

Inductive Fixpoints in Higher Order Logic

Sava Krstić

OGI School of Science and Engineering
Oregon Health and Sciences University

Abstract

We show that an analogue of the domain-theoretic least fixpoint operator can be defined in a purely set-theoretic framework. It can be formalized in classical higher order logic, serving as a solid foundation for proving termination of (possibly nested) recursive programs in a variety of mechanized proof systems.

1. Introduction

The standard denotation of recursively defined programs is domain-theoretical: a recursive declaration is interpreted as a continuous functional whose least fixpoint is then the program’s meaning.¹ In practice, however, reasoning about recursive programs typically goes without mentioning domains and least fixpoints. For example, since termination amounts to wellfoundedness of the calling relation², we can in many cases of interest prove it using the information about the calling relation that can be easily discerned from the program declaration. Paulson remarks in [12], where he analyzes several termination proofs of a complex nested recursive program, that “perhaps domains and partial objects are not essential even for difficult proofs of termination”. The goal of this paper is to substantiate this claim with a little theory of fixpoints of functionals in a purely set-theoretical setting. This theory can be formalized in classical Higher Order Logic (*HOL*), a powerful and popular framework for mechanized theorem proving.

HOL is a logic of total functions and there are several ways of modeling partiality in it [10]. For us, a partial function of type $A \Rightarrow B$ with domain D is represented by a total function of type $A \Rightarrow B$, we just do not care about the values of this total function for arguments outside of D . Any two functions $f, f' : A \Rightarrow B$ that agree on D (for which we

will use the notation $f =_D f'$) are to be considered representatives of the same partial function with domain D . This treatment of partiality is commonly used in formalized reasoning about programs [11, 14]. It calls for the following notion of fixpoint.

Definition 1 Given $F : (A \Rightarrow B) \Rightarrow (A \Rightarrow B)$ and $D \subseteq A$, we say that $h : A \Rightarrow B$ is a fixpoint of F on D if $F h' =_D h'$ holds for every h' satisfying $h' =_D h$.

The concept of termination of a program on a subset of its input type can be formalized as follows. The definition takes as a parameter a wellfounded relation capturing the program’s calling relation.

Definition 2 Let ρ be a wellfounded relation on D . A function $h : A \Rightarrow B$ is an inductive fixpoint of F on D with respect to ρ when the following condition holds:

$$\forall f. \forall x \in D. f =_{\rho^{-1}x} h \longrightarrow F f x = h x. \quad (1)$$

It is easy to check that such h is a fixpoint in the sense of Definition 1. Also, it follows by wellfounded induction along ρ that the condition (1) characterizes h uniquely as a partial function.

We introduced inductive fixpoints in [8] and later learned that in a different guise they were used by Harrison in the proof of a recursion theorem in [6]. As a model of termination, they are not perfect, since some computational information is lost in regarding terms equivalent modulo the ground theory as equal. (Thus, the program $f x = f x - f x$ is considered terminating for all inputs.) But this is the inevitable price of the shallow embedding of programs into *HOL*.

The main result of the paper is the following theorem expressing the existence of the largest inductive fixpoint and its uniqueness in a strong sense.

Theorem 1 For every functional F , there exists an inductive fixpoint whose domain contains the domain of any other inductive fixpoint. Moreover, this maximal inductive fixpoint agrees with any other (inductive or not) fixpoint of F on the intersection of their domains.

¹See any book on denotational semantics, e.g. [15].

²By an informal definition, y is related to x if during the execution of our program on input x , the program recursively calls itself with input y .

The largest domain for F will be denoted D_F and called *the inductive domain* of F . The *largest inductive fixpoint* νF is defined as an arbitrarily chosen inductive fixpoint of F on D_F .

The uniqueness part of Theorem 1 is proved in Section 2, where we also give a review of a technique for proving termination, developed in collaboration with John Matthews [8]. Section 3 is devoted to the proof of the existence part of Theorem 1. In Section 4 we show that an alternative fixpoint operator can be defined by a suitable version of the iterative construction. Theorem 3 shows that the two fixpoints are equal in most cases of interest.

