Acta Informatica 26, 409–419 (1989)

Acta Informatica

A Sharp Proof Rule for Procedures in wp Semantics

A. Bijlsma [1], P.A. Matthews [2], and J.G. Wiltink [3]

[1] Department of Computer Science, ETH, P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands
[2] Department of Computer Science, Aarhus University, Ny Munkegade, DK-8000 Aarhus, Denmark
[3] Hildebrandhove 107, NL-2726 AT Zoetermeer, The Netherlands

Summary. A proof rule for the procedure call is proposed that has the property that the precondition it defines is the weakest precondition that can be inferred solely from the procedure's specification. Thus the rule enforces exactly the abstraction introduced by the specification. Gries's proof rule for the procedure call is shown not to have this property in cases when the specification involves so-called specification variables.

Introduction

In specifying a procedure, it is often necessary to refer in the postcondition to the initial values of parameters modified by the procedure. Some authors have suggested using a special notation for the initial value of a modifiable parameter, but as D. Gries has observed, this need can be met in a simpler way by allowing a specification to contain additional variables called *specification variables* or *logical variables*. Such variables are allowed to appear only in the pre- and postconditions of the procedure, not in the program text. They can be used to communicate values between the pre- and postcondition. For example, a procedure for finding the integer square root of a parameter y could be specified using the specification variable m by the precondition

$$m^2 \leqq y < (m+1)^2$$

and the postcondition

$$y = m.$$

The use of specification variables in specifying procedures allows Gries to derive (in [3]) a simple but general proof rule for the call on a procedure. We can describe his proof rule as follows. Say we have a procedure with three

Offprint requests to: P.A. Matthews

parameters: a value parameter x, a value-result parameter y, and a result parameter z. The body of our procedure we will call S and we write

$$S(a, b, c)$$

to mean the call on our procedure with arguments a, b, c corresponding to the parameters x, y, z. Assume in addition that our procedure is specified with precondition U and postcondition V.

A proof rule for the call $S(a, b, c)$ is a method of associating with any predicate E a second predicate that implies $\mathrm{wp}(S(a, b, c), E)$. This second predicate, which we call the *derived precondition*, must depend only on the specification and not on the details of S's construction. Gries's rule states that under certain conditions, to be detailed below, the predicate

$$(\exists m\colon U_{a,b}^{x,y} \wedge (\forall y, z\colon V_a^x \Rightarrow E_{y,z}^{b,c})) \tag{1}$$

is a derived precondition. (Readers wondering where the "$\exists m$" comes from should note that $(\forall m \cdot P \Rightarrow \mathrm{wp}(S, R))$ is the same as $(\exists m \cdot P) \Rightarrow \mathrm{wp}(S, R)$ if m occurs only in P. See also our Remark 2 in [1]).

This rule is currently the best proof rule known; it having supplanted a long list of more limited or even incorrect predecessors. Unfortunately, Gries's rule can, at times, give a precondition for the call that is unnecessarily strong. The following example illustrates this problem.

Consider a procedure for rounding real numbers to a nearby integer. Let the procedure have a real value parameter x and an integer result parameter z. Not needing any stronger property, we state that z is obtained from x by rounding any non-integer x either up or down to an integer, and by using x itself if x happens to be an integer. Using an integer specification variable m, we specify the procedure as follows.

pre $m \leqq x \leqq m+1$
post $z = m \vee z = m+1$

(if x is an integer, then the precondition is satisfied with m equal to either x or $x-1$; this forces z to be x). Many procedure bodies satisfy this specification: "z:=**floor**(x)", "z:=**ceil**(x)", "z:=**round**(x)" are but a few examples.

Now consider the call $S(a, c)$ with postcondition $c=0$. Gries's rule gives the derived precondition *false*, whereas $a=0.0$ is sufficient.

This difficulty with Gries's rule raises the following question. Is it possible to find a rule that is never overly restrictive? Such a rule would associate with each specification and each call postcondition E, a predicate D with the two properties:

(a) $D \Rightarrow \mathrm{wp}(S(a, b, c), E)$ for all procedure bodies that satisfy the specification, and
(b) D is the weakest predicate satisfying property (a).

The first property comes from the definition of a proof rule; the second says that the proof rule is *sharp*.

