A Calculus of Primitive Recursive Constructions

Hugo Herbelin and Ludovic Patey

Université Paris Diderot, Paris, France
Hugo.Herbelin@inria.fr
Ludovic.Patey@computablity.fr

Primitive Recursive Arithmetic (PRA) is a predicative formal subsystem of Peano Arithmetic, which aims to capture finitist reasoning. PRA is expressive enough to manipulate basic structures using codings to natural numbers. The tradeoff between its expressive power and the consensus on finitist reasoning makes it a good candidate as a meta-system in proof theory [3].

However the necessity to go through coding for manipulating any structure different from natural numbers makes proofs more tedious. Moreover, a wide class of formal structures can be easily proven to be encodable within PRA, so the restriction to natural numbers as the only primitive structure of the language is not justified. It becomes clear that proof theory would benefit from the introduction of a typed language enabling mathematicians to express in a natural way their objects manipulated. Such a language should be proven to be able to express only structures encodable in natural numbers within Primitive Recursive Arithmetic.

With the advent of computers, formal proof assistants have been developed [2, 5], answering to the request of more reliability in the proofs, enabling more collaboration in the research process and therefore to tackle bigger projects. Designed to capture the broadest audience, they provide very expressive formal languages to be able to mechanise in a natural way a wide range of proofs. This policy finds its limits in proof theory where mathematicians want to ensure that their proofs are formalisable in a weak formal system. A proof theorist may want to parametrise the expressiveness of his proof assistant, leading to a syntactical restriction of the formal language.

We design a Calculus of Primitive Recursive Constructions (CPRC) as a subsystem of the Calculus of Inductive Constructions without dependent product. In practice, in our approach of CPRC, inductive types are presented algebraically from the type constructors 0, 1, $+$, Σ, $=$ and μ, while recursion is decomposed into pattern-matching and guarded fixpoint as done in [4]. We provide two translations α and β between a logic-free presentation of PRA [1] and CPRC. We also provide proofs of their correctness in a sense defined below. Hence the CPRC has the same expressiveness as PRA, while being able to express natural structures in a simpler way.

Let Nat be the inductive type $\mu X.1 + X$ and $=_{Nat}$ be the equality over Nat.

$$t_1 = u_1, \ldots, t_n = u_n \vdash_{PRA} t = u$$

is a valid PRA judgement iff there exists a proof term p such that

$$x_1 : [\![t_1]\!]_\alpha =_{Nat} [\![u_1]\!]_\alpha, \ldots, x_n : [\![t_n]\!]_\alpha =_{Nat} [\![u_n]\!]_\alpha \vdash_{CPRC} p : [\![t]\!]_\alpha =_{Nat} [\![u]\!]_\alpha$$

is a valid CPRC judgement. Conversely, terms of CPRC are coded as terms in PRA whereas types are interpreted as characteristic functions of the coding of their inhabitants. If

$$x_1 : A_1 \ldots, x_n : A_n \vdash_{CPRC} p : A$$

is a valid CPRC judgement, then

$$[\![A_1]\!]_\beta\ x_1 = 0, \ldots, [\![A_n]\!]_\beta\ x_n = 0 \vdash_{PRA} [\![A]\!]_\beta\ [\![p]\!]_\beta = 0$$

is a valid PRA judgement. The conditions under which the converse holds are still under study.

A CPRC judgement is a refinement of a PRA judgement as it contains the proof terms, hence is more informative about the derivation tree. Note that in CPRC as in CIC, there is no distinction between formulae and types. In particular equality is a type.

This calculus is a first step to an implementation in Coq, opening the door to proofs within Primitive Recursive Arithmetic using formal proof assistants.

References

[1] Haskell B Curry. A formalization of recursive arithmetic. *American Journal of Mathematics*, pages 263–282, 1941.

[2] Gilles Dowek, Amy Felty, Hugo Herbelin, Gérard Huet, Benjamin Werner, Christine Paulin-Mohring, et al. The coq proof assistant user's guide: Version 5.6. 1991.

[3] Gerhard Gentzen. Die widerspruchsfreiheit der reinen zahlentheorie. *Mathematische Annalen*, 112(1):493–565, 1936.

[4] Hugo Herbelin and Arnaud Spiwack. The rooster and the syntactic bracket. *arXiv preprint arXiv:1309.5767*, 2013.

[5] Tobias Nipkow, Lawrence C Paulson, and Markus Wenzel. *Isabelle/HOL: a proof assistant for higher-order logic*, volume 2283. Springer, 2002.