On fixpoint/iteration/variant induction principles for proving total correctness of programs with denotational semantics

Patrick Cousot

Courant Institute of Mathematical Sciences, New York University
and IMDEA Software Institute

Abstract. We study partial and total correctness proof methods based on generalized fixpoint/iteration/variant induction principles applied to the denotational semantics of first-order functional and iterative programs.

Keywords: Induction principles · Denotational semantics · Partial and total correctness · Verification

1 Introduction

Imperative and functional programming are very often separate worlds, even in languages like OCaml [18] which combines both styles. Most programmers definitely prefer one style to the other. This reflects in semantics mostly denotational for functional and operational for imperative. This also reflects on verification, mostly Turing/Floyd/Naur/Hoare for invariance and Turing/Floyd/Manna-Pnueli variant/convergence function for termination of imperative languages while Scott proof method is preferred for functional programming.

We show that after appropriate generalization the principles underlying the verification of these programming styles boils down to the same unified verification (hence analysis) methods.

2 Basic notions in denotational semantics

The denotational semantics of first-order functions $f \in \mathcal{D} \rightarrow \mathcal{D}_\perp$ uses a complete partial order (cpo) $\langle \mathcal{D}_\perp, \sqsubseteq, \perp, \sqcup \rangle$ where $\perp$ denotes non-termination and $\mathcal{D}_\perp = \mathcal{D} \cup \{\perp\}$ is the flat domain ordered by $\perp \sqsubseteq \perp \sqsubsetneq d \sqsubseteq d$ for all $d \in \mathcal{D}$. $\sqcup$ is the least upper bound (lub) in $\mathcal{D}_\perp$. This is extended pointwise to $\langle \mathcal{D} \rightarrow \mathcal{D}_\perp, \dot{\sqsubseteq}, \dot{\perp}, \dot{\sqcup} \rangle$ by $f \mathrel{\dot{\sqsubseteq}} g$ if and only if $\forall d \in \mathcal{D} \mathrel{.} f(d) \sqsubseteq g(d)$, $\dot{\perp} \triangleq \lambda\, x \bullet \perp$, and $\dot{\bigsqcup\limits_{i \in \Delta}} f_i \triangleq \lambda\, x \bullet \bigsqcup\limits_{i \in \Delta} f_i(x)$. First-order functions f are defined recursively $f(x) = F(f)x$ as least fixpoints $f = \mathsf{lfp}^{\dot{\sqsubseteq}}\, F$ of continuous transformers $F \in (\mathcal{D} \rightarrow \mathcal{D}_\perp) \xrightarrow{uc} (\mathcal{D} \rightarrow \mathcal{D}_\perp)$. The iterates of F from f are $F^0(f) = f$ and $F^{i+1}(f) = F(F^i(f))$. F is continuous if and only iff for every denumerable increasing chain $f_0 \mathrel{\dot{\sqsubseteq}} f_1 \mathrel{\dot{\sqsubseteq}} \ldots \mathrel{\dot{\sqsubseteq}} f_i \mathrel{\dot{\sqsubseteq}} \ldots$, $\dot{\bigsqcup\limits_{i \in \mathbb{N}}} F(f_i) = F(\dot{\bigsqcup\limits_{i \in \mathbb{N}}} f_i)$. Continuity implies monotonically increasing ($f \mathrel{\dot{\sqsubseteq}} g \Rightarrow F(f) \mathrel{\dot{\sqsubseteq}} F(g)$). Since $F^0(\dot{\perp}) = \dot{\perp}$ and F is monotonically increasing, it follows that the iterates of F from $\dot{\perp}$ form an increasing chain. Then continuity guarantees that $\mathsf{lfp}^{\dot{\sqsubseteq}}\, F = \dot{\bigsqcup\limits_{i \in \mathbb{N}}} F^i(\dot{\perp})$ is the limit of the iterates $F^i(\dot{\perp})$ of F from $\dot{\perp}$. By def. of $\dot{\sqsubseteq}$ and $\dot{\sqcup}$, $(\mathsf{lfp}^{\dot{\sqsubseteq}}\, F)x = y$ if and only if $\exists i \in \mathbb{N} \mathrel{.} (\forall j < i \mathrel{.} F^j(\dot{\perp})(x) = \perp) \wedge (\forall j \geqslant i \mathrel{.} F^j(\dot{\perp})(x) = y)$.

Example 1 (while iteration). The iteration `W = while (B) S` operating on a vector $x \in \mathcal{D}$ of values of variables has denotational semantics $[\![\mathtt{W}]\!] = \mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}}$ where $F_{\mathtt{W}}(f)x = \llparenthesis \neg B(x) \mathrel{?} x \mathrel{:} f(S(x)) \rrparenthesis$, $B \in \mathcal{D} \rightarrow \{\mathtt{tt}, \mathtt{ff}\}$ is the semantics of boolean expression `B`, $S \in \mathcal{D} \rightarrow \mathcal{D}_{\perp}$ that of statement `S` (which, by structural induction, may contain conditionals and inner loop), and $\llparenthesis \mathtt{tt} \mathrel{?} a \mathrel{:} b \rrparenthesis = a$ and $\llparenthesis \mathtt{ff} \mathrel{?} a \mathrel{:} b \rrparenthesis = b$ is the conditional. The iterates of $F_{\mathtt{W}}$ from $\perp$ are

$$F_{\mathtt{W}}^{0}(\perp)x = \perp$$

$$F_{\mathtt{W}}^{1}(\perp)x = F_{\mathtt{W}}(F_{\mathtt{W}}^{0}(\perp))x \;=\; \llparenthesis \neg B(x) \mathrel{?} x \mathrel{:} \perp \rrparenthesis$$

$$\begin{aligned}F_{\mathtt{W}}^{2}(\perp)x &= F_{\mathtt{W}}(F_{\mathtt{W}}^{1}(\perp))x \;=\; \llparenthesis \neg B(x) \mathrel{?} x \mathrel{:} F_{\mathtt{W}}^{1}(\perp)(S(x)) \rrparenthesis \;=\; \llparenthesis \neg B(x) \mathrel{?} x \mathrel{:} \llparenthesis \neg B(S(x)) \mathrel{?} S(x) \mathrel{:} \perp \rrparenthesis \rrparenthesis\\ &= \llparenthesis \neg B(x) \mathrel{?} x \mathrel{:} \perp \rrparenthesis \sqcup \llparenthesis B(x) \wedge \neg B(S(x)) \mathrel{?} S(x) \mathrel{:} \perp \rrparenthesis\end{aligned}$$

...

$$F_{\mathtt{W}}^{n}(\perp)x = \bigsqcup_{i=0}^{n-1} \llparenthesis \bigwedge_{j=0}^{i-1} B(S^{j}(x)) \wedge \neg B(S^{i}(x)) \mathrel{?} S^{i}(x) \mathrel{:} \perp \rrparenthesis \;\wr\text{where } S^{0}(x) \triangleq x,\ S^{i+1}(x) \triangleq S(S^{i}(x)), \text{ and } \bigwedge \varnothing = \mathtt{tt}\wr$$

...

$$\begin{aligned}(\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})x &= \bigsqcup_{n \in \mathbb{N}} F_{\mathtt{W}}^{n}(\perp)x \;=\; \bigsqcup_{n \in \mathbb{N}} \bigsqcup_{i=0}^{n-1} \llparenthesis \bigwedge_{j=0}^{i-1} B(S^{j}(x)) \wedge \neg B(S^{i}(x)) \mathrel{?} S^{i}(x) \mathrel{:} \perp \rrparenthesis \qquad \wr\text{where } \bigsqcup \varnothing = \perp\wr\\ &= \bigsqcup_{n \in \mathbb{N}} \llparenthesis \bigwedge_{j=0}^{n-1} B(S^{j}(x)) \wedge \neg B(S^{n}(x)) \mathrel{?} S^{n}(x) \mathrel{:} \perp \rrparenthesis\end{aligned}$$

Note that in the lub, at most one condition is true, none if the iteration does not terminate. Moreover, if $(\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})x \neq \perp$, then, by def. $\sqcup$, $\exists j \in \mathbb{N} \,.\, (\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})x = F_{\mathtt{W}}^{j}(\perp)x$ and so $\neg B(FW^{j}(\perp)x)$ holds proving $\neg B(\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})$. □

3 Termination specification

The termination of function $f \in \mathcal{D} \rightarrow \mathcal{D}_{\perp}$ on a termination domain $T \in \wp(\mathcal{D})$ can be specified as $f \in \mathcal{P}_T$ where $\mathcal{P}_T \triangleq \{f \mid \forall x \in T \,.\, f(x) \neq \perp\}$. So $\mathcal{P}_T$ is the property of functions that terminate on domain T.

Example 2 (termination). For imperative program, the termination problem is usually solved by the Turing [29]/Floyd [12]/Manna-Pnueli [20] variant/convergence function method. For first-order functions, one can consider Jones size-change termination method [13,17]. □

4 Fixpoint induction principle

In case $\langle \mathcal{D}_{\perp}, \sqsubseteq, \perp, \sqcup, \sqcap \rangle$ is a complete lattice (*e.g.* by adding a supremum $\top$ as in Scott's original papers [27]), we can make proofs by fixpoint induction. [7, 3.4.1] and [23, (2.3)] observed that fixpoint induction directly follows from Tarski's fixpoint theorem [28].

Theorem 1 (Tarski fixpoint theorem [28]) *A monotonically increasing function* $F \in L \xrightarrow{\nearrow} L$ *on a complete lattice* $\langle L, \sqsubseteq, \perp, \top, \sqcap, \sqcup \rangle$ *has a least fixpoint* $\mathsf{lfp}^{\sqsubseteq} F = \bigsqcap\{x \in L \mid F(x) \sqsubseteq x\}$.

Fixpoint induction relies on properties of F above its least fixpoint *i.e.* the $x \in L$ such that $F(x) \sqsubseteq x$ and therefore $\mathsf{lfp}^{\sqsubseteq} F \sqsubseteq x$.

