axioms

Article

Contribution of Warsaw Logicians to Computational Logic

Damian Niwiński

Institute of Informatics, University of Warsaw, 02-097 Warsaw, Poland; niwinski@mimuw.edu.pl;
Tel.: +48-22-554-4460

Academic Editor: Urszula Wybraniec-Skardowska
Received: 22 April 2016; Accepted: 31 May 2016; Published: 3 June 2016

Abstract: The newly emerging branch of research of Computer Science received encouragement from the successors of the Warsaw mathematical school: Kuratowski, Mazur, Mostowski, Grzegorczyk, and Rasiowa. Rasiowa realized very early that the spectrum of computer programs should be incorporated into the realm of mathematical logic in order to make a rigorous treatment of program correctness. This gave rise to the concept of algorithmic logic developed since the 1970s by Rasiowa, Salwicki, Mirkowska, and their followers. Together with Pratt's dynamic logic, algorithmic logic evolved into a mainstream branch of research: logic of programs. In the late 1980s, Warsaw logicians Tiuryn and Urzyczyn categorized various logics of programs, depending on the class of programs involved. Quite unexpectedly, they discovered that some persistent open questions about the expressive power of logics are equivalent to famous open problems in complexity theory. This, along with parallel discoveries by Harel, Immerman and Vardi, contributed to the creation of an important area of theoretical computer science: descriptive complexity. By that time, the modal μ-calculus was recognized as a sort of a universal logic of programs. The mid 1990s saw a landmark result by Walukiewicz, who showed completeness of a natural axiomatization for the μ-calculus proposed by Kozen. The difficult proof of this result, based on automata theory, opened a path to further investigations. Later, Bojanczyk opened a new chapter by introducing an unboundedness quantifier, which allowed for expressing some quantitative properties of programs. Yet another topic, linking the past with the future, is the subject of automata founded in the Fraenkel-Mostowski set theory. The studies on intuitionism found their continuation in the studies of Curry-Howard isomorphism. Łukasiewicz's landmark idea of many-valued logic found its continuation in various approaches to incompleteness and uncertainty.

Keywords: algorithmic logic; dynamic logic; μ-calculus; λ-calculus

1. Introduction

The first universal digital computer made in Poland was called *XYZ* and constructed as early as in 1958 at Śniadeckich street 8 in Warsaw—an address well known nowadays to mathematicians visiting Warsaw, as it is the site of the International Stefan Banach Mathematical Center. In spite of limitations in contacts between Poland and rest of the world at that time, the ideas of computing machines attracted Polish engineers and scientists since the 1950s. It took some time, however, to realize that computing machines not only make a revolutionary tool to support cryptanalysts, physicists, geologists, engineers *etc.*, but they should give rise to a new academic discipline, computer science, which by the way in Polish has been called *informatyka* (informatics, *cf.* French *informatique*), thus highlighting information rather than physical machine itself.

At the early stage, the newly emerging branch of research received encouragement from leading mathematicians such as Kazimierz Kuratowski, Stanisław Mazur, Andrzej Mostowski, Andrzej Grzegorczyk, Helena Rasiowa. A foundational perspective on mathematics being characteristic for the Warsaw circle at that time, the connection with mathematical logic was soon established. Thus started a new "offspring" of the Warsaw logical school, which has been progressing through the next decades: Logic in Computer Science.

In this survey we review some ideas and results of the Warsaw group of logic in computer science, which the author finds particularly original and far-reaching. The selection is subjective and by no means exhaustive. Relation to the heritage of the Lvov–Warsaw school is taken into consideration. The important aspect of practical applications is not pursued in this survey. The author does not attempt to give complete pictures of research teams or individual researchers. The emphasis is on some particular results and ideas, and the references may guide an interested reader further.

2. Computability

Andrzej Grzegorczyk's paper *Some classes of recursive functions* [1] published in 1953, is often considered as an anticipation of computational complexity theory. Grzegorczyk organized the class of primitive recursive functions into an elegant hierarchy of classes $\mathcal{E}_n$. Roughly, the functions on the $n+1$-th level are obtained by iterating the functions from the previous level the number of times indicated by one of the arguments, and by the closure under primitive recursion scheme bounded from above by an already defined function. In particular, the class $\mathcal{E}_3$ captures the Kalmar elementary functions. The Grzegorczyk hierarchy, originally recursion-theoretic, is now a classic concept in computability theory.