2. Fixpoints and *HOL* Termination

Throughout the paper, F is an arbitrary functional of type $(A \Rightarrow B) \Rightarrow (A \Rightarrow B)$, D is a subset of A , and ρ is a wellfounded relation on D .

In some sense, the most natural partial function associated to a recursive program is the optimal fixpoint of Manna and Shamir; the definitions of [9] translate into our context as follows.

- Definition 3** (a) A fixpoint h of F on D is *fixp-consistent* if for every other fixpoint h' on D' one has $h =_{D \cap D'} h'$.
- (b) A *fixp-consistent* fixpoint h of F on D is *optimal* if D contains the domain of every other *fixp-consistent* fixpoint of F .

Only *fixp-consistent* fixpoints are of interest because they guarantee unambiguous outputs for all inputs in their domain. Thus, the following result gives legitimacy to (all) inductive fixpoints. It also implies the uniqueness part of Theorem 1.

Proposition 1 *Inductive fixpoints are fixp-consistent.*

Proof. Let h be an inductive fixpoint of F on D with respect to ρ . To prove that h is *fixp-consistent*, let g be any other fixpoint on E and define

$$g' x = \begin{cases} g x & \text{if } x \in E \\ h x & \text{otherwise.} \end{cases}$$

Since $g' =_E g$, to deduce $g =_{D \cap E} h$, it will suffice to prove that $g' x = h x$ holds for every $x \in D$. We argue by induction along ρ . If $x \notin E$, our goal is true by definition of g' . Consider the remaining case where $x \in D \cap E$. The induction hypothesis $g' =_{\rho^{-1}x} h$ implies $F g' x = h x$. Finally, we have $F g' x = g' x$ because $g' =_E g$ and g is a fixpoint on E . \square

The optimal fixpoint carries the maximum unambiguous information implied by a recursive declaration, but it ignores the computational aspect entirely. For example, the

optimal fixpoint of the program $f x = f x + f x$ is the constant zero function, while the least fixpoint is completely undefined. Being mostly interested in program termination, we will content ourselves with just confirming the existence of optimal fixpoints in our *HOL* model.

Proposition 2 *Every functional has an optimal fixpoint.*

Proof. Consider the collection $\{(D_i, h_i)\}_{i \in I}$ of all pairs such that h_i is a *fixp-consistent* fixpoint of F on D_i . Let D be the union of all the D_i and let h be a function such that $h =_{D_i} h_i$ holds for every i . Since $h_i =_{D_i \cap D_j} h_j$, such function h exists. Since $h =_{D_i} h_i$, it is a fixpoint on D_i for every i , so it is a fixpoint on D . If h' is any fixpoint of F on D' , then $h_i =_{D_i \cap D'} h'$ (because h_i is *fixp-consistent*), so $h =_{D_i \cap D'} h'$ for every i , so $h =_{D \cap D'} h'$. \square

2.1. A Proof Technique for Termination

If f is the domain-theoretic fixpoint of F , then to check termination of F on a subset D of A one needs to prove $f x \neq \perp$ for every $x \in D$. Typically such a proof is done by wellfounded induction, where the user needs to supply a wellfounded relation that contains the calling relation associated with F . Our approach looks more complicated at the moment, because for a termination proof with a given F and (a correctly guessed) ρ , we need to prove the existence of the inductive fixpoint. However, we will see in Lemma 1 below that, given F and ρ , a candidate inductive fixpoint is possible to define directly. And then Theorem 2 gives a practical method for proving that this candidate is indeed an inductive fixpoint.

Consider the equation

$$\forall x. \psi x = F (\psi \upharpoonright_{\rho^{-1}x}) x, \quad (2)$$

where the notation $f \upharpoonright_{A'}$ is for the “restriction” of the function $f: A \Rightarrow B$ on A' . This restriction is defined by

$$(f \upharpoonright_{A'}) x = \begin{cases} f x & \text{if } x \in A' \\ \text{Arb} & \text{otherwise,} \end{cases}$$

where *Arb* is an arbitrary³ element of B .

