Zdzisław Pawlak, Databases and Rough Sets

V. W. Marek
Department of Computer Science
University of Kentucky
Lexington, KY 40506

Abstract

We discuss work of Zdzisław Pawlak in the area of databases and the extension of that work to the theory of rough sets. In particular we look at his motivations for introducing information storage and retrieval systems and how this, eventually, led to rough sets theory.

1 Introduction

In this memoir I am recalling my collaboration with Professor Zdzisław Pawlak, especially during 1970ies and early 1980ies. This period coincides with two ideas that originated with Pawlak during that time: a model of databases (it was called *information storage and retrieval systems* and was pursued by a group of scientists in Poland and in other research centers mostly in Eastern Europe) and then later work on approximating sets (of records or other objects) by means of some pairs of sets of objects. This latter theory is now called *rough sets* and again originated with Pawlak. The reason why I write about these areas is that during that specific period I was a close collaborator of Pawlak and worked with him on a variety of projects related to these two areas.

Of course, Pawlak studied many other areas of Computer Science, and more generally, Mathematics. Specifically, he contributed to the area of models of computation, data structures, combinatorial optimization, theory of conflicts – to name a few. I am sure these contributions will be discussed by others – I will focus on information storage and retrieval systems and on rough sets.

Let me first describe how this all started. In 1960, fresh from high school, I started studies of Mathematics at Warsaw University. There was no Computer

Science program at Warsaw at the time, but some aspects of Computer Science were taught in the Numerical Analysis program which was part of Mathematics. Very soon I was attracted to Foundations of Mathematics. Warsaw, of course, had a strong tradition of Foundations. There were several groups of researchers pursuing foundational studies. The strongest group was centered around Professor Andrzej Mostowski who was both the head of Foundations section at the Mathematical Institute of Polish Academy of Sciences and, at the same time, the Chair of Algebra at the University. Other notable logicians at the time included Professor Helena Rasiowa (Chair of Logic at the Warsaw University and later close collaborator of Pawlak), Professor Andrzej Grzegorczyk (first at the University, then at Polish Academy of Science), Professor Wanda Szmielew (Chair of Foundations of Geometry at the University) and Professor Jerzy Łoś. Professors Rasiowa and Grzegorczyk were Mostowski's students. Professor Szmielew was Alfred Tarski's student. Although Tarski (since the middle of WW II) was at the University of California, Berkeley, he somehow influenced Warsaw foundational research – in spite of the fact that at the time a "Cold War" was raging between the countries dependent of Soviet Union and so-called the West and so communications were sporadic, censored, and slow. In addition to the forenamed researchers I soon met two other: One was Andrzej Ehrenfeucht, and the other Zdzisław Pawlak. Both, at the time, were working at the Mathematical Institute of Polish Academy of Sciences. Zdzisław was a computer engineer, and Andrzej was a logician, with clear interests in Foundations.

There were several opportunities for participation in classes and seminars devoted to Foundations. Each of the principals mentioned above taught some lectures and lead seminar series. As a sophomore and then a junior at the University I joined two. One was the General Foundations Seminar, usually convening on Wednesday, 5pm (essentially in Tarskian tradition). That seminar was lead by a distant (at least at that time) figure – Professor Andrzej Mostowski. All current major results in Foundations were presented there. There were also other series. Besides of Professor Rasiowa seminar (dominated by the algebraic approach to logic), Professor Szmielew seminar (Foundations of Geometry) and, occasionally, Professor Grzegorczyk seminar, there was Ehrenfeucht and Pawlak seminar at the Mathematical Institute. The atmosphere there was very informal. Unlike in other seminars the "thou" form was used there, and this informality and relaxed atmosphere certainly appealed to people like myself. The audience was, unlike in other seminars, very diverse: logicians, probabilists, philosophers, computer scientists and even medical researchers. A number of papers devoted to automated theorem proving were read there. I remember two – pioneering Hao Wang work on proving tautologies using computers and Davis and Putnam work on resolution. The memories of events that happened some 50 years ago are blurred; somehow the presentation of Professor

Ewa Orłowska comes to mind.