Somewhat less remembered but worth to mention is a visionary work of Stanisław Mazur on computable analysis. This mathematician, whose primary fields of interest were nonlinear functional analysis and Banach algebras, elaborated the concept of computable real numbers and functions with Stefan Banach in 1936–39, but the results remained unpublished except for a short abstract [2]. After the war, Mazur alone attempted to extend his work to general computable mathematical objects. The results were assembled and published only in 1963 [3] (edited by A.Grzegorczyk and H.Rasiowa), and had some influence on the field of computable analysis, which was developed later by several researchers (in particular, Klaus Weihrauch). It should be mentioned, however, that in meantime the subject of computable analysis was approached by other logicians, in particular Ernst Specker [4].

There is no evidence that Alfred Tarski had much interest in computing machines, but his ideas greatly influenced computer science in many aspects; we refer the reader to an essay by Solomon Feferman on this subject [5]. The issue of algorithmic decidability of formalized mathematical theories, a persistent theme in computer science logic, can be traced back to the fundamental result by Tarski obtained around 1930: the decidability of the first-order theory of real numbers with addition and multiplication [6]. Mojżesz Presburger, a master student of Tarski solved another problem: the decidability of the theory of integers with addition only [7]. The importance of this result was recognized 50 years later, in context of computer science rather than general mathematics. Indeed, Presburger's arithmetic served as a crucial step in solutions of difficult algorithmic problems, in particular the reachability question in Petri nets. Presburger, who tragically disappeared in the Holocaust, gave his name to the annual award conferred since 2010 by the European Association for Theoretical Computer Science to a young scientist for outstanding contributions in theoretical computer science.

3. Logic of Algorithms

Helena Rasiowa made an impact on mathematical logic by her celebrated book, co-authored by Roman Sikorski and published in the same year 1963, *The Mathematics of Metamathematics* [8]. The authors undertake the program of investigating mathematical theories as mathematical objects, which can be traced back to Hilbert, but, as they declare in introduction, the *finitistic approach of Hilbert's school is completely abandoned* in their book. Instead, they promote the *use of the more advanced infinitistic methods*, in particular of algebra and topology, which *makes it easier to explain the mathematical structure of formalized theories*. As a remarkable example, the Gödel completeness theorem for the predicate calculus is obtained as a consequence of the Baire category theorem for topological spaces. This approach also shapes the way the intuitionistic logic is treated in the book. The authors put aside the philosophical motivation of Brouwer, and focus on the properties of algebraic structures that are needed to establish the completeness theorem for intuitionistic logic. Besides, Brouwer's logic appears as one among other non-classical logics, in particular its usefulness for the working mathematician is doubtful, but which nevertheless constitute an interesting object of mathematical studies. (We will come back to this point in Section 6).

While this approach to intuitionism may not satisfy some philosophers, it proved to be extremely fruitful when applied to the challenges of computer science, as in particular the need of a rigorous definition of semantics and correctness of computer programs. Since what else is a program if not a formal expression, whose semantics can be given as a function over the set of states, and whose correctness can be eventually formally proven? This idea underlines the concept of *algorithmic logic*, which incorporates programs into the language of predicate logic via a modal construction $[\mathit{program}]\ \mathit{formula}$, expressing the property that the computation of the *program* terminates in a state satisfying the *formula*. The use of modal construction has been an alternative to a previous pioneering work of Erwin Engeler in Zurich, who incorporated programs in logic using infinitary formulas [9]. The algorithmic logic was introduced by Andrzej Salwicki [10], at that time Rasiowa's student, whereas another student of hers, Grażyna Mirkowska [11] established an infinitary completeness theorem for the logic. Further development due to these authors, Antoni Kreczmar [12], Rasiowa herself [13], and others, has been summarized in the monograph [14], where the use of the algorithmic logic in education of programmers is also advocated. The class of programs in consideration is constructed from substitutions and tests by means of the *if then else* and *while* operators, but it is gradually extended throughout the book, by more advanced data structures like stacks and queues, nondeterminism, and concurrency. It is worth to mention that the logic has been helpful in the design of LOGLAN, an original programming language constructed and implemented at the University of Warsaw by a team led by Salwicki in 1977-1982 (see [15]).

