

Well-Founded Induction and the Invariance Theorem for Loops

Joseph M. Morris

Department of Computing Science
University of Glasgow
Glasgow G12 8QQ
Scotland, U.K.

1 Well-founded Induction

For C a set well-founded with respect to the partial ordering \leq , and for $Q.x$ a predicate possibly with free variable x , the theorem of well-founded induction states that $(\forall x: x \in C: Q.x)$ is equivalent to $(\forall x: x \in C: (\forall y: y \in C \wedge y < x: Q.y) \Rightarrow Q.x)$. Well-founded induction is very general. For example the so-called "principle of strong induction" is well-founded induction on the set of natural numbers with their usual \leq ordering, while so-called "transfinite induction" is well-founded induction when the ordering \leq is total.

Proofs using well-founded induction may not proceed as smoothly as one would like when the theorem to be proved is not in the neat form $(\forall x: x \in C: Q.x)$. We address this inconvenience in what follows by formulating well-founded induction for formulae of the shape $(\forall x: f.x \in C: Q.x)$ where C is a well-founded set and f is any function (whose range may or may not be a subset of C). We will use the result to make a short formula-driven proof of the fundamental invariance theorem for loops, and indeed this is the immediate motivation for the work.

Let (C, \leq) abbreviate "set C partially ordered by \leq ", and define $x < y$ to be $x \leq y \wedge x \neq y$, for all x, y and any partial ordering \leq . " (C, \leq) is well-founded" means that for each subset S of C the following holds: $S \neq \emptyset \equiv (\exists k: k \in S: (\forall i: i \in S: \neg i < k))$.

Theorem 1. Let (C, \leq) be well-founded and $P.x$ be a predicate; then

$$(\forall x: x \in C: P.x) \equiv (\forall x: x \in C: (\forall y: y \in C \wedge y < x: P.y) \Rightarrow P.x).$$

Proof. This is the standard well-founded induction theorem proved, for example, in [2]. (End)

Theorem 2. Let (C, \leq) be well-founded, $Q.x$ be a predicate, and $f.x$ be a function; then

$$(\forall x: f.x \in C: Q.x) \equiv (\forall y: y \in C: (\forall x: f.x \in C \wedge f.x < y: Q.x) \Rightarrow (\forall x: f.x = y: Q.x)).$$

Proof.

$$\begin{aligned} & (\forall x: f.x \in C: Q.x) \\ = & \text{"1-point rule"} \\ & (\forall x: (\exists y: y = f.x: y \in C): Q.x) \\ = & \text{"eliminate ranges"} \\ & (\forall x: (\exists y: y = f.x \wedge y \in C) \Rightarrow Q.x) \\ = & \text{"remove } \Rightarrow \text{; extend scope of } y\text{"} \\ & (\forall x: (\forall y: y \neq f.x \vee y \notin C \vee Q.x)) \\ = & \text{"rearrange quantifiers; introduce ranges"} \\ & (\forall y: y \in C: (\forall x: y = f.x: Q.x)) \\ = & \text{"Theorem 1 with } P.y := (\forall x: y = f.x: Q.x)\text{"} \\ & (\forall y: y \in C: (\forall x: x \in C \wedge x < y: P.x) \Rightarrow (\forall x: y = f.x: Q.x)) \end{aligned}$$

Comparing the preceding line with the right-hand side of the demonstrandum we see that the proof is completed by showing the equivalence of $(\forall x: x \in C \wedge x < y: P.x)$ and $(\forall x: f.x \in C \wedge f.x < y: Q.x)$. Proceeding with this:

$$\begin{aligned} & (\forall x: x \in C \wedge x < y: P.x) \\ = & \text{"P.x"} \\ & (\forall x: x \in C \wedge x < y: (\forall z: x = f.z: Q.z)) \\ = & \text{"unnesting"} \\ & (\forall z, x: x \in C \wedge x < y \wedge x = f.z: Q.z) \\ = & \text{"range and term manipulation"} \\ & (\forall z: (\forall x: x = f.z: x \in C \wedge x < y \Rightarrow Q.z)) \\ = & \text{"1-point"} \\ & (\forall z: f.z \in C \wedge f.z < y \Rightarrow Q.z) \\ = & \text{"rename dummy; introduce range"} \\ & (\forall x: f.x \in C \wedge f.x < y: Q.x) \qquad \text{(End)} \end{aligned}$$

