Inductive Construction in Nuprl Type Theory Using Bar Induction

Mark S Bickford & Robert Constable

Cornell University

Contructive type theories such as Coq, Agda, and Nuprl all have some powerful primitive form of inductive construction. The soundness of the rules for these inductive constructions can be difficult to prove. In this note we show that one powerful form of inductive construction, *parameterized families of W-types*, can be internally constructed in type theory using a general form of Brouwer's bar induction rule and induction on a primitive type of natural numbers, from types that need not be defined inductively. We first construct the corecursive family of non-wellfounded types and then construct their wellfounded parts in such a way that the desired induction principle follows from bar induction. All the results have been formally proved in Nuprl, and details can be found here: `http://www.nuprl.org/LibrarySnapshots/Published/Version1/Standard/co-recursion/sbi-param-W-induction.html`.

$S \sqsubseteq T$ means that type S is a subtype of type T. A type function F is monotone if $S \sqsubseteq T \Rightarrow F(S) \sqsubseteq F(T)$, and preserves ω-limits if $\bigcap_{n\in\mathbb{N}} F(X_n) \sqsubseteq F(\bigcap_{n\in\mathbb{N}} X_n)$. A type T is a fixed point of F if $T \sqsubseteq F(T)$ and $F(T) \sqsubseteq T$. For any type T, $T \sqsubseteq Top$, where $Top = \bigcap_{x\in Void} Void$. For any monotone, ω-limit preserving function F, the type $\mathrm{corec}(F) = \bigcap_{n\in\mathbb{N}} F^n(\mathrm{Top})$, where the iteration F^n is defined by primitive recursion, is the greatest fixed point of F. We often write A_p rather than $A(p)$ and $B_{p,a}$ rather than $B(p,a)$.

Parameterized families of co-W and W-types. For parameter type P and functions $A \in P \to \mathrm{Type}$, $B \in p\colon P \to A_p \to \mathrm{Type}$, and $C \in p\colon P \to a\colon A \to B_{p,a} \to P$, the family $W_{A,B,C}(p)$ is the least fixed point of the functional $F_{A,B,C}$ on type families $G \in P \to \mathrm{Type}$ defined by

$$F_{A,B,C}(G) = \lambda p.\, a\colon A \times (b\colon B_{p,a} \to G(C_{p,a,b}))$$

Since $F_{A,B,C}$ is monotone and preserves ω-limits (on type families), we can easily construct the greatest fixed point family, $coW_{A,B,C}$, as follows:

$$coW_{A,B,C} = \lambda p.\, \bigcap_{n\in\mathbb{N}} F^n_{A,B,C}(\lambda q.\,\mathrm{Top})(p)$$

Then, for $S_{A,B,C} = p\colon P \times w\colon coW_{A,B,C} \times (B_{p,\pi_1(w)} + \mathrm{Unit})$, a path has type $\mathrm{Path}^{A,B,C} = \{s\colon \mathbb{N} \to S_{A,B,C} \mid \forall n\colon \mathbb{N}.\, \mathrm{con}(s(n), s(n+1))\}$ where

$$\mathrm{con}(\langle p, \langle a, f\rangle, d_1\rangle, \langle q, w_2, d_2\rangle) \Leftrightarrow (d_1 = \mathrm{inl}(b) \Rightarrow (q = C_{p,a,b} \wedge w_2 = f(b)))$$

A path s halts, $\mathrm{halts}(s)$, if $\downarrow\exists n\colon \mathbb{N}.\, \exists p.\exists w.\, s(n) = \langle p, w, \mathrm{inr}()\rangle$, where the *squash* of a type T is the type $\downarrow T$ that is empty if T is empty and is Unit if T is non-empty. Paths that start at p, w have type

$$\mathrm{Path}^{A,B,C}_{p,w} = \left\{ s\colon \mathrm{Path}^{A,B,C} \mid \exists d.\, s(0) = \langle p, w, d\rangle \right\}$$

and we define the type $W_{A,B,C}(p)$ by

$$W_{A,B,C}(p) = \left\{ w\colon coW_{A,B,C} \mid \forall s : \mathrm{Path}^{A,B,C}_{p,w}.\ \mathrm{halts}(s) \right\}$$

It is relatively straightforward to prove that $W_{A,B,C}$ is a fixed point of the functional $F_{A,B,C}$.

