▸ GIACOMO BARLUCCHI, BAHAREH AFSHARI, AND GRAHAM E. LEIGH,

Extracting computational content from cyclic proofs using higher order recursion schemes.

Gothenburg University, Göteborg , Sweden.
E-mail: giacomo.barlucchi@gu.se.
E-mail: bahareh.afshari@gu.se.
E-mail: graham.leigh@gu.se.

Presented here is a method to extract computational content from cyclic proofs by means of aptly defined higher order recursion schemes (HORS). The already existing technique ([2], [3]) is modified and adapted for a cyclic sequent calculus for intuitionistic arithmetic, with inductively defined quantifiers. Cut-elimination for such a cyclic system is shown via a strategy of continuous multicut reduction (see [7] and [4]). A notable property of these recursion systems is that their language is invariant with respect to the cut-reduction strategy. As a result, considerations about the order of the HORS and the properties of the sequents show that the computation terminates for proofs of Σ_1 end-sequents, producing the desired language.

§1. Preliminaries. A proof π of a formula φ bears in itself more information than the simple validity of φ. In the context of classical logic, for example, information of great interest are sets of terms satisfying Herbrand's Theorem. A direct way of computing a set of terms $t_0, \ldots, t_n$ consists in passing π through a process of cut-elimination – a costly procedure that might produce distinct cut-free proofs – resulting in different possible sets. Over time, attention has been directed to alternative ways of obtaining such sets. One fruitful way consists of exploiting the strict correspondence that exists between intuitionistic proofs and some known formalism: by adopting the BHK interpretation of constructive provability, a proof can be approached via a number of different tools. For example: among the first to go in that direction, Gerhardy and Kohlenbach showed a method to extract Herbrand's disjunctions from proofs using functional interpretation ([8]).

The idea of using recursion schemes to extract computational content from proofs was introduced by Hetzl in [9] and extended by Afshari, Hetzl and Leigh in [2] and then [3], where they present a method to extract the Herbrand set from proofs of a classical one-sided sequent calculus.

§2. Present work: intuitionistic cyclic arithmetic. In the present work a similar strategy is applied to a cyclic system for arithmetic. Cyclic proof systems have seen a surge in their importance in the recent decades (see e.g. [5] and [6]). The core idea is to allow for open assumptions as long as they a) generate a cycle with respect to some previous step of the proof, and b) some kind of regression along a well-founded order occurs, to ensure soundness.

The overall motivation is to test the method described above, i.e. extraction of computational content using HORS, against a system with potentially infinite proofs, and with inductive predicates. Recent work from Afshari, Enqvist and Leigh [1] have provided some answers from a general perspective in the classical case.

In the present study we approach the issue with a focus on a sequent-style calculus of intuitionistic arithmetic, called ICA. Its particular feature is that there are not primitive inductive predicates, as it is the case in [1] and in many systems ([6], [5], [10]). Here, first order quantifiers are responsible for (co–)inductive definitions: they are the least and greatest fixpoint of the functions induced by a predicate with free variables, over the natural numbers:

$$\forall x \geq k\ \varphi \equiv \varphi(k) \wedge \forall x \geq k'\ \varphi \qquad \exists x \geq k\ \varphi \equiv \varphi(k) \vee \exists x \geq k'\ \varphi$$

The corresponding rules in ICA are

$$\frac{\varphi(k), \Gamma \Rightarrow \chi \qquad \exists x \geq k'\ \varphi(x), \Gamma \Rightarrow \chi}{\exists x \geq k\ \varphi(x), \Gamma \Rightarrow \chi}\ (L\exists)$$

$$\frac{\Gamma \Rightarrow \varphi(k)}{\Gamma \Rightarrow \exists x \geq k\ \varphi(x)}\ (R\exists_0) \qquad \frac{\Gamma \Rightarrow \exists x \geq k'\ \varphi(x)}{\Gamma \Rightarrow \exists x \geq k\ \varphi(x)}\ (R\exists_1)$$

$$\frac{\varphi(k), \forall x \geq k'\ \varphi(x), \Gamma \Rightarrow \chi}{\forall x \geq k\ \varphi(x), \Gamma \Rightarrow \chi}\ (L\forall) \qquad \frac{\Gamma \Rightarrow \varphi(k) \qquad \Gamma \Rightarrow \forall x \geq k'\ \varphi(x)}{\Gamma \Rightarrow \forall x \geq k\ \varphi(x)}\ (R\forall)$$

The rest of the system consists of standard rules of intuitionistic sequent calculus, arithmetic axioms and explicit substitutions. A pre-proof π is a finite derivation tree obtained according to the rules, such that to each open assumption is associated an inner node with the same sequent. Call *progressing trace* a trace of formulas such that infinitely many of them are principal in a $(L\exists)$ or a $(R\forall)$ rule. A global condition on traces ensures soundness: a pre-proof π is a ICA proof if every infinite path in π contains a progressing trace.

