Automatic Construction of Implicative Theories for Mathematical Domains

Summary

Artem Revenko

Introduction

According to Wikipedia, logic can be considered as "the use and study of valid reasoning". In mathematics and computer sciences one is usually interested in *formal* logic, i.e. "the study of inference with purely formal content. An inference possesses a purely formal content if it can be expressed as a particular application of a wholly abstract rule, that is, a rule that is not about any particular thing or property." The *formality* of the content under logical consideration allows one for examining the reasoning without references to particular entities. Hence, formal logic offers a uniform approach to reasoning in all possible domains.

An attempt to formalize a domain and bring it to logical discourse leads to an idea of *logical theories*, i.e. a set of axioms describing the domain. We can classify logical theories with respect to the expressive power of the logical language used to construct the theory. Although very expressive languages are being successfully investigated, and modern computer system are able to arguably efficiently reason using very expressive languages, the composition of queries to the theories and the interpretation of the results of reasoning are left to humans. Thereby, the most intuitive and easy to understand axioms are of a great value. However, the most simple theory is an empty theory. Hence, a good trade-off between *expressiveness* and *complexity* of a language should be kept. Arguably the most useful theories should mimic the way people organize knowledge in their minds. With this in mind we focus on logical theories containing only *implications*, i.e. logical connectives corresponding to the rule of causality "if ... then ...". We call these theories *implicative theories*. Such theories not only correspond to our requirements, but also have certain theoretical advantages (e.g. polynomial complexity of standard reasoning tasks).

Another advantage of implicative theories is that they can be constructed from data – collection of facts about particular entities from a domain. However, if the data is not complete, the obtained theory may contain invalid implication. Namely, some implications may be violated by further facts. Entities violating the implications are called *counter-examples*. This suggests an idea of constructing implicative theories through refinement of existing theories via adding new data.

Moreover, any implicative theory may be compactly represented through its *(implication) basis*. All the valid implications of the implicative theory are logical consequences of the basis. In [GD86] a minimal in cardinality implication basis was introduced (the *canonical basis*). Therefore, it is possible to use this basis representation and look for counter-examples to the implications from the basis. The iterative procedure of computing an implication basis from examples, looking for a counter-example, and refining the data with a newly found counter-example is called *Attribute Exploration* (AE) [GW99].

The mathematical domains are probably the most suitable domains for performing AE, because the mathematical statements, if decidable, are either true or false. Moreover, we may try to generate the desired counter-examples algorithmically.

Structure of Current Study

Formal Concept Analysis (FCA) [GW99] is extensively used in the current investigation as the main toolbox for derivation, analysis, and representation of knowledge. Moreover, developed in the frames of the current study methods add to the methodology of FCA. We represent the data in form of a triple (G, M, I) called a *formal context*; the set G is called the set of *object*; M – the set of *attributes*; I – (binary) relation between objects and attributes.

The study contains a theoretical introduction (Chapter 1) and descriptions of two investigations about automation of the explorations of knowledge in two domains: parametric expressibility of logical functions and interrelations between algebraic identities.

The domains of application of AE are chosen such that the respective implicative theory is of an interest for experts from respective domain. The *goal of both investigations* is to explore the implicative theory of the domain, i.e. find all valid implications (in basis form) over selected attributes.

The present study is in a large part based on two articles [Rev14, Rev15]. Further concepts and ideas are borrowed from the articles [KR15, RK12, Rev13]. The main results of the articles are embedded into this study, which represents them coherently in a more wholistic way. The results of the current study were communicated at international conferences (ICFCA 2014, submitted to CLA 2015), at international workshops (AAA 89, EPCL Workshop 2013), and at research seminars (Institut für Algebra TU Dresden, Knowledge Systems group TU Dresden, Theory and Logic group TU Wien, School of Data Analysis and Artificial Intelligence HSE Moscow).

The program code written for the execution of the current investigations is stored at `https://github.com/artreven`.

Chapter 2: Lattice of P-Clones

The expressibility of functions is a major topic in mathematics and has a long history of investigation. The interest is explainable: when one aims at investigating any kind of functional properties, which classes of functions should one consider? If a function f is expressible through a function h then it often means that f inherits properties of h and should not be treated separately. Moreover, if h in turn is expressible through f then both have similar or even the same properties. Therefore, partition with respect to expressibility is meaningful and can be the first step in the investigation of functions.

