A note on program verification

Peter Aczel

Abstract. The note presents a complete system of proof rules for the total correctness of iterative programs. Central to the note is the use of program specifications that may depend not only on the state of a computation but also on the starting state.

1. Introduction. In [Hoare 1969] a program specification consists of a pair of predicates called the precondition and postcondition of the specification. Each of these predicates depends on the state of a computation and a program meets the specification if any terminating execution of it, starting in a state satisfying the precondition, terminates in a state satisfying the postcondition. For example the program

$$\underline{\text{while}}\ (y+1)^2 \leq x\ \underline{\text{do}}\ y := y+1$$

meets the specification having precondition $y=0 \& x \geq 0$ and postcondition $y^2 \leq x < (y+1)^2$.

It is a familiar fact that this specification does not explicitly express all that we have in mind. For example if the above program is

prefixed by $x := 0$; the resulting program will still formally meet the specification, although the implicit understanding that x is supposed to remain fixed has been violated. One natural convention is to make this understanding explicit by using special symbols for variables that are to remain fixed throughout a computation. But a more flexible and powerful approach has been advocated by Cliff Jones in his book [Jones 1980]. His approach is to allow the postcondition of a specification to depend on the starting state of a computation. So, in our example the postcondition should be

$$(y^2 \leq x < (y+1)^2) \;\&\; (x = \overleftarrow{x})$$

where we use $\overleftarrow{x}$ to denote the value of x at the start of the computation. In his book Cliff Jones presents some rules for proving the total correctness of programs for his notion of specification. His rules appear elaborate and unmemorable compared with the original rules for partial correctness of Hoare. Moreover they are not complete. The aim of this note is to present simple complete proof rules. It will be seen that the use of the more general notion of postcondition actually leads to a surprisingly simple complete rule for the total correctness of iteration.

2. Some terminology. We shall use expressions (in particular predicates) that, in general, depend not only on the state σ of a computation but also on the starting state $\overleftarrow{\sigma}$.

If Q is a predicate we shall make its dependence

on $\overleftarrow{\sigma}$ and σ explicit by writing it $\overleftarrow{\sigma} Q \sigma$. Using this convention we can transfer standard relational terminology to our predicates. In particular the composition $Q_1 | Q_2$ of the two predicates Q_1 and Q_2 can be defined by

$$\overleftarrow{\sigma}\,(Q_1|Q_2)\sigma \equiv \exists \sigma' [\overleftarrow{\sigma} Q_1 \sigma' \,\&\, \sigma' Q_2 \sigma].$$

A predicate Q is transitive if $Q|Q \supset Q$. A finite or infinite sequence $\sigma_0, \sigma_1, \sigma_2, \ldots$ is called a Q-sequence if $\sigma_0 Q \sigma_1 \,\&\, \sigma_1 Q \sigma_2 \,\&\, \ldots$ Q is well-founded if every Q-sequence terminates. Q is twf if it is transitive and well-founded. The reflexive closure Q^r of Q is defined by

$$\overleftarrow{\sigma}\, Q^r \sigma \equiv \overleftarrow{\sigma} Q \sigma \vee \overleftarrow{\sigma} = \sigma$$

We will call an expression pure if it does not depend on the starting state. If E is pure then we write it $E\sigma$, if we wish to make the dependence on σ explicit, and we say that σ satisfies E. For pure E, $\overleftarrow{E}$ is the expression $E\overleftarrow{\sigma}$, i.e. the expression that depends on $\overleftarrow{\sigma}$ in the same way that E depends on σ.

If the state σ is made up of components $\ldots x, y, \ldots$ then we write $\ldots \overleftarrow{x}, \overleftarrow{y}, \ldots$ for the corresponding components of $\overleftarrow{\sigma}$. For each component x I_x is the predicate $\ldots \,\&\, (y = \overleftarrow{y}) \,\&\, \ldots$, the conjunction taken over all components y except x.

3. Program meaning. We take the meaning of a program statement S to be specified by two predicates $S\!\downarrow$ and $\langle S \rangle$. $S\!\downarrow$ is pure and a state satisfies it if every execution of S, starting from that state, must properly terminate. $\overleftarrow{\sigma} \langle S \rangle \sigma$ if S can be executed, starting in

state $\overleftarrow{\sigma}$ so as to properly terminate in state σ.

We shall restrict attention to the elementary programming constructs of assignment, cases, composition and iteration. Their meaning is naturally specified by the following.

