

Building explicit induction schemas for cyclic induction reasoning

- extended abstract -

Sorin Stratulat

Université de Lorraine
LITA, Department of Computer Science
Ile du Saulcy, Metz, F-57000, FRANCE
`sorin.stratulat@univ-lorraine.fr`

Introduction. (Mathematical) induction reasoning is known to be one of the most effective reasoning methods. Its distinctive feature is the use, during the proof process, of *induction hypotheses* representing ‘not yet proved’ formulas.

The induction hypotheses can be defined before their use, by *explicit* induction schemas directly embedded in inference systems using explicit induction rules. On the other hand, the induction hypotheses can also be defined by need, at the moment of their use. In this case, the induction reasoning is schemata-free but should be performed at the proof level because the induction hypotheses can be (instances of) formulas positioned anywhere in the derivation.

The relation between the two kinds of induction reasoning was previously studied for Gentzen-style μ -calculus proof systems and proved *equivalent*, i.e., the proofs are convertible between systems. In [5], the explicit induction proofs are finite and the induction rules locally implement the Noetherian induction principle. The second kind of reasoning is captured by ω -regular proof trees that are finitely represented by stopping the development of the (sub-)derivations for nodes labelled with formulas already encountered in the derivation. The soundness of the method is ensured if the *cyclic* induction reasoning associated to each strongly connected component of the proof graph is well-founded.

Goals. We are interested to extrapolate the equivalence result for sequent-based systems that extend the Gentzen’s LK system [4] to reason on inductively defined predicates [1]. We focuss on two representative systems, proposed by Brotherston [2,3]: i) the LKID structural system that integrates induction rules generalizing Noetherian induction reasoning by the means of schemas issued from the recursion analysis of (mutually defined) inductive predicates, and ii) the CLKID $^\omega$ cyclic system, issued from LKID by replacing the induction rules with case-split rules. A main result of [2,3] is that any LKID proof is convertible into a CLKID $^\omega$ proof. However, the conversion in the other direction is only conjectured.

Results. We present a procedure that builds *dynamically* explicit induction schemas from the structural analysis of CLKID $^\omega$ proofs. Firstly, we show the equivalence between CLKID $^\omega$ and an *extension* of LKID, denoted by LKID $^\sim$, with induction rules that embed cyclic reasoning from existing CLKID $^\omega$ proofs.

The procedure is further refined to build Noetherian induction schemas from the productions defining new inductive predicate symbols attached to strongly connected components. LKID and CLKID^ω, extended to deal with the new symbols, are shown equivalent. We conclude that Noetherian induction is sufficient to represent the structural and cyclic induction reasoning in this setting. Aside the theoretical and technical interests, this result can help developing i) effective sequent-based induction methods that combine the advantages of the two kinds of reasoning, and ii) automated techniques for their validation, supported by (higher-order) certification environments integrating the Noetherian induction principle, like Coq [8].

In this way, we show that structural and cyclic induction reasoning in FOL_{ID} are convertible and are subsumed by Noetherian induction. Compared to LKID, LKID_e can generate proofs that combine the advantages of structural and cyclic reasoning by the dynamic definition of new induction schemas. Our procedure transforms effectively cyclic to structural proofs and does not require unfolding operations as in [5] or as suggested in [2,3].

Future work. These results open the perspective for validating CLKID^ω-like proofs by (higher-order) certification environments integrating the Noetherian induction principle, like Coq [8], by using similar certification methodologies as for implicit induction [7]. Also, it would be interesting to extend CLKID^ω for integrating reductive-free cycles [6]. More details can be found at <http://hal.archives-ouvertes.fr/docs/00/95/67/69/PDF/document.pdf>.

References

1. P. Aczel. An introduction to inductive definitions. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 739–782. North Holland, 1977.
2. J. Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *Proceedings of TABLEAUX-14*, volume 3702 of *LNAI*, pages 78–92. Springer-Verlag, 2005.
3. J. Brotherston. *Sequent Calculus Proof Systems for Inductive Definitions*. PhD thesis, University of Edinburgh, November 2006.
4. G. Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.
5. C. Sprenger and M. Dam. On the structure of inductive reasoning: Circular and tree-shaped proofs in the μ calculus. In A. Gordon, editor, *Foundations of Software Science and Computation Structures*, volume 2620 of *Lecture Notes in Computer Science*, pages 425–440. Springer Berlin / Heidelberg, 2003.
6. S. Stratulat. A unified view of induction reasoning for first-order logic. In A. Voronkov, editor, *Turing-100 (The Alan Turing Centenary Conference)*, volume 10 of *EPiC Series*, pages 326–352. EasyChair, 2012.
7. S. Stratulat and V. Demange. Automated certification of implicit induction proofs. In *CPP’2011 (First International Conference on Certified Programs and Proofs)*, volume 7086 of *Lecture Notes Computer Science*, pages 37–53. Springer Verlag, 2011.
8. The Coq development team. *The Coq Reference Manual - version 8.4*. INRIA, 2013.