To show that it is the least fixed point we use bar induction to prove that its induction principle is witnessed by:

$$\lambda C.\lambda ind.\lambda par.\lambda w.\ \texttt{letrec}\ F(p,w)\ =\ \texttt{let}\ a,f=w\ \texttt{in}\ ind(p,a,f,\lambda b.F(C[p;a;b],f(b)))\ \texttt{in}\ F(par;w)$$

Bar Induction. A finite sequence s of length k has type $V_k(T) = \mathbb{N}_k \to T$, and we append t to s using $s \oplus_k t = \lambda i.$ if $i < k$ then $s(i)$ else t. Our bar induction rule is restricted to conclusions of the form $a(k,s) \in X(k,s)$, which, in Nuprl, have trivial constructive content. Let $ind(R,T,a,X,k,s,t)$ be the formula

$$\forall t\colon \{t\colon T \mid R(k,s,t)\}\,.\, a(k+1, s \oplus_k t) \in X(k+1, s \oplus_k t)$$

The (restricted) bar induction rule is:

$$\frac{\begin{array}{c} H \vdash T \in \text{Type} \qquad H,\ k\colon \mathbb{N},\ s\colon V_k(T),\ t\colon T \vdash R(k,s,t) \in \text{Type} \\ H,\ k\colon \mathbb{N}\,,s\colon V_k(T),\ con(R,k,s) \vdash B(k,s)\ \vee\ \neg B(k,s) \\ H,\ f\colon \mathbb{N} \to T,\ \forall i\colon \mathbb{N}.\, R(i,f,f(i)) \vdash\downarrow\exists n\colon \mathbb{N}.\, B(n,f) \\ H,\ k\colon \mathbb{N}\,,s\colon V_k(T),\ con(R,k,s),\ B(k,s) \vdash a(k,s) \in X(k,s) \\ H,\ k\colon \mathbb{N}\,,s\colon V_k(T),\ con(R,k,s),\ ind(R,T,a,X,k,s,t) \vdash a(k,s) \in X(k,s) \end{array}}{H \vdash a(0,z) \in X(0,z)}$$

The first two premises give the type of the spread law R. The next two premises state that B is a decidable bar on the spread defined by R. The fifth and sixth premises are the base and induction steps of the proof by bar induction for the term $a(0,z) \in X(0,z)$ in the conclusion of the rule. This is a strong form of bar induction because the spread law R can be any relation not necessarily decidable.

Since Nuprl allows general recursive definitions we can define bar recursion as

$$\text{br}(d,b,i,n,s) = \text{if } d{=}\text{inl}(x) \text{ then } b(n,s,x) \text{ else } i(n,s,\lambda t.\, \text{br}(d,b,i,n+1,s \oplus_n t))$$

and, using the restricted bar induction rule, we prove that bar recursion is the realizer for the general, unrestricted form of bar induction.

Remarks.

1. As described in a companion paper, all the non-inductive types can be built using only three type constructors, intersection, equality, and PER, which forms a type from a partial equivalence relation on closed terms.
2. Anand and Rahli have implemented Nuprl in Coq by defining its computation system, type system, sequents and rules. The type system they define has W types as primitives and does not include Mendler's recursive types. They have both an impredicative model of all the universes and a predicative model of finitely many.
3. Nuprl currently uses Mendler's recursive types, but every use of a recursive type in our library could be replaced with a W-type.
4. The results in this paper reduces the soundness of inductive constructions to the soundness of the bar induction rule given above. Because bar induction is true in classical logic, we should be able to prove it in the impredicative Coq model of Nuprl using the excluded middle axiom. This is work in progress.
5. We believe that analogues of Coq's inductive types can be defined using parameterized W-types because Nuprl's type theory satisfies function extensionality.
6. We do not know whether Agda's inductive-recursive constructions can be defined using the method of this paper (corecursion and bar induction).