4. Dynamic Logic Meets Complexity Theory

Several other approaches incorporating programs into logic have been proposed in the literature since the 1960s. The dynamic logic of Pratt [16] (see also [17]) is similar to algorithmic logic, except that the class of atomic programs in consideration is an abstract parameter of the setting. The structured programs are constructed from the atomic ones by means of regular expressions. Note that if two classes of atomic programs are equivalent, say $\mathcal{C} \equiv \mathcal{C}'$, then the induced dynamic logics are equivalent as well, *i.e.*, $\mathrm{DL}(\mathcal{C}) \equiv \mathrm{DL}(\mathcal{C}')$, but the reverse implication does not need to hold. Hence, dynamic logic gives us a subtle tool to compare various features of programs; for example we can compare the power of non-deterministic *vs.* deterministic programs in this way. Jerzy Tiuryn with his collaborators at the University of Warsaw ran an extensive research project in the 1980s, aiming to classify the expressive power of the logics $\mathrm{DL}(\mathcal{C})$, depending on the type of atomic programs in $\mathcal{C}$. The investigations progressed successfully, but some problems proved to be unexpectedly resistant, as in particular the comparison of programs with (algebraic) stacks *vs.* programs with arrays. It was not incidental, though. In 1983, Tiuryn and Paweł Urzyczyn [18] made the surprising discovery that equivalence of the dynamic logics over these two classes of programs amounts to equality of the complexity classes P and PSPACE, which is one of the famous open problems in complexity theory (the equality would imply a positive answer to the Millenium problem *P=NP?*). This discovery was possible thanks to a new technique invented by the authors, namely the spectral method. The idea is that the class of those families of *finite* models, which can be defined by sentences φ of a logic (*i.e.*, definable as $\{\mathbb{M} : \mathbb{M} \text{ is finite and } \mathbb{M} \models \varphi\}$), can be, via a suitable encoding, essentially identified with a complexity class. Similar phenomena have been discovered about the same time in the context of databases, by researchers investigating the expressive power of various query languages. In particular, Moshe Vardi and independently Neil Immerman discovered in 1982 that inductive queries "capture" in this way the class P (over ordered structures). These discoveries gave birth to a new branch of logic in computer science, called finite model theory, or *descriptive complexity*, which approaches the complexity theoretic questions via finite interpretations (see, e.g., [19]).

5. Completeness Theorem for the μ-Calculus

Another perspective on logic of programs was adopted by Amir Pnueli in his proposal of a Temporal logic of programs in the late 1970s. The primary difference with previous approaches concerns the role of a model. Formerly, it represented a computation domain, so that programs could be viewed as "generalized terms" acting on its elements. In temporal logic, one focuses on a single program and the model represents all possible computations of the program. This structure can be quite rich, especially if the program is concurrent, non-deterministic, reactive, stochastic, *etc*. The approach was followed by many researchers and has become dominating in program verification since the 1990s.

The modal μ-calculus $L\mu$ reconciles the two approaches. It subsumes propositional versions of dynamic and algorithmic logics, as well as most of (propositional) temporal logics, including LTL, CTL, CTL*. The syntax of $L\mu$ extends the syntax of propositional modal logic by the least and greatest fixed point operators μ and ν. A formula $\mu X.\varphi(X)$ is interpreted in a Kripke model as a *least* set of states V, such that whenever we interpret the formula $\varphi(X)$ in this model with X interpreted as V, we get the set V again; similarly for $\nu X.\varphi(X)$ and greatest fixed point. For example, a formula $\nu X.\Diamond X$ defines the set of states, from which there exists an infinite path, whereas the formula $\mu X.\nu Y.ox\,((p \to X) \wedge (\neg p \to Y))$ defines the property that, on each infinite path, p occurs only finitely often. The above modal version was proposed by Dexter Kozen in 1982 [20], building on numerous previous works by Scott and de Bakker, Moschovakis, Hitchcock and Park, and others. Later the present author observed [21] that, whenever interpreted in the binary tree, the μ-calculus is as expressive as monadic second order logic, whose decidability was established in the celebrated Rabin Tree Theorem [22].