The answer to this question is yes. We prove in this paper that both properties are satisfied if D is the predicate

$$(\exists m\colon \mathrm{U}^{x,y}_{a,b}) \wedge (\forall y, z\colon (\forall m\colon \mathrm{U}^{x,y}_{a,b} \Rightarrow \mathrm{V}^{x}_{a}) \Rightarrow \mathrm{E}^{b,c}_{y,z}). \tag{2}$$

How does our predicate (2) differ from Gries's (1)? The following calculation shows that (1)$\Rightarrow$(2):

$$\begin{array}{ll}
 & (2) \\
= & \{\text{distribution of } \wedge \text{ over } \exists\} \\
 & (\exists m\colon \mathrm{U}^{x,y}_{a,b} \wedge (\forall y, z\colon (\forall m\colon \mathrm{U}^{x,y}_{a,b} \Rightarrow \mathrm{V}^{x}_{a}) \Rightarrow \mathrm{E}^{b,c}_{y,z})) \\
\Leftarrow & \{\text{instantiation}, (\mathrm{A} \Rightarrow \mathrm{B}) \Rightarrow ((\mathrm{B} \Rightarrow \mathrm{C}) \Rightarrow (\mathrm{A} \Rightarrow \mathrm{C}))\} \\
 & (\exists m\colon \mathrm{U}^{x,y}_{a,b} \wedge (\forall y, z\colon (\mathrm{U}^{x,y}_{a,b} \Rightarrow \mathrm{V}^{x}_{a}) \Rightarrow \mathrm{E}^{b,c}_{y,z})) \\
= & \{\text{from } \mathrm{U}^{x,y}_{a,b} \text{ follows equivalence of } \mathrm{U}^{x,y}_{a,b} \Rightarrow \mathrm{V}^{x}_{a} \text{ and } \mathrm{V}^{x}_{a}\} \\
 & (\exists m\colon \mathrm{U}^{x,y}_{a,b} \wedge (\forall y, z\colon \mathrm{V}^{x}_{a} \Rightarrow \mathrm{E}^{b,c}_{y,z})) \\
= & \{\ \} \\
 & (1)
\end{array}$$

Furthermore, this calculation shows that the two rules coincide when there is at most one value of m satisfying $\mathrm{U}^{x,y}_{a,b}$ and when m does not actually occur in U or V, for then $(\forall m\colon \mathrm{U}^{x,y}_{a,b} \Rightarrow \mathrm{V}^{x}_{a})$ is equivalent to $(\mathrm{U}^{x,y}_{a,b} \Rightarrow \mathrm{V}^{x}_{a})$.

What is the significance of this result? As a tool for proving procedures correct, our rule may not be a significant improvement over Gries's rule. Examples illustrating the difference between the rules seem a bit artifical – at least, the authors have yet to find a convincing one. Instead, we believe the real significance of the result is the knowledge that there is no rule that gives a weaker derived precondition, and that the proof of this fact is fairly short and easily understood. We hope this result will inspire the publication of sharpness proofs for other proof rules.

In order to keep the presentation simple, we discuss only non-recursive calls without global variables. These restrictions, however, are inessential; global variables may be handled just like parameters (see [6] for details), and for recursive calls, a local renaming suffices (see [5], Chap. 4).

Statement of Results

We begin by stating our assumptions about the programming language and procedures.

The Programming Language

We assume the language contains a multi-assignment statement, a conditional statement, and procedures with value, value-result, and result parameters. For concreteness, we express the conditional statement using E.W. Dijkstra's guarded

commands. We assume that every value of the type of a parameter can be denoted by some expression, that such expressions can be compared for equality, and that they can be assigned.

Procedure Bodies

As in the introduction, let our illustrative procedure have value parameter x, value-result parameter y, and result parameter z. Let $\mathscr{B}$ denote the state space consisting of x, y, and z only. A sequence of statements S over $\mathscr{B}$ is a valid body for our procedure if it satisfies the following two conditions (where the square brackets denote universal quantification over all enclosed program and specification variables):

- S is *transparent* to x, which is defined to mean that $[x=v \Rightarrow \mathrm{wlp}(\mathrm{S}, x=v)]$ holds for any constant v of appropriate type, and
- wp(S, Q) is independent of z for any predicate Q, where A independent of z means $[\forall v\colon \mathrm{A}_v^z \equiv \mathrm{A}]$.

Using structural induction, one can prove that the first condition (transparency) is satisfied if S does not contain any assignments to x or procedure calls with x as a result or value-result parameter. Thus transparency serves as a logical formulation of the notion of a value parameter. Similarly, the second condition (independence) serves as the logical formulation of the notion of a result parameter; it is easily seen to be satisfied if S assigns a value to z before referencing it.