Assignment. If S is $x := E$ then

$$S\!\downarrow \;\equiv\; \text{true}$$
$$\langle S \rangle \;\equiv\; (x = \overleftarrow{E} \;\&\; I_x)$$

Cases. If S is if B then S_1 else S_2 then

$$S\!\downarrow \;\equiv\; (B \supset S_1\!\downarrow) \;\&\; (\neg B \supset S_2\!\downarrow)$$
$$\langle S \rangle \;\equiv\; (\overleftarrow{B} \;\&\; \langle S_1 \rangle) \vee (\neg \overleftarrow{B} \;\&\; \langle S_2 \rangle)$$

Composition. If S is $S_1 ; S_2$ then

$$S\!\downarrow\!\sigma \;\equiv\; S_1\!\downarrow\!\sigma \;\&\; \forall \sigma' (\sigma \langle S_1 \rangle \sigma' \supset S_2\!\downarrow\!\sigma')$$
$$\langle S \rangle \;\equiv\; \langle S_1 \rangle \,|\, \langle S_2 \rangle$$

Iteration. If S is while B do S_1 then

$$S\!\downarrow\!\sigma \;\equiv\; \text{Every } (\overleftarrow{B} \;\&\; \langle S_1 \rangle)\text{-sequence } \sigma, \ldots \text{ terminates, and every term of it satisfies } B \supset S_1\!\downarrow.$$
$$\overleftarrow{\sigma} \langle S \rangle \sigma \;\equiv\; \neg B\sigma \text{ and there is a } (\overleftarrow{B} \;\&\; \langle S_1 \rangle)\text{-sequence } \overleftarrow{\sigma}, \ldots, \sigma.$$

4. The rules. A specification $\{P\}?\{Q\}$ consists of a pure predicate P and a, not necessarily pure, predicate Q. A program statement S meets $\{P\}?\{Q\}$, written $\{P\}S\{Q\}$, if

(1) $P \supset S\downarrow$,

and (2) $\overleftarrow{P} \,\&\, \langle S \rangle \supset Q$.

Note that this is a notion of total correctness. (2) alone gives partial correctness.

We are now ready to list our rules. There are two general rules and one rule for each of the elementary constructs.

General.

$$(1)\quad \frac{P \supset P' \quad \{P'\}S\{Q'\} \quad Q' \supset Q}{\{P\}\,S\,\{Q\}}$$

$$(2)\quad \frac{\{P\}\,S\,\{Q\}}{\{P\}\,S\,\{\overleftarrow{P}\,\&\,Q\}}$$

Assignment.

$$\{\text{true}\}\ x := E\ \{x = \overleftarrow{E}\,\&\,I_x\}$$

Cases.

$$\frac{\{P\,\&\,B\}\,S_1\,\{Q\} \quad \{P\,\&\,\neg B\}\,S_2\,\{Q\}}{\{P\}\ \underline{\text{if}}\ B\ \underline{\text{then}}\ S_1\ \underline{\text{else}}\ S_2\ \{Q\}}$$

Composition.

$$\frac{\{P\}\,S_1\,\{Q_1\,\&\,P'\} \quad \{P'\}\,S_2\,\{Q_2\}}{\{P\}\ S_1\,;\,S_2\ \{Q_1 \mid Q_2\}}$$

Iteration.

$$\frac{\{P\,\&\,B\}\,S\,\{P\,\&\,R\}}{\{P\}\ \underline{\text{while}}\ B\ \underline{\text{do}}\ S\ \{R^r\,\&\,\neg B\}}\ R \text{ twf}$$

The above rules are both sound and complete. By this we mean that, for any program statement S built up using the four constructs and any specification $\{P\}?\{Q\}$, $\{P\}S\{Q\}$ is true if and only if it can be derived using the rules. These rules can be reformulated so as to incorporate the general rules into each of the remaining rules. The soundness and completeness of the new rules can be expressed by the following.

Theorem.

Assignment. If S is $x:=E$ then $\{P\}S\{Q\}$ iff $\overleftarrow{P} \& (x = \overleftarrow{E} \& I_x) \supset Q$.

Cases. If S is *if* B *then* S_1 *else* S_2 then $\{P\}S\{Q\}$ iff $\{P \& B\}S_1\{Q\}$ and $\{P \& \neg B\}S_2\{Q\}$.

Composition. If S is $S_1; S_2$ then $\{P\}S\{Q\}$ iff for some predicates Q_1, Q_2 and P', where P' is pure,

(1) $\{P\}S_1\{Q_1 \& P'\}$
(2) $\{P'\}S_2\{Q_2\}$
(3) $Q_1 / Q_2 \supset Q$

Iteration. If S is *while* B *do* S_1 then $\{P\}S\{Q\}$ iff for some pure predicate P' and some wff R

(1) $\{P' \& B\}S_1\{R \& P'\}$
(2) $P \supset P'$
(3) $\overleftarrow{P} \& R^{\tau} \& \neg B \supset Q$.

5. Proof of the theorem.

Assignment. Trivial.

