A simple fix-point argument without the restriction to continuity.

by

Edsger W. Dykstra and A. J. M. van Gasteren

Introduction

Notation In this text, the letters B, P, R, X, and Y stand for predicates on the state space of a program and square brackets are used as notation for universal quantification of the enclosed over the program variables. The letter S stands for a statement and DO for the repetitive construct **do** $B \rightarrow S$ **od** (see [3]). (End of Notation.)

Requiring DO to be semantically equivalent to its first unfolding

$$\textbf{if}\ B \rightarrow S; \mathrm{DO} \ [\!] \ \neg B \rightarrow \mathrm{skip}\ \textbf{fi}$$

yields that $wp(\mathrm{DO}, \neg B \wedge P)$ is a solution of the equation in predicate X

$$(0) \quad [X \equiv (B \wedge wp(S, X)) \vee (\neg B \wedge P)] \ .$$

Explanation Equation (0) follows from the required semantic equivalence and

(i) the semantic definition of skip, viz

$$[wp(\text{"skip"}, R) \equiv R] \quad \text{for all } R;$$

(ii) the semantic definition of statement concatenation, viz.

$[wp("S0;S1", R) \equiv wp(S0, wp(S1, R))]$
for all $S0$, $S1$, and R ;

(iii) the semantic definition of the alternative construct, in particular

$[wp("\textbf{if}\ B \to S0 \ [\!]\ \neg B \to S1\ \textbf{fi}", R) \equiv$
$(B \wedge wp(S0, R)) \vee (\neg B \wedge wp(S1, R))]$
for all B, $S0$, $S1$, and R .

For further details, see [3]. (End of Explanation.)

Notation and terminology With the exception of wp, functional application is denoted by juxtaposition and iterated functional composition by exponentiation.

"X is at least as strong as Y" means "$[X \Rightarrow Y]$".

"predicate transformer f is monotonic" means "$[X \Rightarrow Y] \Rrightarrow [fX \Rightarrow fY]$ for all X and Y".

"predicate transformer f is or-continuous" means "$[f(\underline{E}n: n \geq 0: R_n) \equiv (\underline{E}n: n \geq 0: f\, R_n)]$ for any weakening sequence of predicates R_n, i.e. such that $(\underline{A}n: n \geq 0: [R_n \Rightarrow R_{n+1}])$".

(End of Notation and Terminology.)

For monotonic predicate transformer f, the equation in X $[X \equiv fX]$ has a strongest solution (see [8]). If, in addition, f is or-continuous, its strongest solution is given in closed form by

(1) $(\underline{E}n: n \geq 0: f^n\, false)$.

Proof sketch Let Y be the strongest solution of $[X \equiv fX]$. Showing $[Y \equiv (\underline{E}n: n \geq 0: f^n \text{false})]$ is done by showing separately

(i) $[(\underline{E}n: n \geq 0: f^n \text{false}) \Rightarrow Y]$ and

(ii) $[Y \Rightarrow (\underline{E}n: n \geq 0: f^n \text{false})]$.

The proof of (i) is by mathematical induction on n and relies on the monotonicity of f; it uses that Y is a solution of $[X \equiv fX]$.

The proof of (ii) consists in showing that $(\underline{E}n: n \geq 0: f^n \text{false})$ is a solution of $[X \equiv fX]$ (of which Y is the strongest solution); it relies on the or-continuity of f (and on its monotonicity for the demonstration that the predicates $f^n \text{false}$ form a weakening sequence). (End of Proof Sketch.)

The predicate $wp(\mathbf{DO}, \neg B \wedge P)$ is (see [5]) defined as the strongest solution of (0). (Because $wp(S, X)$ is a monotonic function of X, so is the right-hand side of (0); hence its strongest solution exists.) If, in addition, $wp(S, X)$ is an or-continuous function of X, so is the right-hand side of (0), and an expression of the form of (1) gives a closed expression for $wp(\mathbf{DO}, \neg B \wedge P)$.