The equation (2) embodies the wellfounded recursion principle, and it is intuitively clear that there is a unique function ψ satisfying it. We will use the notation $\psi_{\rho, F}$ for this function.

The functions $\psi_{\rho, F}$ are at the core of the method developed by Nipkow and Slind [11, 14] that has been used for termination proofs in *Isabelle/HOL*. The current implementation of *Isabelle/HOL* contains a higher order function *wfrec* such that *wfrec* ρ $F = \psi_{\rho, F}$ holds for every ρ and F . The function *wfrec* provides a definitional mechanism that plays the role of the fixpoint operator in *HOL*.

³In *HOL* implementations, *Arb* is defined using Hilbert’s choice operator, e.g. *Arb* $\equiv \epsilon z. \text{True}$.

Lemma 1 ([8]) *An inductive fixpoint of F on D with respect to ρ exists if and only if $\psi_{\rho,F}$ is such a fixpoint.* \square

Thus, the termination of F on D in our model amounts to $\psi_{\rho,F}$ being an inductive fixpoint of F on D for a suitable ρ . The recursion equation

$$\forall x \in D. \psi_{\rho,F} x = F \psi_{\rho,F} x \quad (3)$$

is a somewhat weaker statement. In the current practice, however, one is satisfied by checking (3) in order to prove termination. Notably, the *redef* package implemented by Slind for *Isabelle/HOL* takes F, ρ, D as inputs, extracts termination conditions⁴ from the expression defining F , and derives automatically from them the equality of the right-hand sides of (2) and (3). Thus, the proof of (3) reduces to the proof of termination conditions—a task that also can often be done automatically.

To see that satisfying the recursion equation and being an inductive fixpoint are not equivalent concepts, consider the functional

$$F f x \equiv \mathbf{if} \ x = 0 \ \mathbf{then} \ f(1) - f(0) \ \mathbf{else} \ f(x - 1).$$

The recursive program corresponding to F does not terminate on any input, and that is faithfully reflected in F not having an inductive fixpoint on any non-empty set. (Otherwise, there would exist at least one value x —a minimal element with respect to the calling relation—such that $F f x = F f' x$ holds for any f, f' . Such a value clearly does not exist for our F .) On the other hand, the constant zero function is equal to $\psi_{<,F}$ and it is also (the only) fixpoint of the functional F .⁵

It has been known that the *contraction condition* for the functional F with respect to the wellfounded relation ρ ,

$$\forall f g. \forall x \in D. f =_{\rho^{-1}x} g \longrightarrow F f x = F g x,$$

implies the unique existence of a fixpoint of F on D [6]. It is in fact true that the contraction condition implies that F has an inductive fixpoint on D . One can also check that proving the termination conditions extracted by the *redef* mechanism of *Isabelle/HOL* suffices for a proof of the contraction condition, so the proofs done with *redef* effectively prove inductive invariance of every introduced $\psi_{\rho,F}$ even though they formally derive only the recursion equation (3).

The real difficulty with termination is in the cases of nested recursion, where the occurrence of recursive calls within recursive calls conceals the calling pattern and the

⁴Termination conditions say that the expressions occurring as arguments in recursive calls are ρ -smaller than the input variable.

⁵Replacing *Arb* in the definition of the restriction functions $f \upharpoonright A'$ with *Arb* x , where this new *Arb* is an arbitrary function of type $A \Rightarrow B$, would eliminate this particular example, but others could be fabricated.

contraction condition fails in general. Consider this simple program taken from [13]:

$$g x \equiv \mathbf{if} \ x = 0 \ \mathbf{then} \ 0 \ \mathbf{else} \ g(g(x - 1)).$$

It is intuitively clear that, no matter what formal reasoning system one uses, to prove by induction that this algorithm terminates for all inputs, one must prove some stronger statement, for example that “for every x , the algorithm terminates with input x and returns the value 0”, or that “for every x , the algorithm terminates with input x and returns a value not greater than x .” In [8], we observed that this is typical of termination proofs for nested recursive programs: one needs to know in advance a property of the program (like $g x = 0$ or $g x \leq x$ in our example), and that property has to be of a special form that we termed *inductive invariant*. By definition, an input-output predicate $S: A \Rightarrow B \Rightarrow \text{bool}$ is an inductive invariant of F when

$$\forall f x. (\forall y. \rho y x \longrightarrow S y (f y)) \longrightarrow S x (F f x).$$

It can be checked that $\psi_{\rho,F}$ satisfies all inductive invariants. Note also that a function h is an inductive fixpoint if and only if its graph S_h (defined by $S_h x y$ iff $y = h x$) is an inductive invariant.