Cases.

$$\{P\}\,S\,\{Q\} \equiv (P \supset S\downarrow) \text{ and } (\overleftarrow{P} \,\&\, \langle S\rangle \supset Q)$$

$$\equiv (P \,\&\, B \supset S_1\downarrow) \text{ and } (P \,\&\, \neg B \supset S_2\downarrow)$$
$$\text{and } (\overleftarrow{P} \,\&\, \overleftarrow{B} \,\&\, \langle S_1\rangle \supset Q)$$
$$\text{and } (\overleftarrow{P} \,\&\, \neg\overleftarrow{B} \,\&\, \langle S_2\rangle \supset Q)$$
$$\equiv \{P \,\&\, B\}\,S_1\,\{Q\} \text{ and } \{P \,\&\, \neg B\}\,S_2\,\{Q\}.$$

Composition.

Let $P_0\sigma \equiv \forall\sigma'(\sigma\langle S_1\rangle\sigma' \supset S_2\downarrow\sigma')$. Then

$P \supset P_0 \equiv \overleftarrow{P} \,\&\, \langle S_1\rangle \supset S_2\downarrow$, and

$\overleftarrow{P} \,\&\, (\langle S_1\rangle \,|\, \langle S_2\rangle) \equiv (\overleftarrow{P} \,\&\, \langle S_1\rangle) \,|\, \langle S_2\rangle$. It easily follows that $\{P\}\,S\,\{Q\}$ is equivalent to the conjunction of

I. $P \supset S_1\downarrow$

II. $\overleftarrow{P} \,\&\, \langle S_1\rangle \supset S_2\downarrow$

III. $(\overleftarrow{P} \,\&\, \langle S_1\rangle) \,|\, \langle S_2\rangle \supset Q$.

Now assume (1), (2) and (3). By (1),

$(1)_a$ $P \supset S_1\downarrow$

$(1)_b$ $\overleftarrow{P} \,\&\, \langle S_1\rangle \supset Q_1 \,\&\, P'$

By (2), $(2)_a$ $P' \supset S_2\downarrow$

$(2)_b$ $\overleftarrow{P'} \,\&\, \langle S_2\rangle \supset Q_2$

Now I is just $(1)_a$ and II follows from $(1)_b$ and $(2)_a$.

For III, $(\overleftarrow{P} \,\&\, \prec S_1 \succ) \mid \prec S_2 \succ$

$\supset (Q_1 \,\&\, P') \mid \prec S_2 \succ$, by $(1)_b$,

$\supset Q_1 \mid (\overleftarrow{P'} \,\&\, \prec S_2 \succ)$

$\supset Q_1 \mid Q_2$, by $(2)_b$

$\supset Q$ by (3).

In these implications we have used some simple properties of the composition operation on predicates.

For the converse implication assume I, II and III. Define Q_1, Q_2 and pure P' by

$$Q_1 \equiv \overleftarrow{P} \,\&\, \prec S_1 \succ$$
$$Q_2 \equiv \prec S_2 \succ$$
$$P'\sigma \equiv \exists \sigma' \; \sigma' Q_1 \sigma$$

Proof of (1). $(1)_a$ is I and $(1)_b$ follows from the definitions of Q_1 and P'.

Proof of (2). By II $Q_1 \supset S_2\!\downarrow$ and hence $P' \supset S_2\!\downarrow$, i.e. $(2)_a$. $(2)_b$ follows from the definition of Q_2.

Proof of (3). (3) is III.

Iteration. Assume given pure P' and tot R such that (1), (2) and (3) hold. By (1) we have

$(1)_a$ $P' \supset (B \supset S_1\!\downarrow)$

and $(1)_b$ $\overleftarrow{P' \,\&\, B} \,\&\, \prec S_1 \succ \supset P' \,\&\, R$.

We must show that $\{P\}S\{Q\}$; i.e.

I. $\overleftarrow{P} \supset S\!\downarrow$

and II. $\overleftarrow{P} \,\&\, \prec S \succ \supset Q$.

Let us call a finite or infinite sequence good if it is a $\overleftarrow{B} \,\&\, \prec S_1 \succ$-sequence. Then I. and II. can be re-expressed

I′. Every good sequence starting in a state satisfying P must terminate and every term must satisfy $B \supset S,\downarrow$

II′. If $\overleftarrow{\sigma}, \ldots, \sigma$ is a good sequence such that $P\overleftarrow{\sigma}$ and $\neg B\sigma$ then $\overleftarrow{\sigma} Q \sigma$.

Claim. Every good sequence starting from a state satisfying P is an R-sequence, all of whose terms satisfy P'.