Theorem 2 (Fixpoint induction) *Let $F \in \mathcal{L} \xrightarrow{\nearrow} \mathcal{L}$ be a monotonically increasing function on a complete lattice $\langle \mathcal{L}, \sqsubseteq, \bot, \top, \sqcap, \sqcup \rangle$ and $P \in \mathcal{L}$. We have*

$$\mathsf{lfp}^{\sqsubseteq} F \sqsubseteq P \Leftrightarrow \exists I \in \mathcal{L} \,.\, \begin{array}{ll} & F(I) \sqsubseteq I \qquad\qquad (2.\mathrm{a}) \\ \wedge & I \sqsubseteq P \qquad\qquad\quad (2.\mathrm{b}) \end{array}$$ □

$J \in \mathcal{L}$ is called an *invariant* of F when $\mathsf{lfp}^{\sqsubseteq} F \sqsubseteq J$ and an *inductive invariant* when satisfying $F(J) \sqsubseteq J$.

Soundness ($\Leftarrow$) states that if a statement is proved by the proof method then that statement is true. Completeness ($\Rightarrow$) states that the proof method is always applicable to prove a true statement.

Proof (of Th. 2). By Tarski fixpoint Th. 1, $\mathsf{lfp}^{\sqsubseteq} F = \bigsqcap\{x \in L \mid F(x) \sqsubseteq x\}$.

Soundness ($\Leftarrow$): If $I \in \mathcal{L}$ satisfies $F(I) \sqsubseteq I$ then $I \in \{x \in L \mid F(x) \sqsubseteq x\}$ so by definition of the glb $\bigsqcap$, $\mathsf{lfp}^{\sqsubseteq} F = \bigsqcap\{x \in L \mid F(x) \sqsubseteq x\} \sqsubseteq I \sqsubseteq P$ by (2.b).

Completeness ($\Rightarrow$): If $\mathsf{lfp}^{\sqsubseteq} F \sqsubseteq P$ then take $I = \mathsf{lfp}^{\sqsubseteq} F$ then $I = F(I)$ so $F(I) \sqsubseteq I$ by reflexivity and $I \sqsubseteq P$ by hypothesis, proving $\exists I \in \mathcal{L} \,.\, F(I) \sqsubseteq I \wedge I \sqsubseteq P$. □

Usually, proofs are done using logics of limited expressive power so completeness is relative to the existence of a logic formula expressing the stronger invariant $I = \mathsf{lfp}^{\sqsubseteq} f$ [5,6]. In Th. 2, we consider invariants to be sets in order to make expressivity a separate problem.

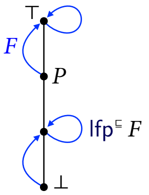
The fixpoint induction principle Th. 2 has been used to justify invariance proof methods for small-step operational semantics/transition systems, including their contrapositive, backward, *etc.* variants [9]. It can also be used with a denotational semantics.

Example 3 (Partial correctness of the factorial). Define $F_!(f) \triangleq \lambda n \bullet [\![n = 0 \mathrel{?} 1 \mathbin{:} n \times f(n-1)]\!]$. Let us prove that $\mathsf{lfp}^{\dot\sqsubseteq} F_! \mathrel{\dot\sqsubseteq} f_! \triangleq \lambda n \bullet [\![x \geqslant 0 \mathrel{?} n! \mathbin{:} \bot]\!]$ where $n!$ is the mathematical factorial function. Applying Th. 2 with $I = P = f_!$ so that (2.b) holds, we have

$$\begin{array}{lll} & F_!(f_!)n & \\ = & [\![n = 0 \mathrel{?} 1 \mathbin{:} n \times f_!(n-1)]\!] & \wr\text{def. } F_!\int \\ = & [\![n = 0 \mathrel{?} f_!(n) \mathbin{:} f_!(n)]\!] & \wr\text{def. } f_!\int \\ \sqsubseteq & f_!(n) & \wr\text{def. conditional and } \sqsubseteq \text{ reflexive}\int \end{array}$$

so $F_!(f_!) \mathrel{\dot\sqsubseteq} f_!$ by pointwise def. of $\dot\sqsubseteq$, proving (2.a). By definition of $\dot\sqsubseteq$, we have $\forall n \in \mathbb{Z} \,.\, (\mathsf{lfp}^{\dot\sqsubseteq} F_!)n \neq \bot \Rightarrow \mathsf{lfp}^{\dot\sqsubseteq} F_!(n) = f_!(n)$ *i.e.* if a call $(\mathsf{lfp}^{\dot\sqsubseteq} F_!)n$ terminates then it returns $n!$. Obviously this is a partial correctness proof since *e.g.* the proof does not exclude that $\mathsf{lfp}^{\dot\sqsubseteq} F_! = \lambda n \bullet \bot \mathrel{\dot\sqsubseteq} f_!$! □

Notice that if $P = \mathsf{lfp}^{\sqsubseteq} f$, fixpoint induction requires to prove that $f(\mathsf{lfp}^{\sqsubseteq} f) \sqsubseteq \mathsf{lfp}^{\sqsubseteq} f$ and $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P$. So to prove $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P$, we have to prove $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq P$! In that case fixpoint induction cannot help. In general, we have to prove $\mathsf{lfp}^{\sqsubseteq} F \sqsubsetneq P$ but nevertheless the only inductive invariant might be $\mathsf{lfp}^{\sqsubseteq} F$, as shown below where P is not inductive.



In such cases fixpoint induction is not useful but it is possible to reason on the iterates of F, as shown in Sect. 8.

5 Impossibility to prove termination by fixpoint induction with a denotational semantics

One can use a function $P \in \mathcal{D} \rightarrow \mathcal{D}_\perp$ to specify a termination domain $\mathsf{dom}(P) \triangleq \{x \in \mathcal{D} \mid P(x) \neq \perp\}$. However, by definition of $\dot{\sqsubseteq}$, $\mathsf{lfp}^{\dot{\sqsubseteq}}\, F \mathrel{\dot{\sqsubseteq}} P$ means that $\mathsf{lfp}^{\dot{\sqsubseteq}}\, F$ terminates less often that P that is $\mathsf{dom}(\mathsf{lfp}^{\dot{\sqsubseteq}}\, F) \subseteq \mathsf{dom}(P)$. This is not a specification of definite termination but of definite non-termination. So fixpoint induction can be used to prove non-termination but not termination. Of course $P \mathrel{\dot{\sqsubseteq}} \mathsf{lfp}^{\dot{\sqsubseteq}}\, F$ would do but this is not what fixpoint induction is intended to prove. Considering the order-dual of Th. 2 will not work either (although it would work for greatest fixpoints) since, in general, $\mathsf{gfp}^{\dot{\sqsubseteq}}\, F \neq \mathsf{lfp}^{\dot{\sqsubseteq}}\, F$.

Example 4 (Termination/total correctness of the factorial). Continuing Ex. 3, termination of the factorial $\mathsf{lfp}^{\dot{\sqsubseteq}}\, F_!$ where $F_!(f) \triangleq \lambda n \bullet \llbracket n = 0 \mathrel{?} 1 \mathrel{:} n \times f(n-1) \rrbracket$ is $f_! \mathrel{\dot{\sqsubseteq}} \mathsf{lfp}^{\dot{\sqsubseteq}}\, F_!$ where $f_! \triangleq \lambda n \bullet \llbracket x \geqslant 0 \mathrel{?} n! \mathrel{:} \perp \rrbracket$ but this is not provable by fixpoint induction Th. 2. □

6 Iteration induction principle

As observed by [19,21,26], iteration induction directly follows from Kleene/Scott's fixpoint theorem below (which we used in Sect. **2** with $\mathcal{L} = \mathcal{D} \rightarrow \mathcal{D}_\perp$). (Th. 3 is often attributed to Stephen Cole Kleene, after its first recursion theorem [16, p. 348] and appears in [2].)

Theorem 3 (Kleene/Scott iterative fixpoint theorem [26]) *If $F \in \mathcal{L} \xrightarrow{uc} \mathcal{L}$ is an upper continuous function on a cpo $\langle \mathcal{L}, \sqsubseteq, \perp, \sqcup \rangle$ then F has a least fixpoint $\mathsf{lfp}^{\sqsubseteq}\, F = \bigsqcup_{n \in \mathbb{N}} F^n(\perp)$.*

Since $F^0(\perp) = \perp$ is the infimum and F is upper continuous hence monotonically increasing, the iterates $\langle F^n(\perp),\ n \in \mathbb{N} \rangle$ form a non-empty, infinite, denumerable, and maximally increasing chain which is either first strictly increasing and then stationary (when the iterates converge in finitely many steps) or else is strictly increasing.

Remark 1. Th. 3 generalizes to chain-α-complete posets (where every α-chain that is increasing chain of cardinality less than or equal to α has a lub) and α-continuous functions (preserving the lubs of α-chains), and to monotonically increasing functions on complete lattices, using transfinite iterations $F^0(\perp) = \perp$, $F^{\delta+1} = F(F^\delta)$ for successor ordinals and $F^\lambda = \bigsqcup_{\delta<\lambda} F^\delta$ for limit ordinals less than or equal to α [22], respectively all ordinals [8]. Th. 3 is then a corollary for the first infinite ordinal $\alpha = \omega$. □

Iteration induction relies on properties of F below its least fixpoint. It is usually referred to as Scott induction or De Bakker and Scott or computational induction and formalized as

$$\text{“If } \mathcal{P} \in \wp(\mathcal{D}) \text{ is an admissible predicate, } \perp \in \mathcal{P}, \text{ and } \forall d \in \mathcal{P} \mathrel{.} F(d) \in \mathcal{P} \text{ then } \mathsf{lfp}^{\sqsubseteq}\, F \in \mathcal{P}\text{”.} \tag{4}$$

The predicate $\mathcal{P}$ is said to be admissible [19] or inclusive [25, p. 118] if and only if it holds for an increasing enumerable chain, it also holds for its limit, that is for all increasing enumerable chains $F_0 \sqsubseteq F_1 \sqsubseteq \ldots \sqsubseteq F_i \sqsubseteq \ldots$, if $\forall i \in \mathbb{N} \mathrel{.} F_i \in \mathcal{P}$ then $\bigsqcup_{i \in \mathbb{N}} F_i \in \mathcal{P}$.