In his introductory paper of 1982, Kozen proposed a finitary axiom system for $L\mu$, based on the standard rules for modal logic, as well as an axiom and rule, which actually define the least fixed point

$$\varphi[X/\mu X.\varphi] \to \mu X.\varphi \qquad \frac{\varphi[X/\psi] \to \psi}{\mu X.\varphi \to \psi}$$

It had been an open problem for more than decade, whether the system was complete, or whether there might exist any other complete finitary system for $L\mu$. The problem was approached by many researchers with various techniques. The completeness of the original Kozen's system was eventually established by Igor Walukiewicz [23], working at the University of Warsaw at that time. Twenty years later, the paper has been distinguished by the *Test of Time Award* of the conference *IEEE LICS 2015*.

Viewing in historical perspective, the result may appear close to the the spirit of Rasiowa's approach discussed above, as it concerns the completeness theorem for a semantics based on ordered algebraic structures, based on the Knaster–Tarski theorem on the existence of fixed points. However, Walukiewicz used quite different concepts and techniques, like automata on infinite trees, introduced by Rabin. The heart of the proof consists in showing that the Simulation Property, claiming that alternating tree automata can be simplified to non-deterministic ones, can be stated and proved in Kozen's system. There have been several attempts to find an algebraic proof of the completeness theorem, but till now no one succeeded. On the other hand, techniques invented by Walukiewicz in this context, turned out to be fruitful for solving many other problems, in particular to prove the so-called Muchnik Theorem on decidability of monadic theories of tree-like structures [24].

6. The λ-Calculus: Proofs *vs.* Programs

Let us come back to an opinion on intuitionism stated in the introduction to *Mathematics of Metamathematics* [8].

> *On the other hand, the mathematical mechanism of intuitionistic logic is interesting: it is amazing that vaguely defined philosophical ideas concerning the notion of existence in mathematics have led to the creation of formalized logical systems which, from mathematical point of view, proved to be equivalent to the theory of lattices of open subsets of topological spaces.*

There is yet another aspect of *the mathematical mechanism of intuitionistic logic*, which inspired interesting mathematical work. It was first discovered by Haskell Brooks Curry in the 1930s, but has been broadly recognized after the work of William Howard in 1969, and is currently known as the *Curry–Howard isomorphism* or *propositions-as-types* paradigm. It concerns an analogy between proofs in intuitionistic logic and terms in λ-calculus, as well as between theorems and types of terms. The algorithmic aspect of intuitionistic logic was investigated in Warsaw by Mordechaj Wajsberg in the 1930s, who essentially gave an algorithm to recognize tautologies of propositional intuitionistic logic [25]. The interest in λ-calculus revived among the computer science logicians in Warsaw since the 1980s, as that calculus, originally proposed by Alonzo Church as general framework for computation, is at the basis of the functional programming languages. Of numerous achievements of the λ team, let us mention the proof of undecidability of the semi-unification problem, by Kfoury, Tiuryn and Urzyczyn [26], which closed a longstanding open problem related to the type reconstruction in the polymorphic functional programming language ML, and a surprisingly elementary proof of undecidability of a simplified version of the second-order unification problem by Aleksy Schubert [27]. Paweł Urzyczyn is a co-author of the first monograph on the Curry–Howard isomorphism [28].

7. Logic of Uncertainty

Mathematical logic arose from a reflection on mathematical thinking, but the realm of logic is, of course, much larger; it comprises in particular experimental and legal sciences, humanities, as well as every day human experience. It took logicians some time to realize that "non-mathematical" logic may also lead to deep mathematical problems. Jan Łukasiewicz arrived at the idea of many-valued logic in 1920s while studying the philosophical opposition of determinism *vs.* indeterminism. The concept found numerous applications in computer science, including the practical fields of circuit design and program optimization. Łukasiewicz's ideas opened new paths in logic, indicating that notions of uncertainty, vagueness, ambiguity, can be dealt with in a rigorous way. This approach has been particularly rewarding in logic for artificial intelligence. One of the problems there is to induce general rules from large sets of observations (data), which can be incomplete, or even inconsistent. Among various techniques proposed for this problem, the method based on *rough sets* introduced by Zdzisław Pawlak in Warsaw in the early 1980s [29] was proved quite successful in practice, and received broad recognition from the AI community. The new ideas arose from a fruitful collaboration of Pawlak and numerous logicians in the Warsaw circle of that time, including Helena Rasiowa, Cecylia Rauszer, Ewa Orłowska, V. Wiktor Marek, Andrzej Skowron, and many others. The survey [30] gives an informative panorama of this development.