Proof of claim. Let $\sigma_0, \sigma_1, \ldots$ be a good sequence such that $P\sigma_0$. For any two consecutive terms σ_i, σ_{i+1}

$$\sigma_i (\overleftarrow{B} \,\&\, \prec S, \succ) \sigma_{i+1}$$

and hence

$$P'\sigma_i \supset \sigma_i (\overleftarrow{P' \,\&\, B} \,\&\, \prec S, \succ) \sigma_{i+1}.$$

So by $(1)_b$

$$(*) \quad P'\sigma_i \supset \sigma_i (P' \,\&\, R) \sigma_{i+1}.$$

In particular $P'\sigma_i \supset P'\sigma_{i+1}$. But by (2), as $P\sigma_0$, $P'\sigma_0$. It follows that every term of the sequence satisfies P' and hence, by $(*)$, the sequence is an R-sequence.

Proof of I′. Let $\sigma_0, \ldots$ be a good sequence such that $P\sigma_0$. By the claim it is an R-sequence and hence terminates because R is well-founded. Moreover each term satisfies P' and hence $B \supset S,\downarrow$ by $(1)_a$.

Proof of II′. Let $\overleftarrow{\sigma}, \ldots, \sigma$ be a good sequence such that $P\overleftarrow{\sigma}$ and $\neg B\sigma$. Then by the claim it is an R-sequence all of whose terms satisfy P'. As R^r is transitive and reflexive and is implied by R, $\overleftarrow{\sigma} R^r \sigma$. Hence $\overleftarrow{\sigma}(\overleftarrow{P} \,\&\, R \,\&\, \neg B)$ so that by (3) $\overleftarrow{\sigma} Q \sigma$.

Given pure P' and twf R satisfying (1), (2) and (3) we have proved I′ and II′. We now assume I′ and II′ and prove (1), (2) and (3) where we define P' and R by

$P'\sigma \equiv (B \supset S,\downarrow)\sigma$ and there is a good sequence $\ldots, \sigma$ starting in a state satisfying P.

$\overleftarrow{\sigma} R \sigma \equiv P'\sigma$ and there is a good sequence $\overleftarrow{\sigma}, \ldots, \sigma$ of length >1.

Proof that R is twf. Suppose that $\sigma_0 R \sigma_1$ and $\sigma_1 R \sigma_2$. Then there are good sequences $\sigma_0, \ldots, \sigma_1$ and $\sigma_1, \ldots, \sigma_2$ of length >1. It follows that $\sigma_0, \ldots, \sigma_1, \ldots, \sigma_2$ is also a good sequence of length >1. As $\sigma_0 R \sigma_1$ we have $P'\sigma_0$. Hence $\sigma_0 R \sigma_2$ and we have shown that R is transitive. Now let $\sigma_0, \sigma_1, \sigma_2, \ldots$ be an R-sequence. So there are good sequences $(\sigma_0, \ldots, \sigma_1)$, $(\sigma_1, \ldots, \sigma_2), \ldots$, each of length >1. Combining these we get a good sequence $\sigma_0, \ldots, \sigma_1, \ldots, \sigma_2, \ldots$ As $P'\sigma_0$ there is also a good sequence $\sigma_*, \ldots, \sigma_0$ such that σ_* satisfies P. Combining again we get a good sequence $\sigma_*, \ldots, \sigma_0, \ldots, \sigma_1, \ldots, \sigma_2, \ldots$ that starts from a state satisfying P. It follows from I' that this sequence must terminate, and as each component $\sigma_i, \ldots, \sigma_{i+1}$ has length >1 it follows that the R-sequence $\sigma_0, \sigma_1, \sigma_2, \ldots$ must also terminate. So R is well-founded.

Proof of (1). $(1)_a$ follows from the definition of P'. For $(1)_b$ assume that $(P' \& B)\overleftarrow{\sigma}$ and $\overleftarrow{\sigma} \prec S_i \succ \sigma$. Then $S_i \downarrow \overleftarrow{\sigma}$ and there is a good sequence $\sigma_0, \ldots, \overleftarrow{\sigma}$ such that $P\sigma_0$. By the second assumption $\sigma_0, \ldots, \overleftarrow{\sigma}, \sigma$ is also a good sequence. It follows from I' that $(B \supset S_i \downarrow)\sigma$ and hence $P'\sigma$. As $P'\overleftarrow{\sigma}$ and $\overleftarrow{\sigma}, \sigma$ is a good sequence of length 2 we also have $\overleftarrow{\sigma} R \sigma$. Hence $\overleftarrow{\sigma}(P' \& R)\sigma$.

Proof of (2). A sequence of length one is always good so that by I' $P \supset P'$.

Proof of (3). Assume $\overleftarrow{\sigma}(\overleftarrow{P} \& R^* \& \neg B)\sigma$. Then $P\overleftarrow{\sigma}, \neg B\sigma$ and $\overleftarrow{\sigma} R \sigma$ or $\overleftarrow{\sigma} = \sigma$, so that there is a good sequence $\overleftarrow{\sigma}, \ldots, \sigma$. Hence by II' $\overleftarrow{\sigma} Q \sigma$.

Manchester
January 1982