With the development of electronics and logical circuits a new question arises: if one wants to be able to express one possible function which minimal set of functions should one have at hands? One of the first investigations in this direction was carried out in [Pos42]. There all the Boolean classes of functions closed under expressibility are found and described. Afterwards many important works were dedicated to related problems such as the investigation of the structure of the lattice of functional classes, for example, [Yab60]. However, it is known that the lattice of classes of functions closed under expressibility is in general uncountably infinite. In [Kuz79] a more general type of functional expressibility was introduced – parametric expressibility. A significant advantage of this type of expressibility is that for any finite domain A_k (k is the size of domain) the lattice of all classes closed under parametric expressibility classes of functions (p-clones) is finite [BW87]. However, finding this lattice is an extremely complex task. For $k = 2$ the lattice of p-clones was known. For $k = 3$ in a thorough and tedious investigation [Dan77] it was proved that a system of 197 functions forms the lattice of all p-clones.

In this chapter we are interested in "minimal" (with respect to parametric expressibility) logical functions. Namely, we are busy with the question "which functions do we need to add to the context in order to be able to reconstruct the lattice of p-clones?" We introduce, develop, and investigate the methods and tools for automation of the exploration of the lattice of p-clones. In this case objects and attributes are the same functions and the role of I plays the commutation between functions. The standard AE is not suitable for such investigation, therefore, an extension is introduced and investigated. Therefore, this chapter "applied" to A_3 can be seen as complementing the work [Dan77] where a proof of the correctness of the results obtained using the elaborated in this chapter tools can be found. Namely, in this chapter we answer the question **how** to find all the p-clones, whereas in [Dan77] it is proved that certain functions allow us to construct the desired lattice. The presented methods and tools are extendable to larger domains as well.

In this chapter we face the task of violating implicative constraints with **finite** counter-examples to implications over functions. We introduce two competing strategies to finding counter-examples and investigate and compare the resulting algorithms.

The first strategy is to start from the premise of the implication ($H \to j$), find functions satisfying the premise H, and afterwards find those of them that do not satisfy the conclusion j. In order to satisfy H a function f has to commute with all functions from H.

The second strategy is to start from the conclusion of the implication ($H \to j$), find functions violating the conclusion j, and afterwards find those of them that satisfy the premise H. In order to violate the conclusion, i.e. $f \not\perp j$, there has to exists a matrix M such that $f(Mj) \neq (fM)j$.

It is shown that the algorithms explore search space in different manners and, therefore, may arrive at different solutions. Moreover, it is shown that none of the algorithms uniformly outperforms the other. Therefore, the best results are obtained when using both algorithms simultaneously.

Further on the necessary extension of AE is introduced and discussed. The extension is particularly suitable for discovering p-indecomposable functions. The resulting procedure is further investigated. According to the proved propositions it is necessary to look for not more than two (second-order irreducible) functions at once in order to find all p-indecomposable functions. This task is infeasible as there exist too many functions. Fortunately, it is feasible to find not only counter-examples to implications over functions, but also find single (first-order irreducible) functions that are not counter-examples, but alter the concept lattice. Moreover, if it would be possible to prove that the undiscovered so far functions are not second-order irreducible then we can guarantee that all the p-indecomposable functions will eventually be discovered. We show that if we start from all unary functions on A_3 all the p-indecomposable functions on A_3 will eventually be discovered.

At the end of the chapter the results are discussed. The elaborated methods and tools allow for successful completion of the exploration. The experiment on A_3 was run. All 197 p-indecomposable functions and 2986 p-clones were successfully found.

Contributions

- New original approach to exploring the lattice of p-clones introduced;
- Two approaches to finding finite counter-examples are introduced;
- Corresponding algorithms are described, compared, implemented;
- An extension of the standard exploration procedure is introduced and investigated;
- The whole procedure is implemented and executed; the obtained results confirm with the previously known results;
- It is proved that for certain starting conditions the desired lattice will necessarily be eventually discovered.

Chapter 3: Implicative Theory of Algebraic Identities

Algebraic identities describe different classes of algebraic structures (equational classes) and therefore play one of the central roles in algebra. The field of research that studies common patterns of algebraic structures is called universal algebra [Bir35]. As noted in [Tay79]: "The role of algebraic equations was pronounced from the start". The studying of equational classes is essentially important for mathematics.