In about the same time, Witold Lipski proposed a mathematical model of a database with incomplete information [31]. Lipski died prematurely in 1985 in the age of 35, but his work had a great influence on the development of database theory.

8. The Limits of Automata Theory

Since 2000, a new generation of logicians entered the stage. Leszek Kołodziejczyk pursued an original path of research in the model theory of arithmetics in connection with complexity theory; let us mention his recent work (with co-authors) on proof complexity [32].

Mikołaj Bojańczyk initiated successful new research in Automata Theory, in connection with logic and database theory. He made a strong impact on research at the University of Warsaw, forming an active international team working in the area. Automata theory, originated in computer linguistic, has in recent years evolved into an autonomous field, which tries to explain the flow of information in complex systems. It bridges concepts and ideas of various fields, such as computability, logic, algebra, games, but also set theory. Of numerous contributions of Bojańczyk (by the way, the first recipient of the Presburger Award mentioned above), we discuss two topics.

The Rabin Tree Theorem mentioned in Section 5 inspired a stream of research, which succeeded in the extension of the decidability result to a larger class of models. One direction concerned trees generated by higher-order recursion schemes, where the Warsaw logicians made some contribution [33]. In contrast, Bojańczyk took an ambitious approach of extending the language of monadic second order theory (MSO) [34]. He introduced the concept of a new *unboundedness* quantifier, while pursuing some decidability questions regarding finite models. Roughly, $UX\,\varphi(X)$, means that there exist arbitrary large finite subsets X of the universe, satisfying φ. The resulting logic turned out to be decidable over finite or infinite words or trees, under the condition that all second order quantifiers are restricted to *finite* sets [35]. The general case of the logic MSO+U remained open, although a large decidable fragment of the logic has been identified by M.Bojańczyk and Thomas Colcombet [36]. Finally, this logic was proved undecidable already for infinite words [37]; the result was preceded by a partial result [38], showing undecidability over infinite trees subject to some set-theoretic assumption. These results, albeit negative, are quite interesting, as they disclose the limits of the regularity paradigm, which has been recognized as one of the most successful methods in computer science to obtain decidability results.

Another novel program extends the concept of automata in order to handle infinite sets of data; it has been developed by M.Bojańczyk, Bartek Klin, Sławomir Lasota, Szymon Toruńczyk, and others. The idea is to found the concept of an automaton in an alternative set theory, namely the theory of nominal sets, or sets over atoms. Such a theory was considered in the 1930 by Fraenkel and also by Andrzej Mostowski, with the aim to show the independence of the Axiom of Choice from the other axioms of Zermelo–Fraenkel set theory (It was beneficial for young Mostowski that in 1936–38 he could study in Vienna and Zurich, where he met, among others, Kurt Gödel, Paul Bernays, and Hermann Weyl [39].). In this theory, the universe of sets is constructed from a countable set of indistinguishable atoms (originally called *urelements*). This gives rise to a concept of *orbit finite* sets which, being physically infinite, enjoy many properties of finite sets; such sets can be used in computation theory almost like finite sets. The approach yielded a proposal of a programing language over nominal sets [40]. The investigation of complexity questions in the new framework revealed a connection with the Constraint Satisfaction Problem, one of the mainstream topics in complexity theory [41].

One may ask if there is any feature of the Warsaw logic group, which passes from one generation to another. It is not clear. Openness to new trends of the world is sometimes evoked as *genius loci* of the city. The young logicians in Warsaw have had always complete freedom to follow their own paths, with no pressure of tradition. And perhaps this, paradoxically, is the essence of our tradition?