In programming terms, or-continuity of wp is the same as nondeterminacy being bounded. The assumption of bounded nondeterminacy is a usual one to make: the closed form for

$wp(DO, \neg B \wedge P)$, which is then available, is traditionally considered an advantage because it readily caters for the avoidance of fancy – and in practice cumbersome (see [1,2]) – techniques like transfinite induction.

Since unbounded nondeterminacy cannot be implemented, the restriction to or-continuity has for a long time been regarded as quite reasonable. It has, however, led to theorems in which the restriction to or-continuity has been introduced not because the theorems demanded it but for the sake of their proofs. The restriction also became a nuisance in the mathematical treatment of abstract programs. Firstly, an abstract program may well contain the as yet unrefined statement "establish P" where P, viewed as equation, may have infinitely many solutions, an observation we owe to [0]. Secondly, the modelling of concurrency as a "fair" interleaving of atomic actions introduces unbounded nondeterminacy (see [6]).

We are therefore very pleased to show for the main theorem about the repetitive construct an elementary proof that, though not relying on or-continuity, does not require transfinite formalisms.

The theorem

Notation In the sequel, x and y stand for elements from a set D. Set membership will be denoted by the infix operator "in"; our convention can thus be summarized by x in D and y in D. Function t is a mapping from the state space (of a program) to D, i.e. in each point of the state space the value of t is an element of D, a state of affairs that can be summarized by [t in D]. Let C be a subset of D; note that then t in C stands for a predicate that may be true in some points of the state space, and false in others. (End of Notation.)

For the notion "well-founded" we refer to the appendix in which we show that well-foundedness is the same as the validity of a proof by mathematical induction. (For that reason, the design of a well-founded set that does the job is a regularly recurring theme in arguments about algorithms.)

After the above preliminaries we are ready to formulate our -well-known-

Theorem Let $(D, <)$ be a partially ordered set; let C be a subset of D such that $(C, <)$ is well-founded; let statement S, predicates B and P, and function t on the state space satisfy

the predicate transformer wp(S,?) is monotonic;

$[t \text{ in } D]$;

(2) $[P \wedge B \Rightarrow t \text{ in } C]$;

(3) $[P \wedge B \wedge t=x \Rightarrow wp(S, P \wedge t<x)]$ for all x .

Then

(4) $[P \Rightarrow wp(DO, \neg B \wedge P)]$,

where $wp(DO, \neg B \wedge P)$ is defined as the strongest solution of (0).

In the above, the well-informed reader will recognize in P the "invariant" of the repetition and in t its "variant function", which is the vehicle for the termination argument.

The proof

The proof consists of three phases. In the first phase, we derive two lemmata we need for the justification of manipulations in the sequel. In the second phase, we massage the proof obligation so as to make it amenable to a proof using mathematical induction (since that is the only way of exploiting the well-foundedness of $(C,<)$). In the third phase, we carry out the inductive argument.

Conclusion (4) is proved by proving $[P \Rightarrow X]$ with X <u>any</u> solution of (0).

(i) To begin with we observe for any R

$$
\begin{array}{ll}
 & [P \wedge R \Rightarrow X] \qquad (*) \\
= & \{\text{since, from (0), } [X \equiv X \vee (\neg B \wedge P)]\} \\
 & [P \wedge R \Rightarrow X \vee (\neg B \wedge P)] \\
= & \{\text{predicate calculus}\} \\
 & [P \wedge (B \vee \neg P) \wedge R \Rightarrow X] \\
= & \{\text{predicate calculus}\} \\
 & [P \wedge B \wedge R \Rightarrow X] \qquad (**) \\
= & \{(2)\} \\
 & [P \wedge B \wedge t \underline{\text{in}} C \wedge R \Rightarrow X] \\
= & \{\text{Lemma 0 - see below - with } R := t \underline{\text{in}} C \wedge R\} \\
 & [P \wedge t \underline{\text{in}} C \wedge R \Rightarrow X] \quad . \qquad (***)
\end{array}
$$

From (*) and (**) we conclude

<u>Lemma 0</u> $[P \wedge R \Rightarrow X] \equiv [P \wedge B \wedge R \Rightarrow X]$ for any R.