7 Impossibility to prove termination by iteration induction

The termination specification of functions $f \in \mathcal{D} \rightarrow \mathcal{D}_\perp$ is the set $\mathcal{P} \triangleq \{f \in \mathcal{D} \rightarrow \mathcal{D}_\perp \mid \forall x \in \mathcal{D} \,.\, f(x) \in \mathcal{D}\} = \mathcal{D} \rightarrow \mathcal{D}$ of all functions that always terminate on the domain $\mathcal{D}$ of their argument. To prove $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}$ by structural induction (4) requires $\perp \in \mathcal{P}$, which is not true since, unless $\mathcal{D} = \emptyset$, $\forall x \in \mathcal{D} \,.\, \perp(x) = \perp \in \mathcal{D}$ is false. So Scott's iteration induction principle is incomplete since it cannot be used to prove termination.

8 Generalized iteration induction principle

This incompleteness calls for a generalization of iteration induction where the characterization $\mathcal{Q}$ of the iterations differs from that of their limit $\mathcal{P}$.

Example 5. For the factorial of Ex. 3, the iterates $F_!^n(\perp)$, $n \in \mathbb{N}$ are partial functions (characterized by $\mathcal{Q}$) while the limit $f_!$ is a total function on $\mathcal{D} = \mathbb{N}$ (characterized by $\mathcal{P}$). □

Let $F \in S \rightarrow S$ and $\langle x_i,\ i \in \Delta\rangle$ be a family of elements of S. The family is *non-empty* if and only if $\Delta \neq \emptyset$. It is *infinite* when the cardinality of Δ is greater than or equal to that of $\mathbb{N}$. It is *denumerable* if and only if $\Delta \subseteq \mathbb{N}$ (up to an isomorphism). It is a $\sqsubseteq$*-increasing chain* if and only if $\forall i, j \in \Delta \,.\, (i \leqslant j) \Rightarrow (x_i \sqsubseteq x_j)$. It is a *strictly increasing chain* if and only if $\forall i, j \in \Delta \,.\, (i < j) \Rightarrow (x_i \sqsubsetneq x_j)$. It is *in* $S' \subseteq S$ if and only if $\forall i \in \Delta \,.\, x_i \in S'$. The sequence $\langle x_i,\ i \in \mathbb{N}\rangle$ is F*-maximally increasing* when it is infinite (hence non-empty), denumerable, iterating F (*i.e.* $\forall i \in \mathbb{N} \,.\, x_{i+1} = F(x_i)$), and either strictly increasing (*i.e.* $\forall i, j \in \mathbb{N} \,.\, (i < j) \Rightarrow (x_i \sqsubsetneq x_j)$) or is first strictly increasing and then stationary (*i.e.* $\exists k \in \mathbb{N} \,.\, \forall i, j \in \mathbb{N} \,.\, (i < j \leqslant k) \Rightarrow (x_i \sqsubsetneq x_j) \wedge (k \leqslant i) \Rightarrow (x_k = x_i)$).

Theorem 5 (Iteration induction) *Let $F \in \mathcal{L} \xrightarrow{uc} \mathcal{L}$ be a continuous function on a cpo $\langle\mathcal{L}, \sqsubseteq, \perp, \sqcup\rangle$ and $\mathcal{P} \in \wp(\mathcal{L})$.*

$$\begin{array}{lll}
\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P} \Leftrightarrow \exists \mathcal{Q} \in \wp(\mathcal{L}) \,. & \perp \in \mathcal{Q} & (5.\text{a}) \\
\wedge & \forall x \in \mathcal{Q} \,.\, F(x) \in \mathcal{Q} & (5.\text{b}) \\
\wedge & \textit{for any } F\textit{-maximally } \sqsubseteq\textit{-increasing chain } \langle x_i,\ i \in \mathbb{N}\rangle \textit{ in } \mathcal{Q}, & (5.\text{c}) \\
& \qquad \displaystyle\bigsqcup_{i \in \mathbb{N}} x_i \in \mathcal{P} & \square
\end{array}$$

The proof below shows that the hypotheses (a), (b), and (c) are necessary only for the iterates of F. The soundness proof shows that $\mathcal{Q}$ is a valid property of the iterates of F from $\perp$ while $\mathcal{P}$ is a property of their least upper bound, that is of the fixpoint. Offering the possibility of choosing $\mathcal{Q} \neq \mathcal{P}$ is essential to solve the incompleteness problem of Scott induction (4) mentioned in the above Sect. **7**. But of course Th. 5 can be used with $\mathcal{Q} = \mathcal{P}$ so that it is a generalization of Scott induction (4) and a proof that (4) is sound.

Condition (5.c) corresponds to the "admissible predicates" in Scott induction. However, (5.c) is requested for maximally $\sqsubseteq$-increasing chains only since requiring it for all increasing chains would amount to Scott induction (4). The quantification over chains iterating F in (5.c) can be relaxed since this condition could also be imposed by an appropriate choice of $\mathcal{Q}$.

Proof (of Th. 5). Soundness ($\Leftarrow$): Let $F^{i+1}(\perp) = F(F^i(\perp))$ be the iterates of F from $F^0(\perp) = \perp$. $F^0(\perp) \in \mathcal{Q}$ by (5.a). By recurrence, $\forall i \in \mathbb{N} \,.\, F^i(\perp) \in \mathcal{Q}$ by (5.b). F is continuous hence monotonically increasing so $\langle F^i(\bot) \in Q,\ i \in \mathbb{N}\rangle$ is a $\sqsubseteq$-increasing enumerable chain iterating F. If it is finite then $F^0(\bot) \sqsubset F^1(\bot) \sqsubset \ldots \sqsubset F^{n-1}(\bot) = F^n(\bot) = \ldots = \ldots$ for some $n \in \mathbb{N}$ proving that the chain is F-maximally increasing in Q. So, by (5.c), $\mathsf{lfp}^{\sqsubseteq} F = F^{n-1}(\bot) = \bigsqcup_{i\in\mathbb{N}} F^i(\bot) \in \mathcal{P}$. Otherwise, the chain is infinite and strictly increasing so F-maximally increasing in Q. By Th. 3 and (5.c), we conclude that $\mathsf{lfp}^{\sqsubseteq} F = \bigsqcup_{i\in\mathbb{N}} F^i(\bot) \in \mathcal{P}$.

Completeness ($\Rightarrow$): Let $F^{i+1}(\bot) = F(F^i(\bot))$ be the iterates of F from $F^0(\bot) = \bot$. Choosing $Q = \{F^i(\bot) \mid i \in \mathbb{N}\}$, we have (5.a) and (5.b). By Th. 3, $\langle F^i(\bot) \in Q,\ i \in \mathbb{N}\rangle$ is a $\sqsubseteq$-increasing chain in Q. It is enumerable and the only F-maximally increasing one so $\{x_i \in Q \mid i \in \mathbb{N}\} = \{F^i(\bot) \in Q \mid i \in \mathbb{N}\}$. By Th. 3, $\mathsf{lfp}^{\sqsubseteq} F = \bigsqcup_{i\in\mathbb{N}} F^i(\bot)$. By hypothesis, $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}$, and so $\bigsqcup_{i\in\mathbb{N}} F^i(\bot) = \bigsqcup_{i\in\mathbb{N}} x_i \in \mathcal{P}$, proving (5.c). □

Remark 2. The same way that the inductive invariant in fixpoint induction need not necessarily be the strongest possible one, Q need not necessarily be the strongest possible one $Q = \{F^i(\bot) \mid i \in \mathbb{N}\}$ in Th. 5, as used in the completeness proof. An example is $\mathcal{L} = \{\bot, a, b, c\}$ with $\bot \sqsubsetneq a \sqsubsetneq b \sqsubsetneq c$, $F(\bot) = F(a) = a$, $F(c) = F(b) = b$, and $\mathcal{P} = \{a, b\}$. Take $Q = \{a, b, c\}$ so that the only F-maximally $\sqsubseteq$-increasing chains in Q are $\bot a^\omega$, a^ω, and b^ω which lubs a and b belong to $\mathcal{P}$, proving $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}$. □

Remark 3. Following Rem. 1, Th. 5 generalizes to α-continuous functions on chain-α-complete posets and monotonically increasing functions on complete lattices, with Th. 5 holding for $\alpha = \omega$. □

Example 6 (Hoare logic). Let $\llbracket \mathtt{W} \rrbracket = \mathsf{lfp}^{\dot\sqsubseteq} F_{\mathtt{W}}$ be the denotational semantics of the iteration $\mathtt{W}$ = **`while (B) S`** where $F_{\mathtt{W}}(f)x = \llparenthesis \neg B(x) \mathrel{?} x \mathbin{:} f(S(x)) \rrparenthesis$ as defined in Ex. 1. Given $P, Q \in \wp(\mathcal{D})$, Hoare notation for partial correctness [14] is $\{\!|P|\!\}\ \mathtt{W}\ \{\!|Q|\!\}$ denoting $\forall x \in P\ .\ (\llbracket \mathtt{W} \rrbracket x \neq \bot) \Rightarrow (\llbracket \mathtt{W} \rrbracket x \in Q)$. Hoare partial correctness rule for the `while` iteration is

$$\frac{\{\!|I \cap B|\!\}\ \mathtt{S}\ \{\!|I|\!\}}{\{\!|I|\!\}\ \mathtt{W}\ \{\!|I \cap \neg B|\!\}} \tag{6}$$

[25, Sect. 6.6.6, p. 115] proves the soundness of the Hoare partial correctness rule for the `while` iteration based on its denotational semantics. The ad-hoc proof proceeds by induction on the semantics of the loop iterates and, assuming termination, passes to the limit. Formally, this consists in proving soundness by applying Th. 5, as follows.