Procedure Specifications

We assume that our procedure has an associated precondition U and postcondition V, which are predicates in both the variables of $\mathscr{B}$ and an additional specification variable m, such that U is independent of z. A procedure body S is said to satisfy the specification (U, V) if

$$[\mathrm{U} \Rightarrow \mathrm{wp}(\mathrm{S}, \mathrm{V})].$$

We assume that there is at least one body S_0 that satisfies the specification.

Procedure Calls

Let $\mathscr{C}$ be a state space disjoint from $\mathscr{B}$ and not containing m. Let a be an expression over $\mathscr{C}$, and let b and c be distinct variables of $\mathscr{C}$. The procedure call statement $\mathrm{S}(a, b, c)$ is defined to be equivalent to the following sequence of statements:

$$x, y := a, b; \mathrm{S}; b, c := y, z.$$

(It is allowable to let b and c denote array components rather than simple variables; in that case the multiple assignments should be interpreted in the sense of [4]). Finally, we let E denote the predicate over $\mathscr{C}$ that the call S(a, b, c) should establish.

We are now ready to state our main results. Let D be the predicate

$$(\exists m\colon \mathrm{U}^{x,y}_{a,b}) \wedge (\forall y, z\colon (\forall m\colon \mathrm{U}^{x,y}_{a,b} \Rightarrow \mathrm{V}^{x}_{a}) \Rightarrow \mathrm{E}^{b,c}_{y,z}).$$

Recall the two properties referred to in the introduction. Our first theorem states that D satisfies property (a).

Soundness Theorem. *For all procedure bodies* S *satisfying* [U$\Rightarrow$wp(S, V]), *we have*

$$[\mathrm{D} \Rightarrow \mathrm{wp}(\mathrm{S}(a, b, c), \mathrm{E})].$$

Our second theorem states that any derived precondition R given by some other rule is at least as strong as D. Hence D satisfies property (b).

Sharpness Theorem. *If predicate* R *satisfies*

$$[\mathrm{R} \Rightarrow \mathrm{wp}(\mathrm{S}(a, b, c), \mathrm{E})]$$

for all procedure bodies S *satisfying* [U$\Rightarrow$wp(S, V)], *then*

$$[\mathrm{R} \Rightarrow \mathrm{D}].$$

Proofs

There is by now a fairly extensive body of theorems about wp and wlp. The ones listed below are some of the most basic, and we make extensive use of them in what follows. For proofs, the reader can consult Dijkstra's original work on the subject [2].

$[\mathrm{wp}(\mathrm{S}, \mathit{false}) \equiv \mathit{false}]$	(Law of the Excluded Miracle)
$[\mathrm{wp}(\mathrm{S}, \mathrm{Q}) \equiv \mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge \mathrm{wlp}(\mathrm{S}, \mathrm{Q})]$	(Definition of wp in terms of wlp)
$[\mathrm{wp}(\mathrm{S}, \mathrm{Q} \wedge \mathrm{R}) \equiv \mathrm{wp}(\mathrm{S}, \mathrm{Q}) \wedge \mathrm{wp}(\mathrm{S}, \mathrm{R})]$	(Conjunctivity)
$[\mathrm{wp}(\mathrm{S}, (\forall x\colon \mathrm{Q})) \equiv (\forall x\colon \mathrm{wp}(\mathrm{S}, \mathrm{Q}))]$	(Infinite conjunctivity)
$[\mathrm{wp}(\mathrm{S}, \mathrm{Q} \vee \mathrm{R}) \Leftarrow \mathrm{wp}(\mathrm{S}, \mathrm{Q}) \vee \mathrm{wp}(\mathrm{S}, \mathrm{R})]$	(Semi-disjunctivity)
$[\mathrm{wp}(\mathrm{S}, (\exists x\colon \mathrm{Q})) \Leftarrow (\exists x\colon \mathrm{wp}(\mathrm{S}, \mathrm{Q}))]$	(Infinite semi-disjunctivity)
$[\mathrm{Q} \Rightarrow \mathrm{R}] \Rightarrow [\mathrm{wp}(\mathrm{S}, \mathrm{Q}) \Rightarrow \mathrm{wp}(\mathrm{S}, \mathrm{R})]$	(Monotonicity)
$[\mathrm{Q}^x_v \equiv (\forall x\colon x = v \Rightarrow \mathrm{Q})]$	(One-point rule with $\forall$)
$[\mathrm{Q}^x_v \equiv (\exists x\colon x = v \wedge \mathrm{Q})]$	(One-point rule with $\exists$).