From (*) and (***) we conclude

<u>Lemma 1</u> $[P \wedge R \Rightarrow X] \equiv [P \wedge t \underline{\text{in}} C \wedge R \Rightarrow X]$ for any R.

(ii) Next we massage our demonstrandum:

$$
\begin{array}{ll}
 & [P \Rightarrow X] \\
= & \{\text{Lemma 1 with } R := \text{true}\} \\
 & [P \wedge t \underline{\text{in}} C \Rightarrow X] \\
= & \{\text{predicate calculus}\} \\
 & [P \wedge (\underline{E}x : x \underline{\text{in}} C : t = x) \Rightarrow X]
\end{array}
$$

$= \{$definition of $\Rightarrow$ and de Morgan$\}$
$[\neg P \vee (\underline{A}x: x \underline{in} C: \neg t=x) \vee X]$
$= \{$disjunction distributes over universal quantification$\}$
$[(\underline{A}x: x \underline{in} C: \neg P \vee \neg t=x \vee X)]$
$= \{$definition of $\Rightarrow$ and de Morgan$\}$
$[(\underline{A}x: x \underline{in} C: P \wedge t=x \Rightarrow X)]$
$= \{$interchange of universal quantifications$\}$
(5) $(\underline{A}x: x \underline{in} C: [P \wedge t=x \Rightarrow X])$.

Remark As an aid to the reader, the above derivation has been carried out in smaller steps than usual. (End of Remark.)

In view of C's well-foundedness, (5) is proved by deriving -for any x in C-

(6) $[P \wedge t=x \Rightarrow X]$

under the hypothesis

(7) $(\underline{A}y: y \underline{in} C \wedge y<x: [P \wedge t=y \Rightarrow X])$.

(iii) To this end we observe:

(7)
$= \{$interchange of universal quantifications$\}$
$[(\underline{A}y: y \underline{in} C \wedge y<x: P \wedge t=y \Rightarrow X)]$
$= \{$range manipulation$\}$
$[(\underline{A}y: t=y: P \wedge y \underline{in} C \wedge y<x \Rightarrow X)]$
$= \{$one-point rule$\}$
$[P \wedge t \underline{in} C \wedge t<x \Rightarrow X]$
$= \{$Lemma 1 with $R := t<x\}$

$$
\begin{array}{ll}
 & [P \wedge t<x \Rightarrow X] \\
\Rightarrow & \{ wp(S,?) \text{ is monotonic} \} \\
 & [wp(S, P \wedge t<x) \Rightarrow wp(S,X)] \\
\Rightarrow & \{(3)\} \\
 & [P \wedge B \wedge t=x \Rightarrow wp(S,X)] \\
= & \{ \text{predicate calculus} \} \\
 & [P \wedge B \wedge t=x \Rightarrow B \wedge wp(S,X)] \\
= & \{ \text{predicate calculus} \} \\
 & [P \wedge (B \vee \neg P) \wedge t=x \Rightarrow B \wedge wp(S,X)] \\
= & \{ \text{predicate calculus} \} \\
 & [P \wedge t=x \Rightarrow (B \wedge wp(S,X)) \vee (\neg B \wedge P)] \\
= & \{(0)\} \\
 & [P \wedge t=x \Rightarrow X] \quad ,
\end{array}
$$

hence (6) has been derived under hypothesis (7).

Remark Also the above derivation has been given in smaller steps than usual. (End of Remark.)

Conclusion

To the best of our knowledge, Floyd (see [4]) has been the first one to formulate termination arguments in terms of well-founded sets; he did, however, restrict himself to deterministic programs, for which the natural numbers suffice.

In the fix-point theory that became en vogue during the seventies, continuity was strictly adhered to, with the result that again the natural numbers sufficed (see [7]).

To the best of our knowledge, the above argument is the first one to connect well-foundedness in its full generality directly to a non-operational notion of termination, i.e. to the strongest solution of a fix-point equation. Its simplicity should dispel the myth that the restriction to continuity for the sake of convenience is justified.