The following theorem yields a practical technique for proving termination with the help of inductive invariants.

Theorem 2 ([8]) *Suppose S is an inductive invariant of F with respect to ρ . If the restricted contraction condition*

$$\forall f g. \forall x \in D. f =_{\rho^{-1}x} g \wedge S_g \subseteq S \longrightarrow F f x = F g x.$$

is satisfied, then $\psi_{\rho,F}$ is an inductive fixpoint of F . \square

Remarks. 1. The inductive invariants technique has been fully formalized and will be included in the next release of *Isabelle/HOL* [7].

2. Dubois and Donzeau-Gouge [5] have also pointed out the importance of user-given input-output relations (“post-conditions”) for termination proofs of nested recursive programs.

3. Techniques for proving termination of nested recursive programs in intuitionistic type theory are given in [5, 4, 3].

3. Proof of Theorem 1

We will say that a subset D of A is a *domain* for F if F has an inductive fixpoint on D .

By Lemma 1, for every domain D there exists a wellfounded relation ρ (not necessarily unique) such that $\psi_{\rho,F}$ is an inductive fixpoint of F on D . We will say that (D, ρ) is a *germ* of F in such cases. We will consider an ordering \preceq on the set of germs, use Zorn’s Lemma to prove that there exists a maximal germ with respect to that ordering,

and then show that the domain corresponding to a maximal germ contains every other domain for F . This will prove Theorem 1 since its uniqueness part has already been checked.

Define $(D', \rho') \preceq (D, \rho)$ to mean that D' is a downward closed subset of (D, ρ) and ρ' is the restriction of ρ on D' . (The definition makes sense for every two sets D, D' and binary relations ρ, ρ' on them.) It is easy to check that the relation \preceq is a partial ordering.

Lemma 2 *Let (D, ρ) be a germ of F and $(D', \rho') \preceq (D, \rho)$. Then (D', ρ') is a germ of F and $\psi_{\rho', F} =_{D'} \psi_{\rho, F}$.*

Proof. Immediate from Proposition 1. \square

Suppose $\mathcal{C} = \{(D_i, \rho_i) \mid i \in I\}$ is a chain of germs in the ordering \preceq and consider $(D, \rho) = (\cup_{i \in I} D_i, \cup_{i \in I} \rho_i)$. First we check that the relation ρ must be wellfounded. Arguing by contradiction, assume that x_1, x_2, \dots is an infinite sequence of elements of D such that $\rho(x_{n+1}, x_n)$ holds for every n . Let i be such that $\rho_i(x_2, x_1)$. If we can show that $x_n \in D_i$ for every n , that would imply that $\rho_i(x_{n+1}, x_n)$ holds for every n . This would contradict the assumption that ρ_i is wellfounded and so finish the proof that ρ is wellfounded. To complete this argument, it remains to prove $x_n \in D_i$ for every n . Arguing by induction, we see that in fact it suffices to prove only $x_3 \in D_i$. Assume the contrary: $x_3 \notin D_i$. Then we must have $\rho_j(x_3, x_2)$ for some j such that $x_3 \in D_j \setminus D_i$. Since \mathcal{C} is a chain, it follows that D_i is a downward closed subset of D_j . Thus, from $x_2 \in D_i$ and $\rho_j(x_3, x_2)$ it follows that $x_3 \in D_i$, directly contradicting our assumption.