(Here S is any statement and Q and R are any predicates).

Before presenting the proofs of the soundness and sharpness theorems, we first state and prove one very useful lemma about transparency. This lemma allows us to move predicates in and out of the second argument position of the wp function, something we will need to do in proving the soundness theorem.

Transparency Lemma. *If statement* S *is transparent to every free variable in predicate* A, *then*

$$[\mathrm{wp}(\mathrm{S}, \mathrm{A}) \equiv \mathrm{A} \wedge \mathrm{wp}(\mathrm{S}, \mathit{true})].$$

Proof. The proof proceeds by induction on the number of free variables occurring in A.

Base Case. If there are no free variables in A, then A must be equivalent to either *true* or *false*. If it is equivalent to *true*, then the lemma states that the equivalence

$$\mathrm{wp}(\mathrm{S}, \mathit{true}) \equiv \mathit{true} \wedge \mathrm{wp}(\mathrm{S}, \mathit{true})$$

holds, which is clearly true. Correspondingly, if it is equivalent to *false*, then the lemma states

$$\mathrm{wp}(\mathrm{S}, \mathit{false}) \equiv \mathit{false} \wedge \mathrm{wp}(\mathrm{S}, \mathit{false}),$$

which follows from the Law of the Excluded Miracle.

Induction Case. We assume the lemma holds for predicates A having up to n free variables and prove it for up to $n+1$ free variables.

The proof goes in two parts. The following derivation proves the $\Leftarrow$ half of the lemma:

$$\begin{array}{ll}
 & \mathrm{wp}(\mathrm{S}, \mathrm{A}) \\
= & \{\text{definition of wp in terms of wlp}\} \\
 & \mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge \mathrm{wlp}(\mathrm{S}, \mathrm{A}) \\
= & \{\text{one-point rule}\} \\
 & \mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge \mathrm{wlp}(\mathrm{S}, (\exists v\colon x = v \wedge \mathrm{A}_v^x)) \\
\Leftarrow & \{\text{Infinite semi-disjunctivity of wlp}\} \\
 & \mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge (\exists v\colon \mathrm{wlp}(\mathrm{S}, x = v \wedge \mathrm{A}_v^x)) \\
= & \{\text{conjunctivity of wlp}\} \\
 & \mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge (\exists v\colon \mathrm{wlp}(\mathrm{S}, x = v) \wedge \mathrm{wlp}(\mathrm{S}, \mathrm{A}_v^x)) \\
= & \{\text{by induction hypothesis, since } \mathrm{A}_v^x \text{ has one fewer free variable}\} \\
 & \mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge (\exists v\colon \mathrm{wlp}(\mathrm{S}, x = v) \wedge \mathrm{A}_v^x) \\
\Leftarrow & \{\mathrm{S} \text{ is transparent to } x\} \\
 & \mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge (\exists v\colon x = v \wedge \mathrm{A}_v^x) \\
= & \{\text{one-point rule}\} \\
 & \mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge \mathrm{A}
\end{array}$$

So now we have proven:

$$[\mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge \mathrm{A} \Rightarrow \mathrm{wp}(\mathrm{S}, \mathrm{A})].$$

By taking $\neg \mathrm{A}$ for A we also have:

$$[\mathrm{wp}(\mathrm{S}, \mathit{true}) \wedge \neg \mathrm{A} \Rightarrow \mathrm{wp}(\mathrm{S}, \neg \mathrm{A})]. \tag{3}$$

To prove the $\Rightarrow$ half of the lemma, we first prove

$$[\mathrm{wp}(\mathrm{S}, \mathrm{A}) \Rightarrow \mathrm{A}] \tag{4}$$

by the following derivation:

$$
\begin{array}{ll}
 & \mathrm{wp}(S, A) \wedge \neg A \\
= & \{\text{monotonicity of wp}\} \\
 & \mathrm{wp}(S, A) \wedge \mathrm{wp}(S, \mathit{true}) \wedge \neg A \\
\Rightarrow & \{(3)\} \\
 & \mathrm{wp}(S, A) \wedge \mathrm{wp}(S, \neg A) \\
= & \{\text{conjunctivity of wp}\} \\
 & \mathrm{wp}(S, A \wedge \neg A) \\
= & \{\ \} \\
 & \mathrm{wp}(S, \mathit{false}) \\
= & \{\text{Law of the Excluded Miracle}\} \\
 & \mathit{false}
\end{array}
$$

Since $[\mathrm{wp}(S, A) \Rightarrow \mathrm{wp}(S, \mathit{true})]$, the $\Rightarrow$ half of the lemma follows directly from (4). $\square$

We turn now to the proofs of our main results. We begin with the soundness theorem.