Acknowledgments: The author wishes to thank Marek Zawadowski for an enlightening conversation about the heritage of Helena Rasiowa. Thanks go also to Aleksy Schubert for useful information, in particular about Mordechaj Wajsberg. The author is deeply indebted to Janos Makowsky, as well to three anonymous referees, for careful reading of the original version of the manuscript, and for many useful comments and suggestions, in particular concerning including some themes and personages in the revised version.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Grzegorczyk, A. Some classes of recursive functions. *Rozpr. Matemayczne* **1953**, *4*, 1–45.
2. Banach, S.; Mazur, S. Sur les fonctions calculables. *Ann. Soc. Pol. Math.* **1937**, *16*, 223.

3. Mazur, S. Computable analysis. *Rozpr. Matemayczne* **1963**, *33*, 1–111.
4. Specker, E. Nicht Konstruktiv Beweisbare Sätze Der Analysis. *J. Symb. Log.* **1949**, *14*, 145–158.
5. Feferman, S. Tarski's Influence on Computer Science. In Proceedings of the 20th IEEE Symposium on Logic in Computer Science (LICS), Chicago, IL, USA, 26–29 June 2005.
6. Tarski, A. Sur les ensembles définissables de nombres réels I. *Fundam. Math.* **1931**, 17, 210–239.
7. Presburger, M. Über der Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchen die Addition als einzige Operation hervortritt. In *Comptes Rendus Premier Congrès des Mathématicienes des Pays Slaves*; Leja, F., Ed.; Książnica Atlas T.N.S.W.: Warsaw, Poland, 1929; pp. 92–101.
8. Rasiowa, H.; Sikorski, R. *The Mathematics of Metamatematics*, 3rd ed.; PWN—Polish Scientific Publishers: Warszawa, Poland, 1963.
9. Engeler, E. Algorithmic properties of structures. *Theor. Comput. Sci.* **1967**, *1*, 183–195.
10. Salwicki, A. Formalized algorithmic languages. *Bull. Acad. Polon. Sci. Ser. Sci. Math. Astron. Phys.* **1970**, *18*, 227–232.
11. Mirkowska, G. On formalized systems of algorithmic logic. *Bull. Acad. Polon. Sci. Ser. Sci. Math. Astron. Phys.* **1971**, *19*, 421–428.
12. Kreczmar, A. Effectivity Problems in Algorithmic Logic. Ph.D. Thesis, Faculty of Mathematics and Mechanics, University of Warsaw, Warsaw, Poland, 1973.
13. Rasiowa, H. ω^{+}-valued algorithmic logic as a tool to investigate procedures. In *Mathematical Foundations of Computer Science*; Springer: Berlin, Germany, 1975; pp. 423–450.
14. Mirkowska, G.; Salwicki, A. *Algorithmic Logic*; PWN—Polish Scientific Publishers: Warszawa, Poland, 1987.
15. Kreczmar, A.; Salwicki, A.; Warpechowski, M. *LOGLAN'88—Report on the Programming Language*; Springer: Berlin, Germany, 1990.
16. Pratt, V. Semantical Considerations on Floyd-Hoare Logic. In Proceedings of the 17th Annual IEEE Symposium on Foundations of Computer Science, Houston, TX, USA, 25–27 October 1976; pp. 109–121.
17. Harel, D.; Kozen, D.; Tiuryn, J. *Dynamic Logic*; MIT Press: Cambridge, MA, USA, 2000.
18. Tiuryn, J.; Urzyczyn, P. Some Relationships between Logics of Programs and Complexity Theory (Extended abstract). In Proceedings of the 24th Annual Symposium on Foundations of Computer Science, Tucson, AZ, USA, 7–9 November 1983; pp. 180–184.
19. Immerman, N. *Descriptive Complexity*; Springer: Berlin, Germany, 1999.
20. Kozen, D. Results on the propositional μ-calculus. In *Automata, Languages, and Programming*; Springer: Berlin, Germany, 1982; pp. 348–359.
21. Niwiński, D. Fixed points *vs.* infinite generation. In Proceedings of the Third Annual Symposium on Logic in Computer Science, Edinburgh, UK, 5–8 July 1988; pp. 402–409.
22. Rabin, M.O. Decidability of second-order theories and automata on infinite trees. *Trans. Am. Soc* **1969**, *141*, 1–35.
23. Walukiewicz, I. Completeness of Kozen's Axiomatisation of the Propositional mu-Calculus. In Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science, San Diego, CA, USA, 26–29 June 1995; pp. 14–24.
24. Walukiewicz, I. Monadic second-order logic on tree-like structures. *Theor. Comput. Sci.* **2002**, *275*, 311–346.
25. Wajsberg, M. Untersuchungen über den Aussagenkalkül von A. Heyting. *Wiad. Matematyczne* **1938**, *46*, 45–101.
26. Kfoury, A.; Tiuryn, J.; Urzyczyn, P. The Undecidability of the Semi-unification Problem. *Inf. Comput.* **1993**, *102*, 83–101.
27. Schubert, A. Second-Order Unification and Type Inference for Church-Style Polymorphism. In Proceedings of the 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Diego, CA, USA, 19–21 January 1998; pp. 279–288.
28. Sørensen, M.H., Urzyczyn, P. *Lectures on the Curry-Howard Isomorphism*; Studies in Logic and Foundations of Mathematics; Elsevier: Amsterdam, The Netherlands, 2006; Volume 149.
29. Pawlak, Z. Rough sets. *Int. J. Parallel Program.* **1982**, *11*, 341–356.
30. Jankowski, A.; Skowron, A. Logic for Artificial Intelligence: A Rasiowa—Pawlak School Perspective. In *Andrzej Mostowski and Foundational Science*; Ehrenfeucht, A., Marek, V.W., Srebrny, M., Eds.; IOS Press: Amsterdam, The Netherlands, 2008; pp. 106–143.