Note that $(D_i, \rho_i) \preceq (D, \rho)$ holds for every i , so (D, ρ) is an upper bound for the chain \mathcal{C} , but we also need to check that (D, ρ) is a germ. We do it by producing an inductive fixpoint g on D with respect to ρ . Define g by

$$g x = \begin{cases} g_i x & \text{if } x \in D_i \\ \text{Arb} & \text{if } x \notin D, \end{cases}$$

where $g_i = \psi_{\rho_i, F}$. By Lemma 2, g is well defined, i.e. $g_i x = g_j x$ when $x \in D_i \cap D_j$. To prove that g is an inductive fixpoint on D we need to check that

$$F f x = g x \tag{4}$$

holds for every f and for every $x \in D$ such that $f =_{\rho^{-1}x} g$. Fixing f and x , we may assume that $x \in D_i$ for some i . Since $(D_i, \rho_i) \preceq (D, \rho)$, we have $\rho^{-1}x = \rho_i^{-1}x$ and so $f =_{\rho_i^{-1}x} g_i$. Since g_i is an inductive fixpoint on D_i with respect to ρ_i , it follows that $F f x = g_i x$, which immediately implies (4).

By Zorn's Lemma, there exists a maximal germ, say (D, ρ) . We claim that D is the largest domain for F .

Assuming the contrary of the claim, suppose (D', ρ') is germ of F such that D' is not contained in D . Now,

there exists $z \in D' \setminus D$ such that all smaller elements (with respect to ρ') than z in D' belong to D as well. In view of Lemma 2, it is no loss of generality to assume that $D' = E \cup \{z\}$, where $E \subseteq D$ and z is the greatest element in D' .

Let $\psi = \psi_{\rho, F}$ and $\psi' = \psi_{\rho', F}$. By Lemma 2, E is a domain for F , being downward closed in D' . It is a subdomain of both D and D' , so by Proposition 1 we obtain that the restrictions of ψ and ψ' on E coincide.

Define the function h by

$$h x = \begin{cases} \psi x & \text{if } x \in D \\ \psi' x & \text{otherwise.} \end{cases}$$

Since ψ and ψ' have equal restrictions on $E = D \cap D'$, we have that $h =_D \psi$ and $h =_{D'} \psi'$ and so h is an inductive fixpoint of F on both D and D' .

We want to prove that h is an inductive fixpoint of F on $D \cup D' = D \cup \{z\}$. Using the (obviously wellfounded) relation $\sigma = \rho \cup \{(x, z) \mid x \rho' z\}$ on $D \cup \{z\}$, it suffices to prove that

$$f =_{\sigma^{-1}x} h \longrightarrow F f x = h x \tag{5}$$

holds for every f and every $x \in D \cup \{z\}$. Consider first the case when $x \in D$. Then $\sigma^{-1}x = \rho^{-1}x$ and (5) follows since h is an inductive fixpoint of F on D . In the remaining case we have $x = z$, so $\sigma^{-1}z = \rho'^{-1}z$ and (5) follows since h is an inductive fixpoint of F on D' .

4. The Iterative Fixpoint Operator

Like the domain-theoretic least fixpoint, the function recursively defined by the functional $F: (A \Rightarrow B) \Rightarrow (A \Rightarrow B)$ can be seen as the limit of the sequence

$$\phi_0, \phi_1, \phi_2, \dots$$

where $\phi_0 = \text{Arb}$ and $\phi_{n+1} x = F \phi_n x$. But what is the limit partial function ϕ ? An obvious candidate would have the domain consisting of all x for which the sequence

$$\phi_0 x, \phi_1 x, \phi_2 x, \dots$$

stabilizes, and for such x define ϕx as the limit value of the sequence. However, the fixpoint equation does not have to hold for this (sometimes too large) domain, as demonstrated by the example

$$F f x = \begin{cases} x & \text{if } f \text{ is a constant function} \\ x - 1 & \text{otherwise.} \end{cases}$$

(In some constrained contexts, the fixpoint equation does hold on this domain; see [2].)