Soundness Theorem. *For all procedure bodies* S *satisfying* $[U \Rightarrow \mathrm{wp}(S, V)]$, *we have*

$$[(\exists m\colon U^{x,y}_{a,b}) \wedge (\forall y, z\colon (\forall m\colon U^{x,y}_{a,b} \Rightarrow V^{x}_{a}) \Rightarrow E^{b,c}_{y,z}) \Rightarrow \mathrm{wp}(S(a, b, c), E)].$$

Proof. The first part of the proof generalizes the idea used by A.J. Martin in [6]. Start with any predicates P and A, where A is independent of y and z, that satisfy

$$[P \wedge A \Rightarrow E^{b,c}_{y,z}] \tag{5}$$

and with a procedure body S satisfying $[U \Rightarrow \mathrm{wp}(S, V)]$. From this simple assumption, one can prove the formula

$$[\mathrm{wp}(S, P)^{x,y}_{a,b} \wedge A^{x}_{a} \Rightarrow \mathrm{wp}(S(a, b, c), E)]$$

which one could call a "skeleton" proof rule. The part to the right of the implication sign is as desired; it requires only intelligent choices for P and A to get an interesting proof rule. The proof of this "skeleton" goes as follows:

$$
\begin{array}{ll}
 & [\mathrm{wp}(S, P)^{x,y}_{a,b} \wedge A^{x}_{a} \Rightarrow \mathrm{wp}(S(a, b, c), E)] \\
= & \{\text{definition of } S(a, b, c)\} \\
 & [\mathrm{wp}(S, P)^{x,y}_{a,b} \wedge A^{x}_{a} \Rightarrow \mathrm{wp}(S, E^{b,c}_{y,z})^{x,y}_{a,b}] \\
\Leftarrow & \{\text{instantiation, using that A is independent of } y\} \\
 & [\mathrm{wp}(S, P) \wedge A \Rightarrow \mathrm{wp}(S, E^{b,c}_{y,z})] \\
= & \{\text{transparency lemma, using that A is independent of } y \text{ and } z\} \\
 & [\mathrm{wp}(S, P) \wedge \mathrm{wp}(S, A) \Rightarrow \mathrm{wp}(S, E^{b,c}_{y,z})] \\
\Leftarrow & \{\text{conjunctivity of wp, monotonicity of wp}\} \\
 & [P \wedge A \Rightarrow E^{b,c}_{y,z}] \\
= & \{\text{by (5)}\} \\
 & \mathit{true}.
\end{array}
$$

Now what are good choices for P and A?

One possibility is to take P to be V and let A be determined by (5); this is the rule presented by Martin in [6]. In this case, the first conjunct is readily seen to be implied by $U_{a,b}^{x,y}$ and the rule looks like

$$[U_{a,b}^{x,y} \wedge A_a^x \Rightarrow \mathrm{wp}(S(a, b, c), E)].$$

A second possibility is to take P to be V and take A to be $(\forall y, z\colon P \Rightarrow E_{y,z}^{b,c})$. This leads to the proof rule of Gries described above (1).

We, however, take P to be $(\forall m\colon U_b^y \Rightarrow V)$, and A to be $(\forall y, z\colon P \Rightarrow E_{y,z}^{b,c})$. This indicates the crucial difference between Gries's rule and ours: a better choice for P.

We first show that our A and P satisfy assumption (5) above. This is a direct consequence of our choice of A, as the following derivation shows:

$$\begin{array}{ll}
 & (5) \\
= & \{\text{inserting our choice of A}\} \\
 & [P \wedge (\forall y, z\colon P \Rightarrow E_{y,z}^{b,c}) \Rightarrow E_{y,z}^{b,c}] \\
\Leftarrow & \{\text{instantiation}\} \\
 & [P \wedge (P \Rightarrow E_{y,z}^{b,c}) \Rightarrow E_{y,z}^{b,c}] \\
= & \{\ \} \\
 & \mathit{true}
\end{array}$$

