begin{cases} \{n \ge 0\} \\ rt := 0; \, sqr := 1; \\ \text{while } sqr \le n \text{ do } rt := rt + 1; \, sqr := sqr + 2 * rt + 1 \\ \{rt^2 \le n \land n < (rt + 1)^2\} \end{cases}$$



#### (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):

$$\mathcal{F}\llbracket\varphi[x\mapsto e]\rrbracket s = \mathcal{F}\llbracket\varphi\rrbracket s[x\mapsto \mathcal{E}\llbrackete\rrbracket s] \underbrace{\{\varphi[x\mapsto e]\} x := e\{\varphi\}}_{x \mapsto e}$$

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

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

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

$$\Phi(F) = cond(\mathcal{B}\llbracket b \rrbracket, \mathcal{S}\llbracket S \rrbracket; F, id_{\mathbf{State}})$$

satisfies

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

$$\frac{\{\varphi \land b\} S \{\varphi\}}{\{\varphi\} \text{ while } b \text{ do } S \{\varphi \land \neg b\}}$$

Proceed by fixed point induction (*this is an admissible property*!). Suppose that  $F(\{\varphi\}) \subseteq \{\varphi \land \neg b\}$  for some F:State  $\rightarrow$  State, and consider  $s \in \{\varphi\}$  with  $s' = \Phi(F)(s) \in$ State. Two cases are possible:

- If  $\mathcal{B}\llbracket b \rrbracket s = \mathbf{ff}$  then  $s' = s \in \{\varphi \land \neg b\}$ .
- If B[[b]] s = tt then s' = F(S[[S]] s). We get s' ∈ {φ ∧ ¬b} by the assumption on F, since {φ ∧ b} [[S]] ⊆ {φ} by the inductive hypothesis, which implies S[[S]] s ∈ {φ}.

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

### **Problems with completeness**

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

Nevertheless:

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