A central question one could ask about equational classes is the following: if the class satisfies a given set of identities which other identities are necessarily satisfied by all the members of the class? The strength and importance of equational deduction can be well appreciated from the words from [CT51]: "it has even been shown that every problem concerning the derivability of a mathematical statement from a given set of axioms can be reduced to the problem of whether an equation is identically satisfied in every relation algebra. One could thus say that, in principle, the whole of mathematical research can be carried out by studying identities in the arithmetic of relation algebras." It is well known that in general it is not possible to decide if an identity is deducible from a given set of identities, see e.g. [Tar41]. Even for a finite set of equations this question can be undecidable [Tay79, p. 28]. However, there are special classes of identities for which the questions is decidable, for example, groups [Deh11]. The modern field of science called automated theorem proving has made a big progress in equational deduction (as a part of deduction in first order logic). To be more precise equational deduction is semidecidable, meaning that it is not always possible to say if the answer is negative, i.e. when an identity does not hold. As a counterpart of automatic theorem provers, automatic model finders are also actively developed. However, modern tools concentrate on finite models.

Deductibility is not at all the only question of interest about equational classes. As pointed out in [BS81, Recent Developments and Open Problems] finding (finite) bases for equational theories and classification of equational classes are in scope of current research activities. For the purpose of solving these two questions in a given set of identities one could find all possible interrelations between identities inside this set (implicative theory of identities). Up to now no automated knowledge processing algorithm was offered to automatize the research of the implicative theory of a given set of identities.

Automation of usage of AE for the exploration of identities and making it efficient issues a number of unique challenges. For example, though only 70 identities of size up to 5 are under investigation, it turns out that it is not possible to finish the investigation considering only finite counter-examples. The structure of possible infinite counter-examples is investigated and, based on this investigation, a method for finding these counter-examples is introduced and implemented.

In this chapter we consider the context of algebras and identities. In particular, we describe an algorithm for checking the satisfaction of identities in finite algebras and an algorithm for finding non-equivalent identities of a given size. The most significant results of this chapter consist in introducing and proving the criteria of the necessity of infinite counter-examples, investigating the structure of infinite counter-examples, and introducing an algorithm for finding them.

We identify that infinite counter-examples are necessary and described classes of implications that have only infinite counter-examples. We investigate possibilities of finding these infinite counter-examples. This investigation starts from the analysis of their structure. We inspect the universe of an infinite counter-example and introduce an infinite subuniverse corresponding to a particular term operation. We prove that it suffices to consider only infinite algebras such that the only infinite subuniverse is the introduced one. Afterwards we introduce a computational model and an algorithm for finding the infinite counter-example with the discovered universe. The algorithm is capable of finding all the needed infinite counter-examples.

In AE of algebraic identities two methods for finding counter-examples are used: finite counter-examples are found using `Mace4` [McC10], infinite counter-examples are found using implementation of the introduced algorithm. Moreover, before finding counter-examples the program `Prover9` [McC10] makes an attempt to prove implications. The exploration is successfully finished. The final context after reduction contains 626 finite algebras and 1529 infinite algebras (altogether 2155 algebras). All 4398 unit implications from the canonical basis are proved.

Contributions

- An algorithm for finding non-equivalent identities is introduced and implemented;
- The conditions of the necessity of infinite counter-examples are introduced and proved;
- The structure of infinite counter-examples is investigated;
- A computational model and an algorithm for generating infinite algebras satisfying a set of identities and not satisfying a given identity is developed and implemented;
- The results of a successful exploration are discussed.

Conclusion

In the frames of the current project general approaches to automatic constructions of implicative theories for mathematical domains are investigated on two applications. The methodology of investigation is based on discovering knowledge from (counter-)examples – the procedure of Attribute Exploration. The relevant procedures are analyzed and implemented. The implementation is independent of the domain of application, hence, may be used for further explorations.

In both applications we succeeded in automatizing the process of exploration of the implicative theories. This goal is achieved thanks both pragmatical approach of Attribute Exploration and discoveries in respective domains of application. The structures of the attributes in the investigated domain is the key to developing efficient methods of finding counter-examples – the core of Attribute Exploration.