Knowing that our P and A satisfy the assumption, we now insert them into the "skeleton" to get

$$[\mathrm{wp}(S, \forall m\colon U_b^y \Rightarrow V))_{a,b}^{x,y} \wedge (\forall y, z\colon (\forall m\colon U_b^y \Rightarrow V)_a^x \Rightarrow E_{y,z}^{b,c}) \Rightarrow \mathrm{wp}(S(a, b, c), E)].$$

As can be seen by comparing this formula with the desired conclusion, we need only prove

$$[(\exists m\colon U_{a,b}^{x,y}) \Rightarrow \mathrm{wp}\,(S, (\forall m\colon U_b^y \Rightarrow V))_{a,b}^{x,y}]$$

to obtain our desired result. This fact follows from the following derivation:

$$\begin{array}{ll}
 & \mathrm{wp}(S, (\forall m\colon U_b^y \Rightarrow V))_{a,b}^{x,y} \\
= & \{\text{infinite conjunctivity}\} \\
 & \forall m\colon \mathrm{wp}(S, \neg U_b^y \vee V))_{a,b}^{x,y} \\
\Leftarrow & \{\text{semi-disjunctivity}\} \\
 & \forall m\colon \mathrm{wp}(S, \neg U_b^y)_{a,b}^{x,y} \vee \mathrm{wp}(S, V)_{a,b}^{x,y} \\
= & \{\text{transparency lemma; } \mathrm{wp}(S, V) \Rightarrow \mathrm{wp}(S, \mathit{true})\} \\
 & \forall m\colon (\mathrm{wp}(S, \mathit{true})_{a,b}^{x,y} \wedge \neg U_{a,b}^{x,y}) \vee (\mathrm{wp}(S, \mathit{true})_{a,b}^{x,y} \wedge \mathrm{wp}(S, V)_{a,b}^{x,y})] \\
= & \{\mathrm{wp}(S, \mathit{true})_{a,b}^{x,y} \text{ independent of } m\} \\
 & \mathrm{wp}(S, \mathit{true})_{a,b}^{x,y} \wedge (\forall m\colon \neg U_{a,b}^{x,y} \vee \mathrm{wp}(S, V)_{a,b}^{x,y}) \\
\Leftarrow & \{S \text{ satisfies } (U, V)\text{; hence in particular } [(\exists m \cdot U) \Rightarrow \mathrm{wp}(S, \mathit{true})]\} \\
 & (\exists m\colon U_{a,b}^{x,y}). \quad \square
\end{array}$$

Sharpness Theorem. *If predicate* R *satisfies*

$$[R \Rightarrow \mathrm{wp}(S(a, b, c), E)],$$

for all procedure bodies S *satisfying* $[U \Rightarrow wp(S, V)]$, *then*

$$[R \Rightarrow (\exists m\colon U^{x,y}_{a,b}) \wedge (\forall y, z\colon (\forall m\colon U^{x,y}_{a,b} \Rightarrow V^{x}_{a}) \Rightarrow E^{b,c}_{y,z})].$$

Proof. First a few words to motivate the technical details of the theorem. Consider the following hypothetical procedure body:

S_H: **if** $(\exists m\colon U) \rightarrow$ "choose at random a pair (y', z') satisfying $(\forall m\colon U \Rightarrow V^{y,z}_{y',z'})$";
$y, z := y', z'$
fi

It is easy to convince oneself that such a procedure body *exactly* satisfies the specification (U, V). Thus, if such an S_H could always be constructed, the theorem would follow immediately. Unfortunately, $(\exists m\colon U)$ and $(\forall m\colon U \Rightarrow V^{y,z}_{y',z'})$ are not always computable predicates. To get around this difficulty, the proof uses two families of procedure bodies, denoted by S^*_t and S^{**}_t. The S^*_t are used to simulate a procedure body guarded by the condition $(\exists m\colon U)$, while the S^{**}_t are used to simulate a body that randomly chooses a satisfying pair (y', z').

We now proceed with the proof itself. The proof is divided into two parts. Part A proves the formula

$$[R \Rightarrow (\exists m\colon U^{x,y}_{a,b})] \tag{6}$$

while Part B proves

$$[R \Rightarrow (\forall y, z\colon (\forall m\colon U^{x,y}_{a,b} \Rightarrow V^{x}_{a}) \Rightarrow E^{x,y}_{a,b})]. \tag{7}$$

Both parts make use of our assumption that there is at least one procedure body S_0 satisfying the specification

$$[U \Rightarrow wp(S_0, V)].$$

Part A. We construct, using S_0, a family of new procedure bodies. Each body S^*_t has an associated tuple of constants $\mathbf{t} = (a_t, b_t)$, and S^*_t is defined to behave like S_0 except that it aborts when $(x, y) = (a_t, b_t)$:

$$S^*_t\colon \textbf{ if } \neg(x = a_t \wedge y = b_t) \rightarrow S_0 \textbf{ fi}.$$

By the definition of wp, it is easily seen that

$$wp(S^*_t, Q) \equiv \neg(x = a_t \wedge y = b_t) \wedge wp(S_0, Q) \tag{8}$$

for any predicate Q.