31. Lipski, W. On Semantic Issues Connected with Incomplete Information Databases. *ACM Trans. Database Syst.* **1979**, *4*, 262–296.
32. Buss, S.R.; Kołodziejczyk, L.; Zdanowski, K. Collapsing modular counting in bounded arithmetic and constant depth propositional proofs. *Trans. Am. Math. Soc.* **2015**, *367*, 7517–7563.
33. Knapik, T.; Niwiński, D.; Urzyczyn, P. Higher-Order Pushdown Trees Are Easy. In *Foundations of Software Science and Computation Structures*; Springer: Berlin, Germany, 2002; pp. 205–222.
34. Bojańczyk, M. A Bounding Quantifier. In *Computer Science Logic*; Springer: Berlin, Germany, 2004; pp. 41–55.
35. Bojańczyk, M.; Toruńczyk, S. Weak MSO+U over infinite trees. In Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science, Paris, France, 29 February–3 March 2012; pp. 48–66.
36. Bojańczyk, M.; Colcombet, T. Bounds in ω-Regularity. In Proceedings of the 21th IEEE Symposium on Logic in Computer Science, Seattle, WA, USA, 12–15 August 2006; pp. 285–296.
37. Bojańczyk, M.; Parys, P.; Toruńczyk, S. The MSO+U Theory of $(N, <)$ is undecidable. In Proceedings of the 33rd Symposium on Theoretical Aspects of Computer Science, Orléan, France, 17–20 February 2016; pp. 1–8.
38. Bojańczyk, M.; Gogacz, T.; Michalewski, H.; Skrzypczak, M. On the Decidability of MSO+U on Infinite Trees. In *Automata, Languages, and Programming—41st International Colloquium*; Springer: Berlin, Germany, 2015; pp. 50–61.
39. Krajewski, S.; Srebrny, M. On the Life and Work of Andrzej Mostowski (1913–1975). In *Andrzej Mostowski and Foundational Science*; Ehrenfeucht, A., Marek, V.W., Srebrny, M., Eds.; IOS Press: Amsterdam, The Netherlands, 2008; pp. 3–14.
40. Bojańczyk, M.; Braud, L.; Klin, B.; Lasota, S. Towards nominal computation. In Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Philadelphia, PA, USA, 22–28 January 2012; pp. 401–412.
41. Klin, B.; Lasota, S.; Ochremiak, J.; Toruńczyk, S. Turing machines with atoms, constraint satisfaction problems, and descriptive complexity. In Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), Vienna, Austria, 14–18 July 2014; pp. 58:1–58:10.