Very soon I started to talk regularly to Andrzej Ehrenfeucht and, eventually, at his suggestion and with his guidance I wrote a master (M.Sc) thesis. Like Mostowski (but unlike the others), Andrzej was a "generalist" of Foundations - he had extensive knowledge of all major areas of Foundations: Proof Theory, Set Theory, Recursion Theory, and especially Model Theory. For reasons not entirely clear today, Andrzej suggested for my M.Sc. thesis a topic from Combinatorial Set Theory. Maybe the reason was that he heard a talk on set-theoretical topology and immediately saw a generalization? Maybe he heard it from someone who heard it from the great combinatorist Paul Erdös? Anyway, as I recall, we were sitting in a cafe at Marszałkowska Street in Warsaw (close to Constitution Square) eating cakes, drinking coffee (likely Andrzej also drank cognac - he could afford it and he liked it then) and Andrzej suggested a problem closely related to so-called Δ-lemma. He told me to use a specific form of induction. I was reading two-year Mostowski's course in Set Theory and it fit together very well. Soon the problem was solved, publication written, thesis defended, and as a result I became a teaching assistant at Mostowski's group. The year was 1964, and at exactly that year a breakthrough in Foundations of Set Theory occurred - Paul J. Cohen of Stanford University invented a new technique called "forcing". He utilized it to prove independence of Continuum Hypothesis, a problem stemming out of famous Hilbert problems. We, in the vicinity of Mostowski, dropped everything and started to research the area of Foundations of Set Theory. So, naturally, Foundations of Computer Science went (fortunately temporarily for me) away. Moreover, soon Andrzej Ehrenfeucht left for United States and after short stint in California settled in Colorado. He is not a person who reads or writes much. Actually, to this day, he writes almost nothing, but magically knows much. This lack of communication created a vacuum, at least for me and for a couple of years I did exotic things like studies of second-order arithmetic (what is it?), constructible hierarchy (even worse!) and other fashionable, but remote from Computer Science areas. After getting a Ph.D. degree in 1968 and post-doc'ing in Holland in 1970/71, as I was returning to Poland I took a detour and visited Janusz Onyszkiewicz (another Warsaw logician) who was at Aarhus University in Denmark for a year. I noticed there that logicians were deeply engaged in various problems stemming from Computer Science. This must have influenced me somehow because once back in Warsaw, in addition to the research discussed above, I started to look at the areas further from Mostowski-style Foundations. Soon a series of phone conversations with Zdzisław on some issues related to something related to databases followed. I knew very little about databases, and at the time I was not familiar with the work of E.F. Codd on relational model of databases. Worse, I did not understand the issues. Likely, nobody in Warsaw did. That is, except Zdzisław. He somehow knew that the time to apply various techniques of logic to databases has come. In the next section I will describe the work done by Zdzisław and myself, and how it is related to databases as we know them now. This research coincided with the significant changes in the attitude of Polish scientific community toward Computer Science. In rapid succession, and obviously Zdzisław was instrumental in these things happening, several things occurred. First, Computer Science program was established at the Warsaw University (Professor Rasiowa, the then Dean of Mathematics and Physics, later Mathematics and Mechanics was also deeply involved). Second, the Academy of Sciences converted its “Computational Center” into a Computer Science Institute (the formal change, including the name change, came later), thus creating another place where Computer Science researchers could be employed. There was yet another important change. Warsaw Technical University opened a highly competitive program called “Technical Physics and Applied Mathematics” which was, in reality, Computer Science. That program, due to its competitiveness and prestige, attracted a cream of computationally minded young students from all over Poland. Many of these individuals soon became young researchers. I did not realize this at the time but a number of these individuals were ready for serious research work in Computer Science. The most advanced among these students and researchers was Witold Lipski (unfortunately died early - we have today the annual Lipski Competition for the brightest Polish young researcher in Computer Science). Zdzisław and I worked with Lipski who very quickly wrote a Ph.D. dissertation on information storage and retrieval systems, became a well-known database theorist and combinatorist. I will discuss how the combinatorics came into the picture in the next section. The most important aspect of that work was that very quickly we had in Warsaw a large group of young researchers working on databases and combinatorics. Soon, Lipski had a number of collaborators, both in database theory and combinatorics. In the next section I will discuss how these two areas related in our work.

The interest in databases and their query languages came naturally to Pawlak. For a number of years Zdzisław collaborated with a number of physicians. There were no subspecialty of Medical Informatics at the time, and one needed a vision to see that Medicine will be revolutionized by the computer applications. Nobody (at least in Warsaw) could imagine databases of medical cases. Except, of course, Zdzisław. He realized the potential of storing the medical data in databases and, more importantly data mining the data so stored.

The formal descriptions of databases, query languages and the possibility of testing formal properties of such databases was a major driving force of his considerations. And of course this projected on the work of the group of people around him. An important aspect (I will discuss it in more detail below) was that the (anonymized) records stored in database did not form a set, but rather a bag (the inventors of relational model were, originally, against such approach, but today it is widely accepted).

There were several consequences of such possibility (i.e. existence of undistinguishable objects). Namely it is possible, even likely, that the query language is inadequate to describe the answers to queries that the user would like to make. This is a common case in medical applications. Often physicians see the symptoms, not the underlying causes, and the language expressing symptoms may be inadequate to describe the essence of the underlying medical problem.