A more restrictive domain on which the stabilization limit function satisfies the fixpoint equation is obtained as

the union of an increasing sequence of subsets Δ_n of A , defined inductively by

$$\begin{aligned}\Delta_0 &= \emptyset \\ \Delta_{n+1} &= \{x \mid \forall f. f =_{\Delta_n} \phi_n \longrightarrow F f x = \phi_{n+1} x\}.\end{aligned}$$

Lemma 3 $\Delta_n \subseteq \Delta_{n+1}$ and $\phi_{n+1} =_{\Delta_n} \phi_n$, for every n .

Proof. Both statements are trivially true for $n = 0$. Proceeding by induction, for $\Delta_n \subseteq \Delta_{n+1}$ we need to prove $F f x = F \phi_n x$ assuming $f =_{\Delta_n} \phi_n$ and $x \in \Delta_n$. We will prove $F f x = F \phi_{n-1} x$ and $F \phi_n x = F \phi_{n-1} x$. The second equation follows from the definition of Δ_n and the induction hypothesis $\phi_n =_{\Delta_{n-1}} \phi_{n-1}$. The first equation similarly follows from the definition of Δ_n and $f =_{\Delta_{n-1}} \phi_{n-1}$. It remains just to prove this last relation. It follows by transitivity of $=_{\Delta_{n-1}}$ from the relations $f =_{\Delta_{n-1}} \phi_n$ and $\phi_n =_{\Delta_{n-1}} \phi_{n-1}$. The second of these two formulas is an already used induction hypothesis, while the first is a consequence of the induction hypothesis $\Delta_{n-1} \subseteq \Delta_n$ and our assumption $f =_{\Delta_n} \phi_n$.

To prove $\phi_{n+1} =_{\Delta_n} \phi_n$, assume $x \in \Delta_n$; the goal is $\phi_{n+1} x = \phi_n x$, or $F \phi_n x = \phi_n x$. This follows from the induction hypothesis $\phi_n =_{\Delta_{n-1}} \phi_{n-1}$ and the definition of Δ_n . \square

We define the *iterative domain* Δ_F of F as the union of the subsets Δ_n above. And we define the *iterative fixpoint* μF of F by

$$\mu F x = \begin{cases} \phi_n x & \text{if } x \in \Delta_n \\ \text{Arb} & \text{if } x \notin \Delta_F. \end{cases}$$

Proposition 3 μF is an inductive fixpoint of F on Δ_F .

Proof. Take the wellfounded relation ρ such that $x \rho y$ holds if and only if $x \in \Delta_n$ and $y \in \Delta_{n+1} \setminus \Delta_n$ for some n . Thus, to prove that μF is an inductive fixpoint of F on Δ with respect to ρ , we need to check that

$$f =_{\Delta_n} \mu F \longrightarrow F f x = \mu F x \quad (6)$$

holds for all f and $x \in \Delta_{n+1}$. We have $\mu F x = \phi_{n+1} x$ (since $x \in \Delta_{n+1}$) and also $\mu F =_{\Delta_n} \phi_n$. Thus, (6) can be rewritten as

$$f =_{\Delta_n} \phi_n \longrightarrow F f x = \phi_{n+1} x,$$

which is true by definition of Δ_{n+1} . \square

As an immediate consequence of Proposition 3, we have

$$\Delta_F \subseteq D_F \quad \text{and} \quad \mu F =_{\Delta_F} \nu F. \quad (7)$$

We are going to show that for all practical purposes the two domains D_F and Δ_F are equal. For an example when

Δ_F is a proper subset of D_F , take the following recursive definition of $f: \text{nat} \times \text{nat} \Rightarrow \text{bool}$:

$$\begin{aligned}f(0, 0) &= \forall i > 0. f(i, 0) \\ f(0, j) &= \text{True} \text{ if } j \neq 0 \\ f(i, j) &= f(i-1, j+1) \text{ if } i \neq 0\end{aligned}$$

Take the wellfounded relation ρ on $\text{nat} \times \text{nat}$ determined by the calling relation of this recursive definition. The minimal elements are $(0, j)$, the maximal element is $(0, 0)$, and the remaining elements are organized into chains of length n going from $(0, n)$ up to $(n, 0)$, for every $n > 0$. The contraction condition holds on the whole input type, so the natural domain is $D_F = \text{nat} \times \text{nat}$ (and we can prove that $\mu F(i, j) = \text{True}$). On the other hand, we have $\Delta_n = \{(i, j) \mid i < n\} \setminus \{(0, 0)\}$ and so the iterative domain $\Delta_F = \text{nat} \times \text{nat} \setminus \{(0, 0)\}$ is strictly smaller than D_F .