In the current study the methods and algorithm for finding both finite and infinite counter-examples are investigated and developed. In the case of infinite counter-examples the preliminary knowledge about the structure of the desired solution is essential for finding the counter-examples. In the case of finite models it is necessary to develop competitive algorithms implementing different computational strategies. The parallel run of the algorithms allows for finding counter-examples in limited time.

In case of p-indecomposable functions it is necessary to extend the procedure of Attribute Exploration and to investigate the extended version. It turned out that in case of growing number of attributes it is difficult to state any assumptions about the overall success of the exploration. However, we succeed in the exploration of p-indecomposable functions on three-valued domain, and the developed tools and methods may be used for an exploration on an even larger domains.

The two diverse application domains favourably illustrate different possible usage patterns of Attribute Exploration – in one case the number of attributes is fixed, however, counter-examples are infinite, in the other case the number of attributes grow, but the counter-examples are finite. The elaborated approaches may be further developed and used not only for constructing the complete implicative theories, but also for a more widespread problem of finding counter-examples to certain implications.

The choice of mathematical domains as domains of application is justified by the fact that the considered in the project mathematical statements are either true or false. Moreover, it is possible to generate the desired counter-examples algorithmically. The real life is usually more complex, the truth of many statements is argued, in order to find a counter-example it is necessary to attract experts of the respective domain. The current investigation (probably, together with methods for finding mistakes in object intents [KR15, Rev13]) may be seen as the first step to elaborating a system for organizing the knowledge and assisting experts from a domain in working with this knowledge.

Bibliography

[Bir35] G. Birkhoff. On the structure of abstract algebras. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 31, pages 433–454. Cambridge Univ Press, 1935.

[BS81] S. Burris and H.P. Sankappanavar. *A course in universal algebra*, volume 78. Springer-Verlag New York, 1981.

[BW87] S. Burris and R. Willard. Finitely many primitive positive clones. *Proceedings of the American Mathematical Society*, 101(3):427–430, 1987.

[CT51] L.H. Chin and A. Tarski. *Distributive and modular laws in the arithmetic of relation algebras*, volume 1. University of California Press, 1951.

[Dan77] A.F. Danil'chenko. Parametric expressibility of functions of three-valued logic. *Algebra and Logic*, 16(4):266–280, 1977.

[Deh11] M. Dehn. Über unendliche diskontinuierliche gruppen. *Mathematische Annalen*, 71(1):116–144, 1911.

[GD86] J.-L. Guigues and V. Duquenne. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Math. Sci. Hum*, 24(95):5–18, 1986.

[GW99] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.

[KR15] S.O. Kuznetsov and A. Revenko. Interactive error correction in implicative theories. *International Journal of Approximate Reasoning*, submitted 2015.

[Kuz79] A.V. Kuznetsov. Means for detection of nondeducibility and inexpressibility. *Logical Inference*, pages 5–33, 1979.

[McC10] W. McCune. Prover9 and mace4. `http://www.cs.unm.edu/~mccune/prover9/`, 2005–2010.

[Pos42] E.L. Post. *The two-valued iterative systems of mathematical logic*. Princeton University Press, 1942.

[Rev13] A. Revenko. Detecting mistakes in binary data tables. *Automatic Documentation and Mathematical Linguistics*, 47(3):102–110, 2013.

[Rev14] A. Revenko. Automatized construction of implicative theory of algebraic identities of size up to 5. In Cynthia Vera Glodeanu, Mehdi Kaytoue, and Christian Sacarea, editors, *Formal Concept Analysis*, volume 8478 of *Lecture Notes in Computer Science*, pages 188–202. Springer International Publishing, 2014.

[Rev15] A. Revenko. Finding p-indecomposable function: Fca approach. In *CLA 2015*, submitted 2015.

[RK12] A. Revenko and S.O. Kuznetsov. Attribute exploration of properties of functions on sets. *Fundamenta Informaticae*, 115(4):377–394, 2012.

[Tar41] A. Tarski. On the calculus of relations. *The Journal of Symbolic Logic*, 6(3):73–89, 1941.

[Tay79] W. Taylor. *Equational logic*. University of Houston, Department of Mathematics, 1979.

[Yab60] S.V. Yablonsky. *Functional Constructions in K-valued Logic*. U.S. Joint Publications Research Service, 1960.