relationships other than the computer time relationships and did so for the reasons stated in my paper.

Fourer also claims that had there been a linear relationship, the ratios I/m_{RLP} would have clustered around a constant. Again, I determined average results; that comment would be valid if asymptotic or predictive relationships had been sought, but they were not.

In conclusion, Fourer's comments are important and valid for the problem with which he is concerned. However, he seeks asymptotic trends and predictive equations, and I determined average performance results over a finite interval. My claims are valid since they are prose statements of numeric results. Further, the results given in my paper are useful since they: (1) provide important model statistics and actual performance results not generally available to linear programming students and researchers; (2) show quantitatively just how well the simplex algorithm performs in practice; (3) confirm numerically the general experience of LP practitioners that the simplex algorithm has a linear time behavior in practice; (4) provide a minimal overall performance level other algorithms must achieve for those algorithms to replace the simplex algorithm in solving real-world LP problems; (5) as noted by Fourer, do " . . . tend to confirm two widely held impressions: that T never 'blows up' as the size of practical LPs increases, and that I is usually a small multiple of the number of constraints."; (6) as also noted by Fourer, " . . . [the results] show that considerable reductions in LP size can sometimes be achieved by automatic elimination of redundant constraints."; and (7) provide an empirical upper bound on the number of iterations required to solve an LP problem.

Edward H. McCall
Sperry Corporation
St. Paul, MN 55164

CRITIQUING A CRITIQUE OF HOARE'S PROGRAMMING LOGICS

I would like to point out certain flaws in Michael J. O'Donnell's paper "A Critique of the Foundations of the Hoare Style Programming Logics" [1]. I confine my remarks to the proofs of inconsistency of the rule Function-1.

The rule Function-1 is meaningful only for the total functions f unless first-order logic is extended by providing an interpretation for the formula $\forall x(A \Rightarrow B(f(x)/y))$, where f is a partial function. In order to prove inconsistency of the rule Function-1 (added to the earlier described fragment of programming logic) one has to prove inconsistency of this rule under every reasonable interpretation. The first proof of inconsistency of Function-1 falls short of this expectation.

The author considers "a predicate calculus formula containing a program-defined function f to be true when it is true for all total functions f consistent with the values computed by the definition of f". The first inconsistency proof is valid under this interpretation. However it fails if we take a point of view that program P defines a single-valued predicate $f(x) = y$, so that the formula $\forall x(A \Rightarrow B(f(x)/y))$ means

$$\forall x \forall y\ [(A\ \&\ f(x) = y) \Rightarrow B].$$

The first approach seems to be more appropriate when one reasons about a total function given partial information about its values, rather than in our case when the given function is definitely partial. Note also that the first interpretation is second-order in its nature whereas the formula $\forall x(A \Rightarrow B(f(x)/y))$ is supposed to be first-order. The second interpretation is certainly in the spirit of first-order logic and is usual among logicians. See, for example, [2] where partial functions are, by definition, single-valued relations.

The second inconsistency proof (*a la* Russell's paradox) is plainly wrong. Every single line of it is illegal because the formulas are not first-order. O'Donnell says it is essentially Russell's paradox in disguise. I do not see any disguise. I see the usual Russell paradox with a superfluous use of Function-1.

Yuri Gurevich
The University of Michigan
Ann Arbor, MI 48109

O'Donnell's critique on Hoare-Style programming logics contains several reasoning errors.

First in section 5 on "Defined functions" (p. 930), O'Donnell's derivation shows a weak inconsistency. His derivation is the following:

$$\text{F-1: } \frac{A\{P\}B}{\forall x[A \Rightarrow B(f(x)/y)]}$$

f:Function(x); Fail; Return (y); end

1) True {**Fail**} False
2) $\forall$x True $\Rightarrow$ False
3) False

But $B(f(x)/y)$ is defined only for x in Df = {x:[E:y:f(x) = y]}, i.e. where $f(x)$ is defined. This means $\forall x(A \Rightarrow B(f(x)/y))$ applies only to x's such that $A \Rightarrow$ (x in *Df*). But Df = { }, so it is true vacuously and 2) to 3) cannot be done: vacuous instantiation is not legitimate.

In other words, substituting something that represents nothing for something that represents something results in nothing. In fact such use implicitly assumes that nothing is something and introduces a false premise into the analysis.

Similarly O'Donnell defines the Russell paradox equivalent as: "r:**Function**(g); $y := 1 - g(g)$; **return**(y); **end**", where g is a characteristic function for a set. He then shows r(r) ⟨⟩ r(r) using Hoare logic steps. This can be derived without Hoare logic in the following manner:

Let $y = r(g)$, i.e. $y = 1 - g(g)$, for any g then
1) $y = g(g) \Rightarrow y = ½$

But characteristic functions range over (0, 1) so that
2) y ⟨⟩ g(g)

This is true for every g so
3) [$\forall$g:r(g) ⟨⟩ g(g)]

Thus for g = r
4) r(r) ⟨⟩ r(r)

Thus the paradox is independent of Hoare logic and this derivation also implies the "nothing for something" premise in the following manner:

Let Dr = (g:[E:y:y = r(g)]) be "the domain of r"

Now **if** r is in Dr **then**
1) r(r) = 1 − r(r)

But r(r) is in (0, 1) thus
2) r(r) = 0 ⇒ 1 − r(r) = 1 while r(r) = 1 ⇒ 1 − r(r) = 0

Either way, 0 ⟨⟩ 1 so
3) r(r) ⟨⟩ 1 − r(r)

But this contradicts the assumption that r is in Dr so
4) r is not in Dr and r(r) is nothing.