We now start with the statement "for any $\mathbf{t}$, if S^*_t satisfies the specification, then R is a sufficient precondition for the call on S^*_t to terminate with E true" (this statement is implied by the theorem's hypothesis). From this statement, we derive expression (6).

$$\begin{aligned}
& \text{for any } \mathbf{t}\colon [U \Rightarrow wp(S^*_t, V)] \Rightarrow [R \Rightarrow wp(S^*_t(a, b, c), E)] \\
= \; & \{\text{by } (8)\} \\
& \text{for all } \mathbf{t}\colon [U \Rightarrow \neg(x = a_t \wedge y = b_t) \wedge wp(S_0, V)] \\
& \qquad \Rightarrow [R \Rightarrow \neg(a = a_t \wedge b = b_t) \wedge wp(S_0, E^{b,c}_{y,z})^{x,y}_{a,b}]
\end{aligned}$$

$$
\begin{aligned}
\Rightarrow\ & \{[U \Rightarrow wp(S_0, V)] \text{ and } [R \Rightarrow wp(S_0, E^{b,c}_{y,z})^{x,y}_{a,b}] \text{ by hypothesis}\} \\
& \text{for all } \mathbf{t}\colon [U \Rightarrow \neg(x=a_t \wedge y=b_t)] \Rightarrow [R \Rightarrow \neg(a=a_t \wedge b=b_t)] \\
=\ & \{x=a_t \wedge y=b_t \text{ independent of } m\} \\
& \text{for all } \mathbf{t}\colon [(\exists m\colon U) \Rightarrow \neg(x=a_t \wedge y=b_t)] \Rightarrow [R \Rightarrow \neg(a=a_t \wedge b=b_t)] \\
=\ & \{\text{since } (\forall x\colon Q \Rightarrow x \neq v) \equiv \neg Q^x_v \text{ by the one-point rule}\} \\
& \text{for all } \mathbf{t}\colon \neg(\exists m\colon U^{x,y}_{a_t,b_t}) \Rightarrow [\neg R \vee \neg(a=a_t \wedge b=b_t)] \\
=\ & \{\neg(\exists m\colon U^{x,y}_{a_t,b_t}) \text{ has no free variables}\} \\
& \text{for all } \mathbf{t}\colon [\neg(\exists m\colon U^{x,y}_{a_t,b_t}) \Rightarrow \neg R \vee \neg(a=a_t \wedge b=b_t)] \\
=\ & \{\text{contrapositive}\} \\
& \text{for all } \mathbf{t}\colon [R \wedge a=a_t \wedge b=b_t \Rightarrow (\exists m\colon U^{x,y}_{a_t,b_t})] \\
=\ & \{\text{taking } (a, b) \text{ to be } (a_t, b_t)\} \\
& [R \Rightarrow (\exists m\colon U^{x,y}_{a,b})]
\end{aligned}
$$

Part B. Again we construct a family of new procedure bodies. We associate with each tuple of constants $\mathbf{t}=(a_t, b_t, y_t, z_t)$ the procedure body $S^{**}_{\mathbf{t}}$ defined by

$$
\begin{aligned}
S^{**}_{\mathbf{t}}\colon\ & \textbf{if } x=a_t \wedge y=b_t \rightarrow y, z := x_t, z_t \\
& \ \square\ x \neq a_t \vee y \neq b_t \rightarrow S_0 \\
& \textbf{fi}.
\end{aligned}
$$

By the definition of wp, we see that

$$
wp(S^{**}_{\mathbf{t}}, Q) \equiv (x=a_t \wedge y=b_t \Rightarrow Q^{y,z}_{y_t,z_t}) \wedge (\neg(x=a_t \wedge y=b_t) \Rightarrow wp(S_0, Q)) \quad (9)
$$

for any predicate Q.

