Unrefutability by clause set cycles

Joint work with Stefan Hetzl

Jannik Vierling

Vienna University of Technology
Institute of Discrete Mathematics and Geometry

jannik.vierling@tuwien.ac.at

The subject of automated inductive theorem proving (AITP) aims at automating the process of finding proofs by mathematical induction. The automation of proof by mathematical induction has applications in formal methods for software engineering and in the formalization of mathematics. A great variety of methods has been developed for the automation of proof by induction. Typically each method operates in a more or less different setting. Furthermore, the design of these methods is driven by efficiency and ease of automation and therefore many AITP methods exist mainly at lower levels of abstraction. Because of their technical nature, AITP methods are traditionally analyzed empirically, and formal results backing the empirical observations are still rare. In particular, there are currently only very few negative results and it is difficult to classify AITP systems by their strength.

We address this situation by analyzing AITP methods formally. The first step of such an analysis consists in abstracting an AITP method, or a family of methods, by a logical theory. Such an abstraction can then be analyzed by applying results and techniques from mathematical logic. In this way we can measure the strength of AITP systems and compare them with each other. Furthermore, abstracting AITP systems by logical theories allows us to obtain negative results which are particularly valuable in revealing the logical features that a given method lacks.

The n-clause calculus [KP13] is an AITP method that extends the superposition calculus by a cycle detection mechanism. A cycle detected by the n-clause calculus represents an argument by infinite descent that establishes the inconsistency of the given clause set and thus terminates the refutation. In [HV20] we have analyzed the n-clause calculus by abstracting its comparatively technical cycle detection mechanism by the notion of clause set cycles. In the following we will recall the notion of refutation by a clause set cycle and some important results. By $0/0$ and $s/1$ we denote the function symbols representing the natural number 0 and the successor function for natural numbers, respectively. Furthermore, we fix a special, fresh constant symbol η on which arguments by infinite descent take place.

Definition 1. *Let L be a first-order language. An $L \cup \{\eta\}$ clause set $\mathcal{C}(\eta)$ is called an L clause set cycle if it satisfies the following conditions*

$$\mathcal{C}(s(\eta)) \models \mathcal{C}(\eta), \tag{C1}$$

$$\mathcal{C}(0) \models \bot. \tag{C2}$$

An $L \cup \{\eta\}$ clause set $\mathcal{D}(\eta)$ is refuted by a clause set cycle $\mathcal{C}(\eta)$ if

$$\mathcal{D}(\eta) \models C(\eta). \tag{C3}$$

By dualizing the definition of clause set cycles and observing that clause set cycles are essentially parameter-free we can we can show that refutations by a clause set cycle can be simulated by the parameter-free induction rule for $\exists_1$ formulas.

Theorem 2 ([HV])**.** *Let $\mathcal{D}(\eta)$ be an $L \cup \{\eta\}$ clause set. If $\mathcal{D}(\eta)$ is refuted by an L clause set cycle, then $[\varnothing, \exists_1(L)^{-}\text{-IND}^{R}] + \mathcal{D}(\eta)$ is inconsistent.*

As mentioned above this upper bound is optimal in terms of the quantifier complexity of the induction formulas. In other words clause set cycles cannot be simulated by quantifier-free induction.

Theorem 3 ([HV20])**.** *There exists a language L and an $L \cup \{\eta\}$ clause set $\mathcal{D}(\eta)$ such that $\mathcal{D}(\eta)$ is refuted by an L clause set cycle, but* $\mathrm{Open}(L)\text{-IND} + \mathcal{D}(\eta)$ *is consistent.*

These results give rise to the question whether clause set cycles are at least as strong as induction for quantifier-free formulas. Empirical evidence has led us to conjecture that refutation by a clause set cycle is incomparable with induction for quantifier-free formulas.