Now, since r is not in Dr, "r(r)" is nonexistent, and is illegal to use. Thus no contradiction from this is derivable; r is not itself a characteristic function of a set. This is why using r(r) is bad.

There is no "paradox" since its derivation depends on an implicit false assumption. And the treating of a falsehood as a truth entails inconsistency and is the source of all paradoxes including this one.

Russell made the same error by assuming his set (S, S) was in Din = ((X, Y): [E:y:y = X in Y]). But here again the argument, S not in Din, is provable while its contradictory is not. This follows from the equivalence of characteristic functions with their sets. Now r is not such a function but is the equivalent to S so S is not a set. This is why S in S, like r(r), is bad.

Thus Russell's paradox is nothing more than the square circle nonsense: S and "square circles" do not exist; S and "square circles" do not reference anything. But their use as something assumes they do.

I suggest that the following general principles should be observed: 1) Existence in the appropriate domain must be shown before the reference can otherwise be used; 2) Nonexistence has no properties other than not existing; 3) Do not reify a zero; 4) Do not treat representative substitution like textual substitution; and 5) Do not ascribe problems to something without having shown that the something is the source of the problem.

Mark L. Millard
8415 S.E. Stephens St.
Portland, OR 97216

AUTHOR'S RESPONSE

The observations of Yuri Gurevich and Mark Millard provide an alternate analysis of one of the contradictions shown in my paper, but do not show errors in my own reasoning. The derivation on p. 930, which Gurevich calls "the first proof of inconsistency of the rule Function-1," and which Millard calls a demonstration of "weak inconsistency," follows the formal rules mentioned in Section 4, which include a standard set of Hoare-style rules, and any reasonable set of rules for reasoning in the classical predicate calculus (such as those given in Prawitz [6]) plus the rule Function-1. The derivation in my paper is taken from Ashcroft, Clint, and Hoare [3]. Clint and Hoare's authorship of [3] seems to constitute acknowledgment that the formal rules applied in this derivation agree with the intent of their function rule, given in [4].

Gurevich and Millard correctly observe that a contradiction in a combination of formal rules may be avoided by changing any one of the rules. They propose to introduce notation for the well-definedness of $f(x)$ into the predicate calculus. This is one legitimate way to avoid the contradiction, and is already given in [3]. I find this resolution unsatisfying because it seems to require either some extra notation ("(x, y)ϵf" in [3] and "x in Df" in Millard's derivation) or a reinterpretation of the familiar notation "$f(x)$" ("$f(x) = y$" denotes a relation between x and y in Gurevich's letter). Also, Ashcroft, Clint and Hoare [3] point out that this change in predicate calculus notation is not very helpful to programmers without a corresponding set of rules for proving termination. The change to total-correctness logic solves the whole problem at once, with no new notation and no reinterpretation of traditional mathematical notation.

My simulation of Russell's paradox on p. 931 violates the standard notational restrictions of the first-order predicate calculus, but may easily be rewritten with a binary operator to represent function application ("apply(f, x)" for "$f(x)$"). Such a translation obscures more than it clarifies. The rule Function-1 in this derivation acts like the rule

$$\frac{x \epsilon \{y | A\}}{A(x/y)}$$

in naive set theory. Millard's derivation of r(r)⟨⟩r(r) needs some such rule in order to get from $y = r(g)$ to $y = 1 - g(g)$ in the very first line. His derivation also needs some notation for introducing the definition of the function r. There are ways to do so without Hoare logic, but not within the pure predicate calculus of first or second order. The paradoxical derivation disappears in total-correctness logic, even with a second-order predicate calculus. Such a resolution of Russell's paradox is very similar to that of Fitch's set theory [5, 6]. Fitch essentially requires a certain program to be run on every supposed proof in his system, and the proof is incorrect if the program fails to halt. A combination of some form of naive set theory with total-correctness logic might improve upon Fitch's ideas in cases where a proof of termination is shorter or easier than the terminating computation itself.

Michael J. O'Donnell
Purdue University
West Lafayette, IN 47907

REFERENCES

1. O'Donnell, M.J. A critique of the foundations of Hoare-style programming logic. Comm. *ACM 25*, 12 (Dec. 1982), 927–935.
2. Rogers, Jr., H. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967.
3. Ashcroft, E.A., Clint, M., and Hoare, C.A.R. Remarks on program proving: jumps and functions, *Acta Informatica 6*, 3, (1976), 317.
4. Clint, M. and Hoare, C.A.R. Program proving: jumps and functions, *Acta Informatica 1*, 3 (1972), 214–224.
5. Fitch, F. *Symbolic Logic*. New York, 1952.
6. Prawitz, D. *Natural Deduction: A Proof-Theoretical Study*. Almquist & Wiksell, Stockholm, 1965.