– Take $Q \triangleq \{f \in \mathcal{D} \to \mathcal{D}_\bot \mid \forall x \in I\ .\ f(x) \neq \bot \Rightarrow f(x) \in I\}$.
– $\dot\bot \in Q$ by def. Q, proving (5.a).
– Assume that $f \in Q$. To prove (5.b), we must show that the premiss of Hoare rule (6) implies that $F_{\mathtt{W}}(f) \in Q$.

If $x \in I$ and $\neg B(x)$ then obviously $x \in I$. Otherwise if $x \in I \cap B(x)$ and $F_{\mathtt{W}}(f)x \neq \bot$ then $\{\!|I \cap B|\!\}\ \mathtt{S}\ \{\!|I|\!\}$ implies $S(x) \in I$ so if $S(x) \neq \bot$ then $f(S(x)) \in I$ since $f \in Q$ proving that $F_{\mathtt{W}}(f)x = f(S(x)) \in I$ that is $F_{\mathtt{W}}(f) \in Q$.

– Let $\langle f_i \in Q,\ i \in \mathbb{N}\rangle$ be any $F_{\mathtt{W}}$-maximally $\sqsubseteq$-increasing enumerable chain. Assume that $x \in I$ and $(\dot{\bigsqcup\limits_{i\in\mathbb{N}}} f_i)x \triangleq \bigsqcup_{i\in\mathbb{N}} f_i(x) \neq \bot$. By def. lub $\sqsubseteq$, $\exists j \in \mathbb{N}\ .\ \bigsqcup_{i\in\mathbb{N}} f_i(x) = f_j(x) \neq \bot$. Since $f_j \in Q$, $f_j(x) \in I$, proving $\bigsqcup_{i\in\mathbb{N}} f_i(x) \in I$ that is $\dot{\bigsqcup\limits_{i\in\mathbb{N}}} f_i \in Q$ which is (5.c).
– By Th. 5, we conclude that $\llbracket \mathtt{W} \rrbracket = \mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}} \in Q$ so $\forall x \in I\ .\ \llbracket \mathtt{W} \rrbracket(x) \neq \bot \Rightarrow \llbracket \mathtt{W} \rrbracket(x) \in I$. Moreover, if $(\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})x \neq \bot$ then $\neg B(\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})$, as shown in Ex. 1, proving $\{\!|I|\!\}\ \mathtt{W}\ \{\!|I \wedge \neg B|\!\}$.

Obviously, this rule is incomplete since I may not be inductive (so, for completeness, [5,6] has to ensure that I is inductive and use the consequence rule). □

9 Proving total correctness by generalized iteration induction

By completeness, the termination of $\mathsf{lfp}^{\dot\sqsubseteq} F$ on a termination domain $T \in \wp(\mathcal{D})$ can always be proved by generalized iteration induction Th. 5, if $\mathsf{lfp}^{\dot\sqsubseteq} F \in \mathcal{P}_T$ does hold.

Example 7 (Total correctness II). Continuing Ex. 3, let us define $\mathcal{P}_{\mathbb{N}} \triangleq \{f \in \mathbb{N} \rightarrow \mathbb{N}_\perp \mid \forall n \in \mathbb{N} \,.\, f(n) \neq \perp\}$ and apply Th. 5 to prove that prove that $\mathsf{lfp}^{\subseteq} F_! \in \mathcal{P}_{\mathbb{N}}$ (which, together with Ex. 3, shows that $\mathsf{lfp}^{\subseteq} F_! = f_!$).

Let us define $\forall i \in \mathbb{N} \,.\, Q_i \triangleq \{f \in \mathbb{N} \rightarrow \mathbb{N}_\perp \mid \forall n \in [0, i[\,.\, f(n) \neq \perp \wedge \forall n \geqslant i \,.\, f(n) = \perp\}$ and $Q \triangleq \bigcup_{i \in \mathbb{N}} Q_i$.

- We have $\perp \in \{\perp\} = Q_0 \subseteq Q$, proving (5.a).
- Assume that $i \in \mathbb{N}$ and $f \in Q_i$. We have

$$
\begin{aligned}
& F_!(f) \\
= \;& \lambda n \bullet \llbracket n = 0 \mathrel{?} 1 \mathrel{⁏} n \times f(n-1) \rrbracket && \wr\text{def. } F_! \text{ in Ex. } 3 \wr \\
\Rightarrow \;& F_!(f)0 \neq \perp \wedge \forall n-1 \in [0, i[\,.\, F_!(f)(n) \neq \perp && \wr f \in Q_i \wr \\
\Rightarrow \;& F_!(f) \in Q_{i+1} && \wr\text{def. } Q_{i+1} \wr
\end{aligned}
$$

- It follows, by def. of Q, that if $f \in Q$ then $f \in Q_i$ for some $i \in \mathbb{N}$ and therefore $F_!(f) \in Q_{i+1} \subseteq Q$, so that (5.b) holds.
- Let $\langle f_n,\ n \in \mathbb{N}\rangle$ be $F_!$-maximally increasing chain of elements of Q. So, by def. Q, we have $f_0 \in Q_{j_0}$, $f_1 \in Q_{j_1}$, ..., $f_n \in Q_{j_n}$, $f_{n+1} \in Q_{j_{n+1}}$, ...

Assume that the chain is stationary at some rank i such that $f_0 \subsetneq f_{i-1} \subsetneq \ldots \subsetneq f_i = f_{i+1} = \ldots$. Then $f_i \in Q_j$ for some $j \in \mathbb{N}$. So $f_{i+1} = f_i \in Q_j$ and $f_{i+1} = F_!(f_i) \in Q_{j+1}$, a contradiction since $Q_j \cap Q_{j+1} = \varnothing$.[1]

It follows that the chain $f_0 \subsetneq f_1 \subsetneq \ldots \subsetneq f_n \subsetneq \ldots$ is strictly increasing and we have $j_0 < j_1 < \ldots < j_n < j_{n+1} < \ldots$ so $j_{n+1} > n + 1$. Since $f_{n+1} \in Q_{j_{n+1}}$, $f_{n+1}(n) \neq \perp$.

To prove that $\dot{\bigsqcup}_{i \in \mathbb{N}} f_i \in \mathcal{P}_{\mathbb{N}}$, assume by contradiction, that $\exists n \in \mathbb{N} \,.\, (\bigsqcup_{i \in \mathbb{N}} f_i)n = \perp$ so, by def. $\dot\sqcup$, $\exists n \in \mathbb{N} \,.\, \forall i \in \mathbb{N} \,.\, f_i(n) = \perp$. In particular $f_{n+1}(n) = \perp$, a contradiction.

We have proved (5.c) hence $\mathsf{lfp}^{\subseteq} F_! \in \mathcal{P}_{\mathbb{N}}$, that is $\forall n \in \mathbb{N} \,.\, (\mathsf{lfp}^{\subseteq} F_!)n \neq \perp$. □

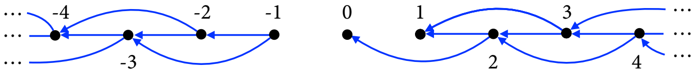
A much simpler way of proving termination of $\mathsf{lfp}^{\subseteq} F_!$ for positive parameters is to observe that parameters strictly decreases on recursive call and remains positive which can be done only a finite number of times since $\langle \mathbb{N}, <\rangle$ is well-founded. Such termination proofs using a variant/convergence function are formalized in Th. 10. Th. 11 shows that this proof is equivalent to the above proof based on Th. 5.

10 Parameter dependency

The fact that the evaluation of $f(x) = F(f)x$ for parameter $x \in \mathcal{D}$ where $f = \mathsf{lfp}^{\subseteq} F$ makes a recursive call to $f(y)$ with parameter $y \in \mathcal{D}$, written $x \xmapsto{F} y$, is usually defined syntactically.

[1] Notice that with our choice of Q, this is not necessarily true for chains that are not F_{W}-iterations.

Example 8. Define $f(n) = F(f)n \triangleq \llbracket n \in [0,1] \mathrel{?} 0 \mathrel{:} f(n-1) + f(n-2) \rrbracket$. A call of f for $n \notin [0,1]$ will recursively call $f(n-1)$ and $f(n-2)$ in the expression $f(n-1) + f(n-2)$. So $\overset{F}{\longmapsto} = \{\langle n,\ n-1\rangle, \langle n,\ n-2\rangle \mid n \in \mathbb{Z} \setminus [0,1]\}$:



□

Since we don't want to provide a specific syntax for defining F, we have to define the call relation $\overset{F}{\longmapsto}$ semantically. We let $f[y \leftarrow d] \in \mathcal{D} \rightarrow \mathcal{D}_\perp$ be the function f except for paramater y for which it has value $d \in \mathcal{D}_\perp$.

$$\begin{aligned} f[y \leftarrow d](y) &\triangleq d \\ f[y \leftarrow d](z) &\triangleq f(z) \quad \text{when} \quad z \neq y \end{aligned}$$

The call relation is semantically defined as follows.

$$\begin{aligned} x \overset{F}{\longmapsto} y \triangleq{} & \mathsf{let}\ f = \mathsf{lfp}^{\sqsubseteq} F\ \mathsf{and}\ f'(z) = \llbracket f(z) = \perp \mathrel{?} 0 \mathrel{:} f(z) \rrbracket\ \mathsf{in} \\ & F(f'[y \leftarrow \perp])x = \perp \wedge F(f')x \neq \perp \end{aligned} \tag{7}$$

For simplicity, we assume F to be always well-defined so choosing $f'(z) = 0$ can never lead to a runtime error. The idea is that forcing f to terminate for all its parameters but for y for which f does not terminate, the main call to x will not terminate so this can only come from a recursive call to $f(y)$ (or the body of F does not terminate independently of its recursive calls to f, which we exclude by $F(f')x \neq \perp$).

