SIAM J. COMPUT.
Vol. 10, No. 3, August 1981

0097-5397/81/1003-0016 $01.00/0

CORRIGENDUM: SOUNDNESS AND COMPLETENESS OF AN AXIOM SYSTEM FOR PROGRAM VERIFICATION*

STEPHEN A. COOK†

K. R. Apt pointed out to me that Theorem 3 (completeness) is technically false, because of a problem with initializing newly declared variables. For example, the formula

$$\text{true}\ \{\textbf{begin begin new}\ x;\ x := 1\ \textbf{end};\ \textbf{begin new}\ x;\ y := x\ \textbf{end end}\}\ y = 1$$

is valid according to the semantics given (because the second declaration of x assigns the same register to x as the first), but it is not provable in $\mathcal{H}$.

Perhaps the simplest way to fix this is to require all newly declared variables to be initialized to some distinguished value $0 \in D$. This would involve changing the first case (that of variable declaration) in the definition of Comp (A, s, δ, π) on p. 74, so that the computation proceeds with a new state s'. Here s' is the same as s except for $s'(X_{k+1}) = 0$. To make $\mathcal{H}$ complete we would slightly modify Rule 1 (Rule of variable declarations) of the system $\mathcal{H}$ to read

$$\frac{x = 0\ \&\ P\frac{y}{x}\{\textbf{begin}\ D^*;\ A^*\ \textbf{end}\}Q\frac{y}{x}}{P\ \{\textbf{begin new}\ x;\ D^*;\ A^*\ \textbf{end}\}\ Q}.$$

A second possible fix, suggested in Apt [1], requires no changes in the proof system $\mathcal{H}$, but changes the semantics so that $\mathcal{H}$ becomes complete. The idea is that each newly declared variable is assigned a register that has never been used before. A state s would be redefined so that it assigns a member of $D \times \{0, 1\}$ to each register X_k instead of simply a member of the domain D. The second component of $s(X_k)$ indicates whether X_k has been assigned previously. We would only consider pairs (s, δ) in the definition of Comp (A, s, δ, π), $P(s, \delta)$, etc. such that $(s(\delta(x)))_2 = 1$ for each variable x in the domain of δ, indicating that register $\delta(X)$ has been assigned. The first case in the definition of Comp would be changed so that $\delta'(x) = X_k$, where X_k is the first register for which $(s(X_k))_2 = 0$. Also the computation would continue in a new state s' such that $(s'(X_k))_2 = 1$. The other cases of Comp would be unchanged except for minor editing.

REFERENCE

[1] K. R. APT, *Ten years of Hoare's logic, a survey*, Proceedings of the 5th Scandinavian Logic Symposium, Aalborg University Press, Aalborg, Denmark, 1979, pp. 1–44.

---

* This Journal, 7 (1978), pp. 70–90.

† Department of Computer Science, University of Toronto, Toronto, Canada M5S 1A7.