Finally we would like the reader to regard the effectiveness and austere rigour of our argument as a plea for the calculational proof method employed.

Acknowledgements

We are greatly indebted to C.A.R. Hoare for his formulation of antecedent (3), which he showed to lead to a shorter proof than our original formulation, to C.S. Scholten for simplifying the formulation of the induction hypothesis, and to the Tuesday Afternoon Club for strengthening the theorem by weakening antecedent (2).

A short appendix on well-foundedness

In the following,

$(C, <)$	is a partially ordered set,
x, y	are elements of C,
S	is a subset of C, and
Q	is a predicate on C,

where S and Q are coupled by

(8) $\quad Q.x \equiv \neg(x \in S)$, or $S = \{x \mid \neg Q.x\}$.

As a result we have

(9) $\quad S = \emptyset \equiv (\underline{A}\, x : x \in C : Q.x)$.

"x is a minimal element of S" means

$$x \in S \wedge (\underline{A}\, y : y < x : \neg(y \in S)) \quad .$$

"C is well-founded" means that any non-empty subset of C contains a minimal element.

We observe

"C is well-founded"
= {definitions of well-foundedness and minimal element}
$(\underline{A}\, S :: (S \neq \emptyset) \equiv$
$\quad (\underline{E}\, x : x \in C : x \in S \wedge (\underline{A}\, y : y < x : \neg(y \in S))))$
= {predicate calculus, de Morgan in particular}
$(\underline{A}\, S :: (S = \emptyset) \equiv$
$\quad (\underline{A}\, x : x \in C : \neg(x \in S) \vee (\underline{E}\, y : y < x : y \in S)))$
= {renaming the dummy with (8) and (9)}
$(\underline{A}\, Q :: (\underline{A}\, x : x \in C : Q.x) \equiv$
$\quad (\underline{A}\, x : x \in C : Q.x \vee (\underline{E}\, y : y < x : \neg Q.y)))$
= {definition of mathematical induction}
"mathematical induction over C is valid" .

Among mathematicians, well-foundedness is not as well-known as it deserves to be: there is, for instance, after half a century not yet a Dutch name for it. One cannot escape the impression

that Emmy Noether, besides being Jewish and female, had the additional disadvantage of having been preceded by Georg Cantor with his stress on countability.

References

[0] Back, R.J.R., Correctness preserving program refinements: proof theory and applications. Mathematical Centre Tracts nr. 131, Amsterdam 1980.

[1] Back, R.J.R., Proving Total Correctness of Nondeterministic Programs in Infinitary Logic. Acta Informatica vol. 15, Fasc. 3, 1981, pp. 233-249.

[2] Boom, H.J. A Weaker Precondition for Loops. ACM Transactions on Programming Languages and Systems, vol. 4, 1982, pp. 668-677.

[3] Dijkstra, Edsger W., A discipline of programming. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1976.

[4] Floyd, R.W., Assigning meanings to programs. Amer. Math. Soc. Symposia in Applied Mathematics, vol. 19, 1967, pp. 19-31.

[5] Park, David, On the semantics of fair parallelism. Lecture Notes in Computer Science, vol. 86, Springer-Verlag, Berlin-Heidelberg 1980, pp. 504 - 526.

[6] Park, David, Concurrency and automata on

infinite sequences. Lecture Notes in Computer Science, vol. 104, Springer-Verlag, Berlin-Heidelberg 1981.

[7] Stoy, J., Denotational Semantics: The Scott-Strachey Approach to Programming. MIT Press, Cambridge, MA, 1977.

[8] Tarski, A., A Lattice-theoretical Fixpoint Theorem and its Applications. Pacific Journal of Mathematics, vol. 5 (1955). pp. 285-309.

drs. A.J.M. van Gasteren
BP Venture Research Fellow
Dept. of Mathematics and Computing Science
University of Technology
5600 MB EINDHOVEN
The Netherlands

13 June 1984

prof.dr. Edsger W. Dijkstra
Burroughs Research Fellow
Plataanstraat 5
5671 AL NUENEN
The Netherlands