Example 9. Continuing Ex. 8, let $f(n) = F(f)n \triangleq \llbracket n \in [0,1] \mathrel{?} 0 \mathrel{:} f(n-1) + f(n-2) \rrbracket$. The semantics is $f = \mathsf{lfp}^{\sqsubseteq} F = \lambda n \bullet \llbracket n \geqslant 0 \mathrel{?} 0 \mathrel{:} \perp \rrbracket$ and $f' = \lambda n \bullet 0$. We have

$$\begin{aligned} & F(f'[n-1 \leftarrow \perp])n \\ ={} & \llbracket n \in [0,1] \mathrel{?} 0 \mathrel{:} f'[n-1 \leftarrow \perp](n-1) + f'[n-1 \leftarrow \perp](n-2) \rrbracket && \text{(def. } F\text{)} \\ ={} & \llbracket n \in [0,1] \mathrel{?} 0 \mathrel{:} \perp + 0 \rrbracket && \text{(def. } f'[n-1 \leftarrow \perp]\text{)} \\ ={} & \llbracket n \in [0,1] \mathrel{?} 0 \mathrel{:} \perp \rrbracket && \text{(def. + assumed to be strict)} \end{aligned}$$

Similarly $F(f'[n-2 \leftarrow \perp])n = \llbracket n \in [0,1] \mathrel{?} 0 \mathrel{:} \perp \rrbracket$. In conclusion, $\overset{F}{\longmapsto} = \{\langle n,\ n-1\rangle, \langle n,\ n-2\rangle \mid n \in \mathbb{Z} \setminus [0,1]\}$. □

11 Recursive non-termination

Since we are interested in the termination of recursive functions $f(x) = F(f)x$, we exclude non-termination of the function due to causes other than recursive calls in F:

Definition 1 (function body termination hypothesis).

$$\forall f \in \mathcal{D} \rightarrow \mathcal{D}_\perp \ .\ \forall x \in \mathcal{D}\ .\ (F(f)x = \perp) \Rightarrow (\exists y \in \mathcal{D}\ .\ x \overset{F}{\longmapsto} y \wedge f(y) = \perp) \tag{8}$$

Example 10 (function body non-termination). Define $F(f)x = \texttt{if (}x = 0\texttt{)}\ 1\ \texttt{else while (}\mathtt{tt}\texttt{) ;}f(0)$, we have $F(f)1 = \bot$ since the iteration is entered and never exited so the function body termination hypothesis (8) is not satisfied. This is because the non-termination is not due to the recursive calls but only to the loop body.

For $F(f)x = f(f(x))$, if $\forall x \in \mathcal{D} \,.\, f(x) \neq \bot$ is assumed to always terminate then $F(f)x = f(f(x)) \neq \bot$ does terminate, so satisfies the function body termination hypothesis (8). □

A recursive function definition satisfying the function body termination hypothesis (8) does not terminate for a given parameter if and only if it makes a recursive call that does not terminate.

Lemma 9 *Let $f = \mathsf{lfp}^{\sqsubseteq} F$ where F is continuous and satisfies the function body termination hypothesis* (8). *Then $f(x) = \bot$ if and only if $\exists y \in \mathcal{D} \,.\, x \overset{F}{\longmapsto} y \wedge f(y) = \bot$.* □

Proof. – Let F satisfying the function body termination hypothesis (8) and $f = \mathsf{lfp}^{\sqsubseteq} F$. We have $f(x) = \bot$ if and only if $F(f)x = \bot$ which, by (8), implies $\exists y \in \mathcal{D} \,.\, x \overset{F}{\longmapsto} y \wedge f(y) = \bot$.
– Conversely, let $f'(z) = \llbracket f(z) = \bot \mathrel{?} 0 \mathrel{⁏} f(z) \rrbracket$. Assume that $\exists y \in \mathcal{D} \,.\, x \overset{F}{\longmapsto} y \wedge f(y) = \bot$. By (7), $x \overset{F}{\longmapsto} y$ implies $F(f'[y \leftarrow \bot])x = \bot$. Since $\bot \sqsubseteq 0$ and $f(y) = \bot$, $f \mathrel{\dot{\sqsubseteq}} f'[y \leftarrow \bot]$ pointwise. Moreover, F is continuous hence monotonically increasing so $f(x) = F(f)x \sqsubseteq F(f'[y \leftarrow \bot]) = \bot$ so $f(x) = \bot$ since $\bot$ is the infimum. □

The function body termination hypothesis (8) is not restrictive. It simply means that, assuming that all recursive calls to f do terminate, the function body $F(f)$ must be proved to terminate. Depending on the considered programming language, this can be done *e.g.* by structural induction, using variant/convergence functions (as in Th. 10), etc. This may involve a preliminary partial correctness proof (*e.g.* using Th. 2) to restrict the values that can be taken by variables.

12 Proving termination by a variant/convergence function

Following Turing [29] and Floyd [12], most termination proofs are done using a variant/convergence function in a well-founded set which strictly decreases at each recursive call (or, equivalently, a well-founded relation). This is the case *e.g.* of the "size change principle" [13]. The variant/convergence function termination proof principle can be formulated as follows.

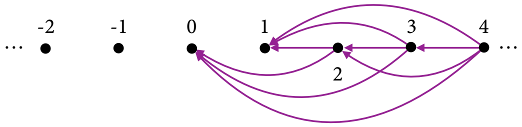
A relation $\langle D, \preccurlyeq\rangle$ such that $\preccurlyeq \in \wp(D \times D)$ is well-founded or Noetherian if and only if there is no infinite strictly $\gtrdot$-decreasing chain of elements of D.

Theorem 10 (variant/convergence function proof principle for termination) *Let $F \in (\mathcal{D} \rightarrow \mathcal{D}_{\bot}) \xrightarrow{uc} (\mathcal{D} \rightarrow \mathcal{D}_{\bot})$ be a continuous function on the cpo $\langle \mathcal{D} \rightarrow \mathcal{D}_{\bot}, \dot{\sqsubseteq}, \dot{\bot}, \dot{\sqcup}\rangle$ satisfying the function body termination hypothesis* (8)*, $T \in \wp(\mathcal{D})$, and $\mathcal{P}_T \triangleq \{f \in \mathcal{D} \rightarrow \mathcal{D}_{\bot} \mid \forall x \in T \,.\, f(x) \neq \bot\}$. Then*

$$
\begin{aligned}
\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}_T \Leftrightarrow \quad & \exists D \in \wp(\mathcal{D}) \,.\, T \subseteq D && \text{(10.a)}\\
\wedge \quad & \exists \preccurlyeq \in \wp(D \times D) \,.\, \langle D, \preccurlyeq\rangle \textit{ is well-founded} && \text{(10.b)}\\
\wedge \quad & \forall x \in D \,.\, \forall y \in \mathcal{D} \,.\, (x \overset{F}{\longmapsto} y) \Rightarrow (y \in D \wedge x \gtrdot y) && \text{(10.c)}
\end{aligned}
$$
□

Intuitively $\mathsf{lfp}^{\sqsubseteq} F$ must terminate since, by contradiction, an infinite call sequence would create an infinite descent along the called parameters. Completeness follows from the fact that $\gtrdot$ can be chosen as $\overset{F}{\longmapsto}$, which is always well-founded for terminating programs. This is proved formally in Cor. 12.

Example 11. Continuing Ex. 8, the well-founded relation $<$ can be chosen as follows



Termination follows from the fact that $\xrightarrow{F}$ restricted to the naturals in included in $\gtrdot$, which is well-founded. □

The variant/convergence function proof principle remains sound when this semantic dependency relation is over-approximated syntactically (but maybe not complete, as shown by $F(f)x \triangleq \llbracket \mathsf{tt} \mathbin{?} x \mathbin{⁏} f(x) \rrbracket$ where $x \xmapsto{F} x$ syntactically, but not semantically because the recursive call $f(x)$ is not reachable).

13 Equivalence of the termination proof by generalized iteration induction and by variant/convergence function principle

Theorem 11 *Let $\mathcal{L} = \mathcal{D} \rightarrow \mathcal{D}_\perp$ and $F \in \mathcal{L} \xrightarrow{uc} \mathcal{L}$ satisfying the function body termination hypothesis (8). There exists a termination proof by the generalized iteration induction of Th. 5 for F if and only if there exists one by the variant/convergence function principle of Th. 10.* □

The proof is constructive in that it shows how to construct a proof by one method knowing a proof by the other method.

Proof. — Let us first show that the existence of a termination proof of $\mathsf{lfp}^{\sqsubseteq} F$ by Th. 5 implies the existence of a termination proof of $\mathsf{lfp}^{\sqsubseteq} F$ by Th. 10.

– If $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}_T$ has been proved by Th. 5, then, as shown by the completeness proof of this theorem, this can also be done by Th. 5 with $\forall i \in \mathbb{N} \mathrel{.} Q^i \triangleq \{F^i\}$ where $F^0 = \lambda x \bullet \perp$ and $F^{i+1} = F(F^i)$ are the iterates of F from $\lambda x \bullet \perp$ and $Q \triangleq \bigcup_{i \in \mathbb{N}} Q^i$ so that (5.a) and (5.b) are satisfied. The only F-maximal $\sqsubseteq$-increasing enumerable chain $\langle x_i \in Q, i \in \mathbb{N}\rangle$ is $\langle F^i, i \in \mathbb{N}\rangle$. By Th. 3, it is such that $\dot{\bigsqcup}_{i \in \mathbb{N}} F^i = \mathsf{lfp}^{\sqsubseteq} F$, By hypothesis $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}_T$ and so $\dot{\bigsqcup}_{n \in \mathbb{N}} F^i \in \mathcal{P}_T$, proving (5.c).

– Let us define $D^0 \triangleq \varnothing$, $D^i \triangleq \{x \mid F^{i-1}(x) = \perp \wedge F^i(x) \neq \perp\}$ for all $i > 0$, and $D \triangleq \bigcup_{i \in \mathbb{N}} D^i$. Let us prove that $T \subseteq D$.

By def. $\dot{\sqcup}$ and $\mathsf{lfp}^{\sqsubseteq} F = \dot{\bigsqcup}_{i \in \mathbb{N}} F^i$, for all $n \in \mathcal{D}$, we have $(\mathsf{lfp}^{\sqsubseteq} F)n \neq \perp \Leftrightarrow \exists i \in \mathbb{N} \mathrel{.} F^i(n) \neq \perp \Leftrightarrow \exists i \in \mathbb{N} \mathrel{.} n \in D^i \Leftrightarrow n \in D$. Since $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}_T$, for all $n \in T$, we have $(\mathsf{lfp}^{\sqsubseteq} F)n \neq \perp \Leftrightarrow n \in D$, proving $T \subseteq D$, that is (10.a).

