Acta Informatica 23, 357–360 (1986)

Equivalence of the Gries and Martin Proof Rules for Procedure Calls

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

[1] Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[2] Computer Science Department, Aarhus University, Ny Munkegade, DK 8000 Aarhus C, Denmark

Summary. In this note we prove the equivalence of the proof rules for procedure calls as given by D. Gries [1] and A.J. Martin [2]. We also discuss a modification of these proof rules for the case that the specification of a procedure contains free constants.

Introduction

In this note, we reconsider the problem addressed in [2]. The reader is referred to that paper for a more complete discussion of the problem and of the notations used; here we shall merely recapitulate the essentials.

Consider a procedure declaration of the form

$$\textbf{proc}\ p(x?, y, z!);\ S.$$

The state space of statement S consists of x, y, and z only. A call to procedure p is of the form

$$p(a, b, c),$$

where b and c are distinct variables and a is an expression. The state space of the call does not contain x, y, or z. Semantically, $p(a, b, c)$ is defined to be equivalent to a block with local variables x, y, and z and statement list

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

Let E be an arbitrary predicate independent of x, y, and z. Computation of $wp(p(a, b, c), E)$ by means of the definition of $p(a, b, c)$ requires full knowledge of S. If procedural abstraction is to be useful, however, one needs a method to specify the semantics of the procedure without mentioning the actual statements of the procedure body and a method to construct, from that specification only, a predicate F such that

$$[F \Rightarrow wp\,(p(a, b, c), E)].$$

(Here, and everywhere else in this note, square brackets denote universal quantification over all variables on which the enclosed predicate depends.)

We choose to specify the procedure by means of a pair of predicates U and V satisfying

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

A trivial method to construct the predicate F would be to take for F the predicate *false*; this solution, however, is useless.

It will be assumed

- that $$[x = X \Rightarrow wlp(S, x = X)]$$

for any constant X of appropriate type,

- that $wp(S, Q)$ is independent of z for any predicate Q,
- that U depends on x and y only, and
- that V depends on x, y, and z only.

Under these hypotheses, the problem has two well-known solutions. One is the rule proposed by D. Gries (Theorem (12.4.1) of [1]); in our notation it consists in taking for F the predicate

$$(\forall u, v: V^{x,y,z}_{a,u,v} : E^{b,c}_{u,v}) \wedge U^{x,y}_{a,b}. \tag{0}$$

The other is A.J. Martin's rule [2], which states that we may take F to be

$$(A \wedge U)^{x,y}_{a,b}$$

for any predicate A independent of y and z satisfying

$$[A \wedge V \Rightarrow E^{b,c}_{y,z}]. \tag{1}$$

In his paper, Martin states that his rule is the more general of the two. It is the purpose of this note to show that this is not the case and that the two rules are, in fact, equivalent in the following sense: the predicate in Gries's rule is the weakest predicate obtainable from Martin's rule. In our opinion, the remaining significance of Martin's paper is that it provides a nicer proof of Gries's rule and a different, perhaps easier, strategy for applying it.

Proof of the Equivalence

As a first step, we show that the weakest A independent of y and z satisfying (1) is

$$(\forall y, z : V : E^{b,c}_{y,z}): \tag{2}$$

for any A independent of y and z we have

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

$$= \{\text{making the quantifications over } y \text{ and } z \text{ explicit}\}$$

$$[(\forall y, z : : A \wedge V \Rightarrow E^{b,c}_{y,z})]$$

$$= \{\text{predicate calculus}\}$$

$$[(\forall\, y, z : V : A \Rightarrow E^{b,c}_{y,z})]$$
$$= \{A \text{ is independent of } y \text{ and } z\}$$
$$[A \Rightarrow (\forall\, y, z : V : E^{b,c}_{y,z})].$$

Thus any solution A of (1) that is independent of y and z implies (2). On the other hand, (2) itself is independent of y and z, and if we substitute (2) for A, then the last line of the derivation (and thereby the first) reduces to *true*. Hence (2) is indeed the weakest A independent of y and z satisfying (1). We conclude that the weakest predicate F obtainable from Martin's rule is

$$(A \wedge U)^{x,y}_{a,b}$$

with (2) for A.

This predicate is equivalent to (0):

$$((\forall\, y, z : V : E^{b,c}_{y,z}) \wedge U)^{x,y}_{a,b}$$
$$= \{\text{renaming the dummies; } E \text{ independent of } y \text{ and } z\}$$
$$((\forall\, u, v : V^{y,z}_{u,v} : E^{b,c}_{u,v}) \wedge U)^{x,y}_{a,b}$$
$$= \{a \text{ independent of } y \text{ and } z;\ E \text{ independent of } x \text{ and } y\}$$
$$(\forall\, u, v : V^{x,y,z}_{a,u,v} : E^{b,c}_{u,v}) \wedge U^{x,y}_{a,b}.$$

Further Remarks

0. In [2], Martin erroneously states that Gries's rule can be obtained from Martin's rule by substituting

$$[V \Rightarrow E^{b,c}_{y,z}]$$

for A. This would, of course, yield a rule that is far less general.

1. Martin [2] requires that

$$[x = X \equiv wlp(S, x = X)]$$

for any constant X. He erroneously states that this requirement is satisfied if S is transparent to x, i.e. contains no assignments to x and no procedure calls with x as an output or input-output argument. As a counterexample, let S denote the statement

$$\textbf{do}\ x = 0 \rightarrow \mathit{skip}\ \textbf{od};$$

then

$$[wlp(S, x = 1) \equiv x = 0 \vee x = 1].$$

However, for the proof of the procedure rule only the weaker

$$[x = X \Rightarrow wlp(S, x = X)]$$

is necessary. This is valid for S transparent to x, as can be proved by induction on the structure of S.

2. In practice, the specification of S often contains so-called free constants, i.e. identifiers that do not occur in any of the relevant state spaces. For instance, the body of a procedure named *Root*, with parameter y only, might have

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

as its precondition and

$$y = Y$$

as its postcondition, where Y is a free constant. (A different example can be found in Sect. 4.3 of [2].) To deal with specifications of this type, the proof rule should be slightly modified.

If S is specified by a pair of predicates $U(Y)$ and $V(Y)$ that contain a free constant Y, this means that

$$(\forall\, Y : : [U(Y) \Rightarrow wp(S, V(Y))]).$$

The proof rule above gives, for each value of Y, a precondition $F(Y)$ such that

$$(\forall\, Y : : [F(Y) \Rightarrow wp(p(a, b, c), E)]),$$

which is equivalent to

$$[(\exists\, Y : : F(Y)) \Rightarrow wp(p(a, b, c), E)].$$

Hence, (0) should be extended with existential quantification over all free constants occurring in U or V.

As an example, let *Root* be the procedure specified above and suppose that we wish to determine a precondition F such that

$$[F \Rightarrow wp(Root(b), b = 5)].$$

Existential quantification of (0) over Y gives

$$(\exists\, Y : : (\forall\, u : (y = Y)^y_u : (b = 5)^b_u) \wedge (Y^2 \leqq y < (Y+1)^2)^y_b)$$

for F. By predicate calculus (the one-point rule in particular), this is equivalent to $25 \leqq b < 36$.

References

1. Gries, D.: The Science of Programming. Berlin, Heidelberg, New York: Springer 1981
2. Martin, A.J.: A general proof rule for procedures in predicate transformer semantics. Acta Inf. **20**, 301-313 (1983)

Received July 8, 1985 / March 11, 1986