In this talk we will show that this conjecture has a positive answer. We define a candidate clause set in the setting of linear arithmetic. The language of linear arithmetic consists of the symbols $0/0$, $s/1$, $p/1$, and $+/2$, where the latter two represent the predecessor function and the addition of natural numbers, respectively. Let $\mathcal{T}$ be the theory axiomatized by the universal closure of $0 \neq s(x)$ and the defining equations of $p/1$ and $+/2$, then the clause set $\mathcal{I}(\eta)$ is given by

$$\mathcal{I}(\eta) \coloneqq cnf(\mathcal{T}) \cup \{\{\eta + \eta = \eta\}, \{\eta \neq 0\}\}.$$

Intuitively, the clause set $\mathcal{I}(\eta)$ asserts the existence of a non-zero additive idempotent. By making use of the upper bound of Theorem 2 we can show the unrefutability of $\mathcal{I}(\eta)$ by a clause set cycle by proving the following independence result.

Theorem 4. $[\mathcal{T}, \exists_1(L(\mathcal{T}))^{-}\text{-IND}^{R}] \nvdash x + x = x \rightarrow x = 0$.

We will proceed by constructing a model M with non-zero idempotents whose domain consists of one copy of $\mathbb{N}$ and $|\mathbb{N}|$ copies of $\mathbb{Z}$. In particular, we will show that for every true, p-free, $\exists_1$ formula $\varphi(x)$, there exists on every non-standard chain an infinite, strictly descending sequence of elements $(z_i)_{i \in \mathbb{N}}$ such that

$$M \models \varphi(z_i), \text{ for all } i \in \mathbb{N}.$$

The unrefutability of $\mathcal{I}(\eta)$ by clause set cycles shows that clause set cycles are very weak and can not even deal with formulas such as $x+x = x \rightarrow x = 0$ that have a straightforward proof by quantifier-free induction. However, the situation may even be much worse. A clause set cycle $\mathcal{C}(\eta)$ corresponds roughly speaking to an inductive $\exists_1$ lemma $\varphi_{\mathcal{C}}(x)$. However, the notion of refutation by a clause set cycle only uses this lemma to infer the instance $\varphi_{\mathcal{C}}(\eta)$. Therefore, we conjecture that refutation by a clause set cycle is even incomparable with parameter-free induction for quantifier-free formulas. The intuition for this is, that proving a sentence like $0 + (\eta + \eta) = \eta + \eta$ requires the lemma $0 + x = x$ and the instance $x \mapsto \eta + \eta$. The conjectured relations between the refutational strength of clause set cycles and some related theories with induction are shown in Figure 1.

References

[HV] Stefan Hetzl and Jannik Vierling. An unprovability result for clause set cycles. In preparation.

[HV20] Stefan Hetzl and Jannik Vierling. Clause Set Cycles and Induction. *Logical Methods in Computer Science*, 16(4):11:1–11:17, November 2020.

[KP13] Abdelkader Kersani and Nicolas Peltier. Combining superposition and induction: A practical realization. In Pascal Fontaine, Christophe Ringeissen, and Renate A. Schmidt, editors, *Frontiers of Combining Systems*, volume 8152 of *Lecture Notes in Computer Science*, pages 7–22. Springer, 2013.

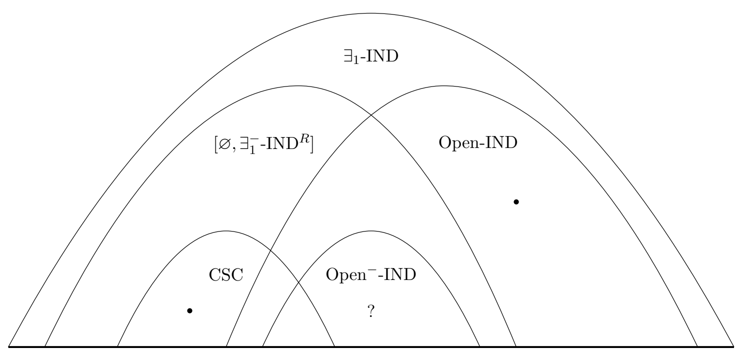

Figure 1: Conjectured relation between the refutational strength of various induction systems. The dots indicate that the surrounding area is not empty, the question mark indicates that we conjecture the are to be non-empty.