– Let us define $x > y$ if and only if $\exists i \in \mathbb{N} \mathrel{.} x \in D^{i+1} \wedge y \in D^i$. Let $\gtrdot$ be the irreflexive transitive closure of $>$. Let us prove that $\langle D, \leqslant\rangle$ is well-founded. By def. of $\leqslant$, for an infinite strictly decreasing chain for $\leqslant$ there exists one $x_0 > x_1 > x_2 \ldots$ for $<$, and so there would exists $x_0 \in D^{i_0}$, $x_1 \in D^{i_1}$, $x_2 \in D^{i_2}$, ... with $i_0 > i_1 > i_2 > \ldots$ which is impossible since this chain on $\mathbb{N}$ cannot be infinite decreasing. This implies (10.b).

– Let x be any $x \in D$ and $y \in \mathcal{D}$ satisfies $x \xmapsto{F} y$.

Since $x \in D$, there exists $i \in \mathbb{N}$ such that $x \in D^i$. Let i be the minimal such i. Since $x \xmapsto{F} y$, we have $\mathsf{lfp}^{\dot{\sqsubseteq}} F(x) \neq F(\mathsf{lfp}^{\dot{\sqsubseteq}} F[y \leftarrow \perp])x$. Therefore $(\mathsf{lfp}^{\sqsubseteq} F)y \neq \perp$ since otherwise $F(\mathsf{lfp}^{\dot{\sqsubseteq}} F[y \leftarrow \perp]) =$

$F(\mathsf{lfp}^{\dot\sqsubseteq} F) = \mathsf{lfp}^{\dot\sqsubseteq} F$, in contradiction with $F(\mathsf{lfp}^{\dot\sqsubseteq} F[y \leftarrow \bot])x \neq \mathsf{lfp}^{\dot\sqsubseteq} F(x)$. It follows that $\exists j \in \mathbb{N} \mathrel{.} y \in D^j \subseteq D$.

- If $j < i$ then there exist $z_0 = y \in D^j$, $z_1 \in D^{j+1}$, ..., $z_{i-j} = x \in D^i$ such that, by def. of D^k, $\forall k \in [0, i-j] \mathrel{.} F^{k-1}(z_k) = \bot \wedge F^k(z_k) \neq \bot$. By def. $<$, we have $y = z_0 < z_1 < \ldots < z_{i-j} = x$ proving that $x \gtrdot y \in D$ *i.e.* (10.c).
- Else $j \geqslant i$. By def. $\dot\sqsubseteq$ $F_{i-1}(x) = \bot$, $F_i(x) = F_j(x) = (\mathsf{lfp}^{\dot\sqsubseteq} F)x \neq \bot$, $F_{i-1}(y) = F_i(y) = F_{j-1}(y) = \bot$, and $F_j(y) = (\mathsf{lfp}^{\dot\sqsubseteq} F)y \neq \bot$. Since $F_j(x) = F(F_{j-1}(x))$ and $F_{j-1}(y) = \bot$, we have $F_j(x) = F(F_{j-1}[y \leftarrow \bot](x))$ so $(\mathsf{lfp}^{\dot\sqsubseteq} F)(x) = F((\mathsf{lfp}^{\dot\sqsubseteq} F)[y \leftarrow \bot](x))$, a contradiction. This case is impossible and so (10.c) holds vacuously.

– Conversely, let us first show that the existence of a termination proof of $f = \mathsf{lfp}^{\sqsubseteq} F$ by Th. 10 implies the existence of a termination proof of $f = \mathsf{lfp}^{\sqsubseteq} F$ by Th. 5. So assume the existence of $D \in \wp(\mathcal{D})$ satisfying (10.a), (10.b), and (10.c).

– Define $Q_i = \{F^i(\bot)\}$ and $Q \triangleq \bigcup_{i \in \mathbb{N}} Q_i$ so (5.a) and (5.b) do hold. It remains to prove (5.c) that is $\dot{\bigsqcup_{i \in \mathbb{N}}} F^i(\bot) \in \mathcal{P}_T$. By reductio ad absurdum, assume that $\exists x_0 \in T \mathrel{.} (\dot{\bigsqcup_{i \in \mathbb{N}}} F^i(\bot))x_0 = \bot$, that is, by (10.a) and Th. 3, $x_0 \in D$ and $(\mathsf{lfp}^{\dot\sqsubseteq} F)x_0 = \bot$. Assume $\exists x_j \in D \mathrel{.} f(x_j) = \bot$ where $f = \mathsf{lfp}^{\dot\sqsubseteq} F$. Then, by the function body termination hypothesis (8), Lem. 9 implies that $\exists x_{j+1} \mathrel{.} x_j \overset{F}{\longmapsto} x_{j+1} \wedge f(x_{j+1}) = \bot$. In this way, we can built an infinite sequence $x_0 \overset{F}{\longmapsto} x_1 \overset{F}{\longmapsto} x_2 \overset{F}{\longmapsto} \ldots$ such that $\forall j \in \mathbb{N} \mathrel{.} (\mathsf{lfp}^{\dot\sqsubseteq} F)x_j = \bot$. By recurrence and (10.c), this sequence is in D and $\gtrdot$-decreasing. This is in contradiction with the well-foundness (10.b) of $\langle D, \lessdot\rangle$. □

Corollary 12 *The variant/convergence function principle* (10) *is sound and complete for proving termination.* □

Proof. By Th. 11 and Th. 5. □

14 Extension to total correctness

The proof by [4] that Hoare logic does not exists for functional languages is based on the restriction of predicates to first-order logic with program variables only. But this is no longer the case without this restriction [11,24] and can be extended to total correctness.

Theorem 13 (The total correctness proof principle) *Let $F \in \mathcal{D} \xrightarrow{uc} \mathcal{D}_\bot$ satisfying the function body termination hypothesis (8) be a continuous function on the cpo $\langle \mathcal{D}_\bot, \sqsubseteq, \bot, \sqcup \rangle$ where $\bot \notin \mathcal{D}$, $\mathcal{D}_\bot = \mathcal{D} \cup \{\bot\}$, $\forall x \in \mathcal{D} \mathrel{.} \bot \sqsubseteq \bot \sqsubsetneq x \sqsubseteq x$, $P \in \wp(\mathcal{D})$, $Q \in \wp(\mathcal{D} \times \mathcal{D})$, and $\mathcal{P}_{P,Q} \triangleq \{f \in \mathcal{D} \rightarrow \mathcal{D}_\bot \mid \forall x \in P \mathrel{.} \langle x,\ f(x) \rangle \in Q\}$. Then*

$$\begin{aligned}
\mathsf{lfp}^{\dot\sqsubseteq} F \in \mathcal{P}_{P,Q} \Leftrightarrow \exists D \in \wp(\mathcal{D}) \mathrel{.} \exists I \in \wp(\mathcal{D} \times \mathcal{D}) \mathrel{.} \qquad & \\
& P \subseteq D && (13.a)\\
\wedge\ & \{\langle x,\ y \rangle \in I \mid x \in P\} \subseteq Q && (13.b)\\
\wedge\ & \exists \preccurlyeq \in \wp(\mathcal{D} \times \mathcal{D}) \mathrel{.} \langle D, \preccurlyeq \rangle \textit{ is well-founded} && (13.c)\\
\wedge\ & \forall x, y \in \mathcal{D} \mathrel{.} (x \in D \wedge x \xmapsto{F} y) \Rightarrow (y \in D \wedge x \succ y) && (13.d)\\
\wedge\ & \mathsf{let}\ \mathcal{P}_{D,I} \triangleq \{f \in \mathcal{D} \rightarrow \mathcal{D}_\bot \mid \forall x \in D \mathrel{.} (f(x) \neq \bot \Rightarrow \langle x,\ f(x) \rangle \in I)\}\ \mathsf{in} && (13.e)\\
& \qquad \forall f \in \mathcal{P}_{D,I} \mathrel{.} F(f) \in \mathcal{P}_{D,I} && \square
\end{aligned}$$

Example 12 (Total correctness of the factorial). Define $F_!(f) \triangleq \lambda n \bullet [\![n = 0 \mathrel{?} 1 \mathrel{:} n \times f(n-1)]\!]$, $P = \mathbb{N}$, $Q = \{\langle n,\ n! \rangle \mid n \in \mathbb{N}\}$ So that $\mathsf{lfp}^{\dot\sqsubseteq} F \in \mathcal{P}_{P,Q}$ expresses that $F_!(f)n$ terminates for $n \in \mathbb{N}$ and returns the factorial $n!$ of n. Take $D = P$ and $I = Q$ so that (13.a), (13.b), (13.c) are trivially satisfied since $D = \mathbb{N}$ and $\langle \mathbb{N}, \leqslant \rangle$ is well-founded. If $n \in D$ and $n \xmapsto{F} y$ then $n \neq 0$ and $y = n - 1$ so $n > n - 1 \in D$, proving (13.d). If $f \in \mathcal{P}_{D,I}$ and $n \in D = \mathbb{N}$ then $f(n) = n!$. So $F_!(f)n = n!$ since either $n = 0$ and $F_!(f)0 = 1 = 0!$ or $n > 0$ so $n - 1 \in \mathbb{N}$, $f(n-1) = (n-1)!$ so $F_!(f)n = n \times (n-1)! = n!$. Therefore $F_!(f) \in \mathcal{P}_{D,I}$, proving (13.e). By Th. 13, $\mathsf{lfp}^{\dot\sqsubseteq} F_! \in \mathcal{P}_{P,Q}$. □

Proof (of Th. 13). Soundness ($\Leftarrow$): Take $T = D$ in Th. 10. Then (13.a) implies (10.a), (13.c) implies (10.b), and (13.d) implies (10.c). By Th. 10, this implies $\forall x \in P \mathrel{.} (\mathsf{lfp}^{\sqsubseteq} F)x \neq \bot$.