We could derive Theorem 1 from Theorem 2 by taking for f in Theorem 2 the identity function on C ; it follows that the two theorems are equivalent. The following minor variation on Theorem 2 will be useful later:

Theorem 3. Let (C, \leq) be well-founded $P.x$ and $Q.x$ be predicates, and $f.x$ be a function; then

$$\begin{aligned} & (\forall x: P.x \wedge f.x \in C: Q.x) \equiv \\ & (\forall y: y \in C: (\forall x: P.x \wedge f.x \in C \wedge f.x < y: Q.x) \Rightarrow (\forall x: P.x \wedge f.x = y: Q.x)). \end{aligned}$$

Proof: Rewrite $(\forall x: P.x \wedge f.x \in C: Q.x)$ as $(\forall x: f.x \in C: P.x \Rightarrow Q.x)$ and apply the preceding theorem. (End)

In the calculational approach to proofs one of the central strategies is that of reducing the demonstrandum to 'true' using equivalence and reverse implication \Leftarrow (read "follows from"). We propose a style for accommodating well-founded induction, in the form of Theorem 2, in calculational proofs. Faced with proving $(\forall x: f.x \in C: Q.x)$ we proceed thus (explanations follow):

$$\begin{aligned} & (\forall x: f.x \in C: Q.x) \\ \Leftarrow & \text{"induction hypothesis; } y:- y \in C \wedge (\forall x: f.x \in C \wedge f.x < y: Q.x)\text{"} \\ & (\forall x: f.x = y: Q.x) \\ \Leftarrow & \bullet \end{aligned}$$

•
•
 $(\forall x: f.x \in C \wedge f.x < y: Q.x)$
="induction hypothesis"
true.

The first proof step in the above is justified thus:

$(\forall x: f.x \in C: Q.x)$
="Theorem 2"
 $(\forall y: y \in C: (\forall x: f.x \in C \wedge f.x < y: Q.x) \Rightarrow (\forall x: f.x = y: Q.x))$
="eliminate range of y"
 $(\forall y:: y \in C \wedge (\forall x: f.x \in C \wedge f.x < y: Q.x) \Rightarrow (\forall x: f.x = y: Q.x))$
<"generalization over y — see Note (i) below"
 $y \in C \wedge (\forall x: f.x \in C \wedge f.x < y: Q.x) \Rightarrow (\forall x: f.x = y: Q.x)$
<"assume $y \in C \wedge (\forall x: f.x \in C \wedge f.x < y: Q.x)$ — see Note (ii) below"
 $(\forall x: f.x = y: Q.x)$

Note (i). The final step but one in the above records an appeal to the rule of generalization of predicate calculus which allows us to conclude $(\forall x::Q.x)$ from a proof of $Q.x$ for any $Q.x$.

Note (ii). The final step in the above appeals to the deduction theorem of predicate calculus which says we may prove $Q.y \Rightarrow R.y$, for any $Q.y$ and $R.y$, by making a proof of $R.y$ in which we may appeal to the "dropped antecedent" $Q.y$. (There is a technicality here in that in the proof of $R.y$ applications of generalization may not involve free variables occurring in $Q.y$, but this will not concern us further.) The dropped antecedent as it arises in the application of induction we call the "induction hypothesis".

Similar remarks on proof style hold for the forms of well-founded induction in Theorems 1 and 3. Note also that each of the three theorems can be trivially reformulated by eliminating ranges, i.e. by replacing each formula of the shape $(\forall x: R.x: S.x)$ by $(\forall x:: R.x \Rightarrow S.x)$. In the proof of the invariance theorem below we will be employing the "rangeless" formulation of Theorem 2.