This work presents a method to define a recursion scheme $\mathcal{H}^\pi$ from every proof π with end-sequent of the form $\Gamma \Rightarrow \varphi$. The sequent is interpreted in the usual BHK fashion as a function mapping inputs of 'type' Γ to an output of 'type' φ. The recursion scheme $\mathcal{H}^\pi$ specifies this function. The scheme is designed to put particular care in encoding individual terms and their manipulation along the unravelling of the proof, with in mind the ultimate goal of extracting witnesses for the end-formula φ. For any proof π with end-sequent $\Gamma \Rightarrow \varphi$, define the language $\mathcal{L}(\mathcal{H}^\pi)$ as the set of closed terms obtained computing $\mathcal{H}^\pi$ on every initial list of input of 'type' Γ.

Breaking down cyclic proofs results in general in an infinite process. A significant part of the work is dedicated to show termination of such a process over Σ_1 sequents, i.e., sequents of the form $\Gamma \Rightarrow \varphi$ where no existential (universal) formula appears positively in Γ (φ).

To that end, a proof of cut-elimination for ICA is given using a strategy of continuous multicut reduction, a method introduced in [7] and implemented for the logic μMALL in [4]. As a result, to every initial cyclic proof π corresponds a cut-free proof π^*, even though potentially a non well-founded proof. However, assuming that the end-sequent is Σ_1, the result of the multicut reduction is certain to produce a well-founded proof, and, hence, the rewrite process terminates.

Finally, it is proven that the language $\mathcal{L}(\mathcal{H}^\pi)$ is invariant under the multicut reduction strategy, so $\mathcal{L}(\mathcal{H}^\pi) = \mathcal{L}(\mathcal{H}^{\pi^*})$. That leads to the main result

Theorem. *For every* ICA *proof* π *of a* Σ_1 *closed sequent* $\Delta \Rightarrow \varphi$ *there exists a recursion scheme* $\mathcal{H}^\pi$ *such that the language* $\mathcal{L}(\mathcal{H}^\pi)$ *is a Herbrand set for* φ.

In the proposed talk I will give an overview of the project, introduce the system ICA and explain the operation of HORSs. Finally, I will show how the multicut elimination strategy works for ICA, and the correspondence with the production of $\mathcal{L}(\mathcal{H}^\pi)$.

[1] Bahareh Afshari, Sebastian Enqvist and Graham E. Leigh, *Herbrand schemes for cyclic proofs*, ***Journal of Logic and Computation***, vol. 35 (2025), no. 4, exaf013.

[2] Bahareh Afshari, Stefan Hetzl and Graham E. Leigh, *Herbrand disjunctions, cut elimination and context-free tree grammars*, ***13th International Conference on Typed Lambda Calculi and Applications (TLCA 2015). Leibniz International Proceedings in Informatics (LIPIcs)*** , (Thorsten Altenkirch editor), vol. 38, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015, pp. 1–16.

[3] ——— *Herbrand's theorem as higher order recursion*, ***Annals of Pure and Applied Logic***, vol. 171 (2020), no. 6, 102792.

[4] David Baelde, Amina Doumane and Alexis Saurin, *Infinitary proof theory: the multiplicative additive case*, ***25th EACSL Annual Conference on Computer Science Logic (CSL 2016). Leibniz International Proceedings in Informatics (LIPIcs)*** , (Jean-Marc Talbot and Laurent Regnier, editors), vol. 62, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016, pp. 42:1–42:17.

[5] James Brotherston, ***Sequent calculus proof systems for inductive definitions***, PhD Thesis, University of Edinburgh. College of Science and Engineering. School of Informatics, 2006.

[6] Stefano Berardi and Makoto Tatsuta, *Equivalence of intuitionistic inductive definitions and intuitionistic cyclic proofs under arithmetic*, ***CoRR***, abs/1712.03502 (2017).

[7] Jérôme Fortier and Luigi Santocanale, *Cuts for circular proofs: semantics and cut-elimination*, ***Computer Science Logic (CSL 2013). Leibniz International Proceedings in Informatics (LIPIcs)***, (Simona Ronchi Della Rocca, editor), vol. 23, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013, pp. 248–262.

[8] Philipp Gerhardy and Ulrich Kohlenbach, *Extracting Herbrand disjunctions by functional interpretation*, ***Archive for Mathematical Logic***, vol. 44 (2005), pp. 633–644.

[9] Stefan Hetzl, *Applying tree languages in proof theory*, ***Language and Automata Theory and Applications. LATA 2012. Lecture Notes in Computer Science*** , (Adrian-Horia Dediu and Carlos Martin-Vide, editors), vol. 7183, Springer Berlin Heidelberg, 2012, pp. 301–312.

[10] Alex Simpson, *Cyclic arithmetic is equivalent to Peano arithmetic*, ***Foundations of Software Science and Computation Structures. FoSSaCS 2017. Lecture Notes in Computer Science*** , (Javier Esparza and Andrzej S. Murawski, editors), vol. 10203, Springer Berlin Heidelberg, 2017, pp. 283–300.