Ta ke $\mathcal{L} = \mathcal{D} \rightarrow \mathcal{D}_\bot$, $\mathcal{P} = \mathcal{Q} = \mathcal{P}_{D,I}$. By def. $\mathcal{P}_{D,I}$, $\dot\bot \in \mathcal{Q}$ proving (5.a). By (13.d), $\forall f \in \mathcal{Q} \mathrel{.} F(f) \in \mathcal{Q}$, proving (5.b). Let $\{f_i \in \mathcal{Q} \mid i \in \mathbb{N}\}$ be any F-maximal $\sqsubseteq$-increasing chain of elements of $\mathcal{Q}$. If $x \in \mathcal{D}$ and $(\dot\bigsqcup_{i \in \mathbb{N}} f_i)x \neq \bot$, then $(\dot\bigsqcup_{i \in \mathbb{N}} f_i)x = d \in \mathcal{D}$ so, by def. lub $\dot\bigsqcup$, there exists $j \in \mathbb{N} \mathrel{.} (\dot\bigsqcup_{i \in \mathbb{N}} f_i)x = f_j(x) = d$. But $f_j \in \mathcal{Q} = \mathcal{P}_{D,I}$ so $\langle x,\ d \rangle = \langle x,\ (\dot\bigsqcup_{i \in \mathbb{N}} f_i)x \rangle \in I$. If follows that $(\dot\bigsqcup_{i \in \mathbb{N}} f_i) \in \mathcal{P}_{D,I} = \mathcal{P}$. By Th. 5, $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}_{D,I}$.

Since $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}_{D,I}$ and $\forall x \in P \mathrel{.} (\mathsf{lfp}^{\sqsubseteq} F)x \neq \bot$, we conclude that $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}_{P,Q}$.

Completeness ($\Rightarrow$): Assume that $\mathsf{lfp}^{\sqsubseteq} F \in \mathcal{P}_{P,Q}$ so $\forall x \in P \mathrel{.} (\mathsf{lfp}^{\sqsubseteq} F)x \neq \bot$. Applying Th. 10 with $T = P$, there exists $D \in \wp(\mathcal{D})$ satisfying (10.a), (10.b), and (10.c). Applying Th. 5 with $\mathcal{L} = \mathcal{D} \rightarrow \mathcal{D}_\bot$, there exists $\mathcal{Q} \in \wp(\mathcal{D} \rightarrow \mathcal{D}_\bot)$ satisfying (5.a), (5.b), (5.c). Moreover, the completeness proof of Th. 5 shows that one can choose $\mathcal{Q} = \{F^i(\bot) \mid i \in \mathbb{N}\}$.

Choose $I \triangleq \{\langle x,\ f(x) \rangle \in \mathcal{D} \times \mathcal{D} \mid f \in \mathcal{Q} \vee f = \bigsqcup \mathcal{Q}\}$, where, as shown in the completeness proof of Th. 5, $\bigsqcup \mathcal{Q}$ is well-defined and equal to $\mathsf{lfp}^{\dot\sqsubseteq} F$.

- We have $P = T \subseteq D$ by (10.a), proving (13.a);
- If $\langle x, y \rangle \in I$ then $y \neq \bot$ and $y = f(x)$ where $f \in \mathcal{Q}$ or $f = \bigsqcup \mathcal{Q}$. In both cases, by $\mathcal{Q} = \{F^i(\bot) \mid i \in \mathbb{N}\}$ and $f(x) \neq \bot$, we have $y = \mathsf{lfp}^{\sqsubseteq} F)x$ so, by hypothesis $\mathsf{lfp}^{\dot\sqsubseteq} F \in \mathcal{P}_{P,Q}$, if $x \in P$, then $\langle x,\ y \rangle \in Q$, proving (13.b);
- (10.b) is exactly (13.c);
- (10.c) is exactly (13.d);

– Assume that $f \in \mathcal{P}_{D,I} = \{f \in \mathcal{D} \rightarrow \mathcal{D}_\perp \mid \forall x \in D \,.\, (f(x) \neq \perp \Rightarrow \langle x,\ f(x)\rangle \in I)\}$. Then either $f \in \mathcal{Q}$ so, by (5.b), $F(f) \in \mathcal{Q}$ and therefore $F(f) \in \mathcal{P}_{D,I}$ or $f = \bigsqcup \mathcal{Q} = \mathsf{lfp}^{\sqsubseteq} F$ so $F(f) = f \in \mathcal{P}_{D,I}$, proving (13.e). □

15 Application to the `while` iteration

Manna and Pnueli [20] generalized Hoare partial correctness rule for total correctness $\llparenthesis P \rrparenthesis\ \mathtt{W}\ \llparenthesis Q \rrparenthesis$ denoting $\forall x \in P \,.\, \llbracket \mathtt{W} \rrbracket x \in Q$ which is traditionally decomposed in partial correctness $\{\!|P|\!\}\ \mathtt{W}\ \{\!|Q|\!\}$ and termination $\forall x \in P \,.\, \llbracket \mathtt{W} \rrbracket x \neq \perp$. They rely on the idea of relating the initial and final values of variables in Q, writing $P(x)$ for $x \in P \in \wp(\mathcal{D})$ and $Q(x, x')$ for $\langle x,\ x'\rangle \in Q \in \wp(\mathcal{D} \times \mathcal{D})$, so that the rule are written in the form $\llparenthesis P(x) \rrparenthesis\ \mathtt{W}\ \llparenthesis Q(x, x') \rrparenthesis$ where x is the value before execution and x' that upon termination.

$\llparenthesis P(x) \rrparenthesis\ \mathtt{W}\ \llparenthesis Q(x, x') \rrparenthesis$ is equivalent to $\mathsf{lfp}^{\dot\sqsubseteq} F_{\mathtt{W}} \in \mathcal{P}_{P,Q}$. So, by the soundness and completeness of Th. 13, this is equivalent to the existence of $D \in \wp(\mathcal{D})$ and $I \in \wp(\mathcal{D} \times \mathcal{D})$ satisfying the conditions.

$$
\begin{array}{ll}
& P(x) \Rightarrow D(x) \qquad (14.\mathrm{a})\\
\wedge & P(x) \wedge I(x, y) \Rightarrow Q(x, y) \qquad (14.\mathrm{b})\\
\wedge & \exists \preccurlyeq \in \wp(\mathcal{D} \times \mathcal{D}) \,.\, \langle D,\ \preccurlyeq\rangle \text{ is well-founded} \qquad (14.\mathrm{c})\\
\wedge & \forall x \in D \,.\, S(x) \in D \wedge x \succ S(x) \qquad (14.\mathrm{d})\\
\wedge & \forall x \in D, x'' \in \mathcal{D} \,.\, (B(x) \wedge I(S(x), x'') \Rightarrow I(x, x'') \qquad (14.\mathrm{e})\\
\wedge & \forall x \in D \,.\, \neg B(x) \Rightarrow I(x, x) \qquad (14.\mathrm{e}')
\end{array}
$$

since for (13.d), $x \xmapsto{F} y$ if and only if $y = S(x)$, by def. $F_{\mathtt{W}}(f)x = \llbracket \neg B(x) \mathrel{?} x \mathbin{⨾} f(S(x)) \rrbracket$ and for (13.e/e′), given $\mathcal{P}_{D,I} \triangleq \{f \in \mathcal{D} \rightarrow \mathcal{D}_\perp \mid \forall x \in D \,.\, f(x) \neq \perp \Rightarrow \langle x,\ f(x)\rangle \in I\}$, we have

$$
\begin{array}{ll}
& \forall f \in \mathcal{P}_{D,I} \,.\, F_{\mathtt{W}}(f) \in \mathcal{P}_{D,I}\\
\Leftrightarrow & \forall f \in \mathcal{D} \rightarrow \mathcal{D}_\perp \,.\, (\forall x \in D \,.\, (f(x) \neq \perp) \Rightarrow (\langle x,\ f(x)\rangle \in I)) \Rightarrow (\forall x \in D \,.\, \llbracket \neg B(x) \mathrel{?} \langle x,\\
& x\rangle \in I \mathbin{⨾} (f(S(x)) \neq \perp) \Rightarrow (\langle x,\ f(S(x))\rangle \in I) \rrbracket) \qquad \wr\text{def. } \mathcal{P}_{D,I} \text{ and } F_{\mathtt{W}}\int\\
\Leftrightarrow & \forall f \in \mathcal{D} \rightarrow \mathcal{D} \,.\, (\forall x \in D \,.\, (\langle x,\ f(x)\rangle \in I)) \Rightarrow (\forall x \in D \,.\, \llbracket \neg B(x) \mathrel{?} \langle x,\ x\rangle \in I \mathbin{⨾} (\langle x,\ f(S(x))\rangle \in I) \rrbracket)\\
& \wr\text{since the } \perp \text{ case is excluded}\int\\
\Leftrightarrow & (\forall x \in D \,.\, \neg B(x) \Rightarrow \langle x,\ x\rangle \in I) \wedge (\forall x \in D \,.\, (B(x) \wedge \langle S(x),\ x''\rangle \in I) \Rightarrow \langle x,\ x'\rangle \in I)\\
& \qquad \wr\text{since } f \text{ is defined by } I \text{ and letting } x' = f(S(x))\int \quad \square
\end{array}
$$

Rewriting (14) in Manna-Pnueli style, we get the sound and complete rule (which incorporate the consequence rule):

$$
\begin{array}{cr}
P(x) \Rightarrow D(x), \quad P(x) \wedge I(x, y) \Rightarrow Q(x, y), & (15.\mathrm{a/b})\\
\exists \preccurlyeq \in \wp(\mathcal{D} \times \mathcal{D}) \,.\, \langle D,\ \preccurlyeq\rangle \text{ is well-founded}, & (15.\mathrm{c})\\
\llparenthesis D(x) \rrparenthesis\ \mathtt{S}\ \llparenthesis D(x') \wedge x \succ x' \rrparenthesis, & (15.\mathrm{d})\\
\llparenthesis D(x) \wedge B(x) \rrparenthesis\ \mathtt{S}\ \llparenthesis I(x, x') \wedge \forall x'' \,.\, I(x', x'') \Rightarrow I(x, x'') \rrparenthesis, & (15.\mathrm{e})\\
\forall x \,.\, D(x) \wedge \neg B(x) \Rightarrow I(x, x) & (15.\mathrm{e}')\\
\hline
\llparenthesis P(x) \rrparenthesis\ \mathtt{W}\ \llparenthesis Q(x, x') \wedge \neg B(x') \rrparenthesis &
\end{array}
$$

(The conjunction with the post-condition $\neg B(x')$ is explained in Ex. 6).