Definition 4 A finitary inductive fixpoint of F on D is an inductive fixpoint on D with respect to a wellfounded relation under which every element of D has finite height⁶.

Theorem 3 $D_F = \Delta_F$ (and $\nu F = \mu F$) if and only if F has a finitary inductive fixpoint on D_F .

Proof. Clearly, μF is a finitary inductive fixpoint on Δ_F . In view of (7), it suffices to prove that $D \subseteq \Delta_F$ holds whenever F has a finitary inductive fixpoint on D .

Let h be a finitary inductive fixpoint as in Definition 4, with respect to a suitable wellfounded relation ρ . Let D_n denote the subset of D consisting of all its elements of height n or less. (Thus, $D_0 = \emptyset$ and D_1 is the set of minimal elements under ρ .) It will suffice to prove

$$D_n \subseteq \Delta_n \quad \text{and} \quad \phi_n =_{D_n} h, \quad (8)$$

which we do by induction on n . The case $n = 0$ is trivial.

With (8) as the induction hypothesis, we need to prove that for every $x \in D_{n+1}$

$$x \in \Delta_{n+1} \quad \text{and} \quad \phi_{n+1} x = h x. \quad (9)$$

Using the induction hypothesis, we can strengthen our assumption to that the height of x is exactly $n + 1$, which implies that $\rho^{-1} x$ is a subset of D_n . Thus, from the induction hypothesis we have $\phi_n =_{\rho^{-1} x} h$. Since h is an inductive fixpoint, this implies $F \phi_n x = h x$, which is the second goal of (9). For the first goal of (9), by definition of Δ_{n+1} , it suffices to prove $F f x = \phi_{n+1} x$ with an additional assumption $f =_{\Delta_n} \phi_n$. We have already obtained $\phi_{n+1} x = h x$, so it suffices to prove $F f x = h x$. Since h is an inductive fixpoint, we only need to check $f =_{\rho^{-1} x} h$.

⁶The *height* of x is the supremum of the lengths of downward chains starting at x .

This follows from our assumption $f =_{\Delta_n} \phi_n$, both parts of the induction hypothesis, and $\rho^{-1}x \subseteq D_n$. \square

The following proposition shows that the most of the inductive fixpoints occurring in practice are finitary. Notice the common special case when the sets $\rho^{-1}x$ are all finite.

Proposition 4 *Suppose h is an inductive fixpoint of F on D with respect to ρ and suppose that for every $x \in D$ there exists a finite subset $D(x)$ of $\rho^{-1}x$ such that*

$$\forall f. \forall x \in D. f =_{D(x)} h \longrightarrow F f x = h x.$$

Then F has a finitary inductive fixpoint on D .

Proof. Let ρ' be the wellfounded relation defined by $\rho'^{-1}x = D(x)$. It is easy to check that h is an inductive fixpoint of F on D with respect to ρ' . It remains show that every element of D has finite height with respect to ρ' . Indeed, since there are only finitely many elements smaller (under ρ') than x , the existence of arbitrarily long downward chains would by König's Lemma imply the existence of an infinite downward chain, which would contradict wellfoundedness of ρ' . \square

In the remainder of the paper we show the $D_F = \Delta_F$ would hold in general if we allowed transfinite iteration in the definition of Δ_F . For every ordinal α , we define

$$\begin{aligned} \phi_{\alpha+1} &= F \phi_\alpha \\ \Delta_{\alpha+1} &= \{x \mid \forall f. f =_{\Delta_\alpha} \phi_\alpha \longrightarrow F f x = \phi_{\alpha+1} x\} \end{aligned}$$

as before, and for a limit ordinal α we define

$$\begin{aligned} \phi_\alpha &= \begin{cases} \phi_\beta x & \text{if } x \in \Delta_\beta \text{ for some } \beta < \alpha \\ \text{Arb} & \text{otherwise} \end{cases} \\ \Delta_\alpha &= \bigcup_{\beta < \alpha} \Delta_\beta. \end{aligned}$$