Starting from the same point as in Part A (but with $S^{**}_{\mathbf{t}}$ for $S^{*}_{\mathbf{t}}$), we now derive expression (7).

$$
\begin{aligned}
& \text{for all } \mathbf{t}\colon [U \Rightarrow wp(S^{**}_{\mathbf{t}}, V)] \Rightarrow [R \Rightarrow wp(S^{**}_{\mathbf{t}}(a, b, c), E)] \\
=\ & \{\text{by } (9)\} \\
& \text{for all } \mathbf{t}\colon [(U \wedge x=a_t \wedge y=b_t \Rightarrow V^{y,z}_{y_t,z_t}) \\
& \qquad \wedge (U \wedge \neg(x=a_t \wedge y=b_t) \Rightarrow wp(S_0, V))] \\
& \quad \Rightarrow [(R \wedge a=a_t \wedge b=b_t \Rightarrow (E^{b,c}_{y,z})^{y,z}_{y_t,z_t}) \\
& \qquad \wedge (R \wedge \neg(a=a_t \wedge b=b_t) \Rightarrow wp(S_0, E^{b,c}_{y,z})^{x,y}_{a,b})] \\
=\ & \{\text{observing that } [U \Rightarrow wp(S_0, V)] \text{ and } [R \Rightarrow wp(S_0, E^{b,c}_{y,z})^{x,y}_{a,b}] \text{ by hypothesis}\} \\
& \text{for all } \mathbf{t}\colon [U \wedge x=a_t \wedge y=b_t \Rightarrow V^{y,z}_{y_t,z_t}] \Rightarrow [R \wedge a=a_t \wedge b=b_t \Rightarrow E^{b,c}_{y_t,z_t}] \\
=\ & \{\text{one-point rule}\} \\
& \text{for all } \mathbf{t}\colon [(U^{x,y}_{a_t,b_t} \Rightarrow V^{x,y,z}_{a_t,y_t,z_t})] \Rightarrow [R \wedge a=a_t \wedge b=b_t \Rightarrow E^{b,c}_{y_t,z_t}] \\
=\ & \{\text{observing that } U^{x,y}_{a_t,b_t} \text{ and } V^{x,y,z}_{a_t,y_t,z_t} \text{ depend only on } m\} \\
& \text{for all } \mathbf{t}\colon [R \wedge a=a_t \wedge b=b_t \Rightarrow ((\forall m \cdot U^{x,y}_{a_t,b_t} \Rightarrow V^{x,y,z}_{a_t,y_t,z_t}) \Rightarrow E^{b,c}_{y_t,z_t})] \\
=\ & \{\text{one-point rule}\} \\
& [\forall y, z \cdot R \Rightarrow ((\forall m\colon U^{x,y}_{a,b} \Rightarrow V^{x}_{a}) \Rightarrow E^{b,c}_{y,z})] \\
=\ & \{R \text{ independent of } y \text{ and } z\} \\
& [R \Rightarrow (\forall y, z\colon (\forall m\colon U^{x,y}_{a,b} \Rightarrow V^{x}_{a}) \Rightarrow E^{b,c}_{y,z})]
\end{aligned}
$$

This finishes the proof of the Sharpness theorem. □

Note: It was recently brought to our attention that a paper by E.-R. Olderog [7] discusses the relative incompleteness of Gries's proof role and presents a

formula (W, page 342) which is reminiscent of our D. His results, however, are not the same as ours: he considers only partial correctness, and his notion of relative completeness is not the same as our notion of sharpness.

References

1. Bijlsma, A., Wiltink, J.G., Matthews, P.A.: Equivalence of the Gries and Martin Proof Rules for Procedure Calls. Acta Inf. **23**, 357–360 (1986)
2. Dijkstra, E.W.: A Discipline of Programming. Englewood Cliffs, New Jersey: Prentice Hall 1976
3. Gries, D.: The Science of Programming. New York: Springer 1981
4. Gries, D., Levin, G.: Assignment and Procedure Call Proof Rules. ACM Trans. Progr. Lang. Syst. **2**, 564–579 (1980)
5. Hemerik, C.: Formal Definitions of Programming Languages as a Basis for Compiler Construction. Thesis, Eindhoven 1984
6. Martin, A.J.: A General Proof Rule for Procedures in Predicate Transformer Semantics. Acta Inf. **20**, 301–313 (1983)
7. Olderog, E.-R.: On the Notion of Expressiveness and the Rule of Adaptation. Theor. Comput. Sci. **24**, 337–347 (1983)

Received August 15, 1988 / November 29, 1988