The original Manna and Pnueli rule [20, Sect. 8.3] is slightly different, as follows.

$$\begin{array}{c}\llparenthesis\, P(x) \wedge B(x) \,\rrparenthesis\; \mathtt{S}\; \llparenthesis\, P(x') \wedge Q(x,x') \wedge x \gtrdot x' \,\rrparenthesis, \qquad (16.\mathrm{i})\\ \forall x, x', x'' \,.\, Q(x,x') \wedge Q(x',x'') \Rightarrow Q(x,x''), \qquad (16.\mathrm{ii})\\ \forall x \,.\, P(x) \wedge \neg B(x) \Rightarrow Q(x,x) \qquad (16.\mathrm{iii})\\ \hline \llparenthesis\, P(x) \,\rrparenthesis\; \mathtt{W}\; \llparenthesis\, Q(x,x') \wedge \neg B(x') \,\rrparenthesis\end{array}$$

As in [14], the proof rules are postulated so no soundness or completeness proof is given. A soundness and completeness proof is provided in [1] based on Scott induction (using transfinite iterates in absence of continuity due to the consideration of unbounded nondeterminism).

Assume the hypotheses of Manna-Pnueli inference rule (16), and define (with informal notations)

- $P' = D \triangleq P(x)$;
- $Q' = I(x,y) \triangleq Q(x,y)$;

so that

- (14.a) holds trivially by reflexivity;
- (14.b) holds trivially since $P(x) \wedge I(x,y) \Rightarrow Q'(x,y)$. Moreover, the conjunction with the term $B(x')$ follows from the semantics of the while iteration, as shown in Ex. 1;
- (14.c) is a side condition in Manna-Pnueli rule (there should be a convergence function u into a well-founded set $\langle \mathcal{W}, \preceq \rangle$ with $x \lesdot y$ if and only if $u(x) \preceq u(y)$);
- Since $\llparenthesis\, P'(x) \,\rrparenthesis\; \mathtt{S}\; \llparenthesis\, Q'(x,x') \,\rrparenthesis$ denotes $\forall x \,.\, P'(x) \Rightarrow (Q'(x,S(x)) \wedge S(x) \neq \bot)$ where $S = [\![\mathtt{S}]\!]$ is the denotational semantics of $\mathtt{S}$, (16.i) implies both
  - $\forall x \,.\, (P(x) \wedge B(x)) \Rightarrow (P(S(x)) \wedge x \gtrdot S(x))$, which is (14.d);
  - and
    $\forall x \,.\, (P(x) \wedge B(x)) \Rightarrow Q(x,S(x))$
    which together with (16.ii)
    $\forall x, x'' \,.\, (Q(x,S(x)) \wedge Q(S(x),x'') \Rightarrow Q(x,x'')$
    yields
    $\forall x \,.\, (P(x) \wedge B(x) \wedge Q(S(x),x'')) \Rightarrow Q(x,x'')$, which is (14.e);
- (14.e′) is exactly (16.ii).

By Th. 13, Moreover, if $(\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})x \neq \bot$ then $\neg B(\mathsf{lfp}^{\sqsubseteq} F_{\mathtt{W}})$, as shown in Ex. 1, we conclude that Manna-Pnueli rule is sound.

Obviously Manna-Pnueli rule (16) is not complete (since $Q(x,x')$ might not be inductive), but it can be applied to the strongest invariant and the conclusion derived by the consequence rule.

16 Conclusion

Park/fixpoint induction is useful to reason on post-fixpoints, above the least fixpoint. Scott/iteration induction is useful to reason on iterates, below the least fixpoint. Traditional Park/fixpoint induction can prove invariance/partial correctness but not termination (at least without introducing auxiliary variables such as bounded loop counters [15]). The traditional Scott/iteration induction cannot prove termination either. We generalized the iteration induction principles to prove termination/total correctness. For termination they are equivalent to the Turing/Floyd termination proof method using variant/convergence functions (which itself is equivalent [10] to Burstall's intermittent assertions induction principle [3]). This applies both to (first-order) functional and imperative programming. In particular, the Manna-Pnueli method for proving the total correctness of `while` loops is equivalent to Scott induction for the denotational semantics of these loops.

Acknowledgements

I thank Francesco Ranzato for comments and suggestions on earlier versions of the paper. Work supported in part by NSF Grant CCF-1617717.

References

1. Apt, K.R., Plotkin, G.D.: Countable nondeterminism and random assignment. J. ACM **33**(4), 724–767 (1986)
2. de Bakker, J.W., Scott, D.S.: A theory of programs (Aug 1969), IBM Seminar Vienna, Austria (Unpublished notes)
3. Burstall, R.M.: Program proving as hand simulation with a little induction. In: IFIP Congress. pp. 308–312. North-Holland (1974)
4. Clarke Jr., E.M.: Programming language constructs for which it is impossible to obtain good hoare axiom systems. J. ACM **26**(1), 129–147 (1979)
5. Cook, S.A.: Soundness and completeness of an axiom system for program verification. SIAM J. Comput. **7**(1), 70–90 (1978)
6. Cook, S.A.: Corrigendum: Soundness and completeness of an axiom system for program verification. SIAM J. Comput. **10**(3), 612 (1981)
7. Cousot, P.: Méthodes itératives de construction et d'approximation de points fixes d'opérateurs monotones sur un treillis, analyse sémantique de programmes (in French). Thèse d'État ès sciences mathématiques, Université de Grenoble Alpes, Grenoble, France (21 March 1978)
8. Cousot, P., Cousot, R.: Constructive versions of Tarski's fixed point theorems. Pacific Journal of Mathematics **82**(1), 43–57 (1979)
9. Cousot, P., Cousot, R.: Induction principles for proving invariance properties of programs. In: Néel, D. (ed.) Tools & Notions for Program Construction: an Advanced Course. pp. 75–119. Cambridge University Press, Cambridge, UK (Aug 1982)
10. Cousot, P., Cousot, R.: ”a la burstall” intermittent assertions induction principles for proving inevitable ability properties of programs. Theor. Comput. Sci. **120**(1), 123–155 (1993)
11. Damm, W., Josko, B.: A sound and relatively* complete hoare-logic for a language with higher type procedures. Acta Inf. **20**, 59–101 (1983)
12. Floyd, R.W.: Assigning meaning to programs. In: Schwartz, J. (ed.) Proc. Symp. in Applied Math., vol. 19, pp. 19–32. Amer. Math. Soc. (1967)
13. Heizmann, M., Jones, N.D., Podelski, A.: Size-change termination and transition invariants. In: SAS. Lecture Notes in Computer Science, vol. 6337, pp. 22–50. Springer (2010)
14. Hoare, C.A.R.: An axiomatic basis for computer programming. Commun. ACM **12**(10), 576–580 (1969), http://doi.acm.org/10.1145/363235.363259
15. Katz, S., Manna, Z.: A closer look at termination. Acta Inf. **5**, 333–352 (1975)
16. Kleene, S.C.: Introduction to Meta-Mathematics. Elsevier North-Holland Pub. Co. (1952)
17. Lee, C.S., Jones, N.D., Ben-Amram, A.M.: The size-change principle for program termination. In: POPL. pp. 81–92. ACM (2001)
18. Leroy, X., Doligez, D., Frisch, A., Garrigue, J., Rémy, D., Vouillon, J.: The OCaml system, release 4.08, Documentation and user's manual (Feb 2019), http://caml.inria.fr/pub/docs/manual-ocaml/, copyright © 2013 Institut National de Recherche en Informatique et en Automatique

19. Manna, Z., Ness, S., Vuillemin, J.: Inductive methods for proving properties of programs. Commun. ACM **16**(8), 491–502 (1973)
20. Manna, Z., Pnueli, A.: Axiomatic approach to total correctness of programs. Acta Inf. **3**, 243–263 (1974)
21. Manna, Z., Vuillemin, J.: Fix point approach to the theory of computation. Commun. ACM **15**(7), 528–536 (1972)
22. Markowsky, G.: Chain-complete posets and directed sets with applications. Algebra Universalis **6**(1), 53–68 (Nov 1976)
23. Park, D.M.R.: On the semantics of fair parallelism. In: Abstract Software Specifications. Lecture Notes in Computer Science, vol. 86, pp. 504–526. Springer (1979)
24. Régis-Gianas, Y., Pottier, F.: A Hoare logic for call-by-value functional programs. In: MPC. Lecture Notes in Computer Science, vol. 5133, pp. 305–335. Springer (2008)
25. Schmidt, D.W.: Denotational Semantics: A Methodology for Language Development. William C. Brown Publishers, Dubuque, IA, USA (Jun 1988), http://people.cs.ksu.edu/~schmidt/text/DenSem-full-book.pdf
26. Scott, D.S.: Outline of a mathematical theory of computation. In: Proceedings of the Fourth Annual Princeton Conference on Information Sciences and Systems. pp. 169–176. Princeton University (Mar 1970)
27. Scott, D.S.: The lattice of flow diagrams. In: Symposium on Semantics of Algorithmic Languages, Lecture Notes in Mathematics, vol. 188, pp. 311–366. Springer (1971)
28. Tarski, A.: A lattice theoretical fixpoint theorem and its applications. Pacific J. of Math. **5**, 285–310 (1955)
29. Turing, A.: Checking a large routine. In: Report of a Conference on High Speed Automatic Calculating Machines, University of Cambridge Mathematical Laboratory, Cambridge, England. pp. 67–69 (1949), http://www.turingarchive.org/browse.php/b/8