Generalizing Proposition 3, we have that for every limit ordinal α , the function ϕ_α is an inductive fixpoint of F on Δ_α . Also, with little additional effort, the main argument in the proof of Theorem 3 can be extended to show that $D \subseteq \Delta_{|\rho|}$ holds whenever F has an inductive fixpoint on D with respect to ρ . Here, the ordinal $|\rho|$ is defined as the supremum of ordinals $|x|_\rho + 1$, where the ordinal $|x|_\rho$ is defined for every $x \in D$ inductively by $|x|_\rho = \sup\{|y|_\rho + 1 \mid y \rho x\}$ [1].

It follows that the increasing sequence of iterative domains Δ_α stabilizes at D_F :

Theorem 4 $D_F = \Delta_{|\rho|}$, where ρ is a wellfounded relation with respect to which F has an inductive fixpoint on D_F . \square

Acknowledgments I am grateful to John Matthews for stimulating discussions and the previous collaboration that motivated the research reported in this paper.

References

- [1] P. Aczel. An introduction to inductive definitions. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 739–782. North-Holland, Amsterdam, 1977.
- [2] A. Balaa and Y. Bertot. Fonctions récursives générales par itération en théorie des types. In *Journées francophones des langages applicatifs (JFLA'02)*. INRIA, 2002.
- [3] Y. Bertot, V. Capretta, and K. D. Barman. Type-theoretic functional semantics. In V. A. Carreno, C. A. Munoz, and S. Tahar, editors, *Theorem Proving in Higher Order Logics (TPHOLS'02)*, volume 2410 of *LNCS*, pages 83–98. Springer, 2002.
- [4] A. Bove and V. Capretta. Nested general recursion and partiality in type theory. In R. J. Boulton and P. B. Jackson, editors, *Theorem Proving in Higher Order Logics (TPHOLS'01)*, volume 2152 of *LNCS*, pages 121–135. Springer, 2001.
- [5] C. Dubois and V. Donzeau-Gouge. A step towards the mechanization of partial functions: domains as inductive predicates. In M. Kerber, editor, *CADE-15 Workshop on Mechanization of Partial Functions*, pages 53–62, 1998.
- [6] J. Harrison. Inductive definitions: Automation and application. In E. T. Schubert, P. J. Windley, and J. Alves-Foss, editors, *Higher Order Logic Theorem Proving and Its Applications*, volume 971 of *LNCS*, pages 200–213. Springer, 1995.
- [7] Isabelle home page. <http://isabelle.in.tum.de/>.
- [8] S. Krstić and J. Matthews. Inductive invariants for nested recursion. In D. Basin and B. Wolff, editors, *Theorem Proving in Higher Order Logics (TPHOLS'03)*, volume 2758 of *LNCS*, pages 253–269. Springer, 2003.
- [9] Z. Manna and A. Shamir. The optimal approach to recursive programs. *Communications of the ACM*, 20(11):824–831, 1977.
- [10] O. Müller and K. Slind. Treating partiality in a logic of total functions. *The Computer Journal*, 40(10), 1997.
- [11] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.
- [12] L. C. Paulson. Proving termination of normalization functions for conditional expressions. *Journal of Automated Reasoning*, 2(1):63–74, 1986.
- [13] K. Slind. Function definition in higher order logic. In J. von Wright et al., editor, *Theorem Proving in Higher Order Logics (TPHOLS'96)*, volume 1125 of *LNCS*, pages 381–397. Springer, 1996.
- [14] K. Slind. *Reasoning about Terminating Functional Programs*. PhD thesis, Institut für Informatik, Technische Universität München, 1999.
- [15] G. Winskel. *The Formal Semantics of Programming Languages: an Introduction*. MIT Press, 1993.