2 The invariance theorem for loops

We now use well-founded induction in a proof of the fundamental invariance theorem for loops [3]. A loop has the form **do** $B \rightarrow s$ **od** where B stands for a predicate on the state space of the program and s stands for a statement; we let DO denote this loop. $wp(DO, P \wedge \neg B)$, i.e. the weakest precondition for DO to terminate in a state satisfying $P \wedge \neg B$, is the (strongest — but we do not need that here) predicate Z satisfying $[Z \equiv (B \wedge wp(s,Z)) \vee (P \wedge \neg B)]$ where $[X]$ denotes universal closure of predicate X over its free program variables. It is known [3] that $wp(t,?)$ is monotonic for all statements t , i.e. $[X \Rightarrow Y] \Rightarrow [wp(t,X) \Rightarrow wp(t,Y)]$ for all predicates X, Y .

Invariance Theorem. Let DO denote **do** $B \rightarrow s$ **od** for B any predicate on the state space of the program and s any statement. Let t be a function on the state space taking values in some set D , i.e. $[t \in D]$. Let D be partially ordered by \leq and C be a subset of D such that (C, \leq) is well-founded.

If $[P \wedge B \Rightarrow t \in C]$ (i)
and $(\forall y: y \in C: [P \wedge B \wedge t=y \Rightarrow wp(s, P \wedge t < y)])$ (ii)
then $[P \Rightarrow wp(DO, P \wedge \neg B)]$

Proof. Let predicate Z denote $wp(DO, P \wedge \neg B)$; then Z satisfies $[Z \equiv (B \wedge wp(s,Z)) \vee (P \wedge \neg B)]$. We will save ourselves a little repetition if we first show that for any predicate X

$[P \wedge X \Rightarrow Z] \equiv [P \wedge B \wedge t \in C \wedge X \Rightarrow wp(s, Z)]$ (iii)

The proof of (iii) is

[$P \wedge X \Rightarrow Z$]
 ="Z"
 [$P \wedge X \Rightarrow (B \wedge \text{wp}(s,Z)) \vee (P \wedge \neg B)$]
 ="predicate calculus"
 [$P \wedge X \Rightarrow (B \wedge \text{wp}(s,Z)) \vee \neg B$]
 ="predicate calculus"
 [$P \wedge X \Rightarrow \text{wp}(s,Z) \vee \neg B$]
 ="predicate calculus"
 [$P \wedge B \wedge X \Rightarrow \text{wp}(s,Z)$]
 ="(i); predicate calculus"
 [$P \wedge B \wedge t \in C \wedge X \Rightarrow \text{wp}(s,Z)$]
 The proof of the theorem is now
 [$P \Rightarrow \text{wp}(DO, P \wedge \neg B)$]
 ="Z"
 [$P \Rightarrow Z$]
 ="(iii) with $X := \text{true}$ "
 [$P \wedge B \wedge t \in C \Rightarrow \text{wp}(s, Z)$]
 ="induction hypothesis, $y:- y \in C \wedge [P \wedge B \wedge t \in C \wedge t < y \Rightarrow \text{wp}(s, Z)]$ "
 [$P \wedge B \wedge t = y \Rightarrow \text{wp}(s, Z)$]
 \Leftarrow "(ii); $y \in C$ "
 [$\text{wp}(s, P \wedge t < y) \Rightarrow \text{wp}(s, Z)$]
 \Leftarrow "wp(s,?) monotonic"
 [$P \wedge t < y \Rightarrow Z$]
 ="(iii) with $X := t < y$ "
 [$P \wedge B \wedge t \in C \wedge t < y \Rightarrow \text{wp}(s, Z)$]
 ="induction hypothesis"
 true (End)

The proof of the preceding theorem depends not at all on the introduction of (iii) which serves only to shorten the proof by a few steps and was introduced in a minor polishing of our first attempt. See [1,2,3] for other proofs of the theorem; the present proof was motivated by a desire to make a shorter and more formula-driven proof than that in [2].

To summarize, we have formulated well-founded induction for formulae of the shape $(\forall x: f.x \in C: Q.x)$, shown a style for using it in calculational proofs, and used it to make a short proof of the invariance theorem.

References

- [1] H. J. Boom, A weaker precondition for loops, ACM Trans. Prog. Lan. and Sys. 4 (4) (1982) 668-677.
- [2] E. W. Dijkstra and A. J. M. van Gasteren, A simple fixpoint argument without the restriction to continuity, Acta Informatica 23(1) (1986) 1-7.
- [3] E. W. Dijkstra, A Discipline of Programming (Prentice-Hall, Englewood Cliffs, NJ, 1976).