The key insight that Zdzisław had was that given a description language (which results in database scheme) that language may be good enough only for the approximate description of the set of objects that interests the user. There is more than one situation that occurs here and I will discuss the reasons for approximations in Section 3. Going into studies of approximations immediately changed the perspective. Namely several new aspects arose. For instance: *What are the measures of approximation?, What are the query languages adequate for specific measures of approximation?, Can one eliminate some attributes without lowering the quality of approximation?* and many other questions. Not surprisingly the resulting theory of *Rough Sets* related to Logic, Universal Algebra, but also to various aspects of Statistics. This is the source of the popular Rough Sets Theory widely studied today throughout the world.

2 Information storage and retrieval systems, databases

So, what were those information storage and retrieval systems (*i.s.r.* for short), and how were they motivated? The general question, namely *What is a database and what are formal properties of databases?* was not settled at the time. Today, the researchers of database theory think about databases as relational systems in the sense used by logicians [1]. Surely, since the database is supposed to be stored, the relations (often called tables) need to be finite. Therefore the corresponding logical system that may be used to describe these relational systems is some form of *finite model theory*, a fragment of model theory first studied by Y. Gurevich. Thinking about databases as collections of relations (tables) was proposed by E.F. Codd of IBM and quickly gained acceptance, first among theoreticians and then also, by use of query languages such as SQL, with a wide community of users (to be fair to others, there are many alternative ways of thinking about databases). Moreover, in a couple of years, there were *implementations* of relational databases, and soon they became competitive in their performance with respect to the older, non-SQL, systems.

Prior to the relational model, database systems were based on so-called network model and on hierarchical model. These previous models could not be, really, explained to the user community since they involved understanding processing of data within database systems, in particular data structures such as linked or double linked lists. Relational model thinking was *declarative*; for the first time the user was thinking about *what* information she wants to get out of the system, not *how* she wants to get it.

Coming back to i.s.r., it was defined as a relational system, but on a single table. Also, it was heavily influenced by logic (rather than relational algebra). Let me shortly describe what happens when logicians look at the databases. First, one has to have a language. In case of databases, if one has to study sets of records, those need to be described. For that reason, one needed a language. The language had means to introduce *descriptors*. For that reason, Pawlak proposed to have a collection (called A) of all descriptors of the system. Actually, SQL does precisely the same (although in a clearer way, by means of types of attributes which are always finite, since even types such as integer, or real, are in reality finite). Descriptors split into classes called attributes. A natural way to do this is by means of an equivalence relation on the set of descriptors. Such construction presupposes that that for attributes a_1 and a_2 the descriptors of type a_1 and of a_2 are disjoint. This may appear to be limiting, but really is not. For if we have three dimensions of a box and measure the size then to describe a red box 35 centimeter long, 36 centimeters wide and 20 centimeters high we can use the record

$$\langle length : 35, width : 35, height : 20, color : red\rangle.$$

In this way the length equal to 35cm and width equal to 35 are disambiguated.

So now, an i.s.r. is a relational system $\mathcal{S} = \langle X, A, R_I, U\rangle$ where X is a set of objects (think about records but not necessarily different), S is the set of descriptors, R_I an equivalence relation partitioning descriptors into *attributes*, and finally, U is a function assigning to each descriptor $d \in A$ a subset of X consisting of records with value d. Since the descriptors (like in our example) carried the information about the attribute to which they belonged, there were no ambiguity. For instance, $U(height : 20)$ was the set of (descriptions of) boxes that had the height equal to 20. This choice of definition was motivated by the concept of *inverted file*, a construction not taught today in database courses (but current at the time), in which one stores for some or all descriptors the set of identifiers of records with that descriptor.

Once we have descriptors, we can build a free Boolean Algebra over that set. It is natural; to find the set of boxes with the length equal to 35 and width 35, we need to compute intersection (Boolean meet) of two sets: of objects with the *length* 35, and of objects with the *width* 35. To facilitate answers to such queries (*give me the set of all boxes of length 35 and width 35*), the i.s.r. had the syntactical category of *terms*. These, in today parlance of SQL, corresponded to the queries that the user can ask. There was an inductive definition of term, and the evaluation function $||\cdot||$. This evaluation function did what (simple) SQL queries do: returned the bag of records satisfying the suitable boolean condition. So the SQL query: SELECT * FROM *boxes* WHERE *length* = 35 AND *width* = 35; would be written as $||length : 35 \cdot width : 35||$. The query language of i.s.r. was significantly weaker than that of (even quite simple) SQL, since there were no comparators; all that were expressible were Boolean operations. We soon realized that this was a problem, and added extensions that allowed for comparisons, but we never truly recognized that the comparators are important. Moreover, SQL allows for "hiding" values of some attributes by means of projection operator. This was not available in i.s.r. SQL treats the answers to all queries as tables and these tables may have different schemata. This again was not available in i.s.r. But there were some advantages, too. Specifically, terms offered a possibility of describing formulas - the properties of the system in its entirety. SQL systems did not offer (and still do not offer) such capabilities, namely imposing general integrity constraints on the system[1]. The i.s.r. research did not study directly the first issue, but studied the second one. To give an example of the issues, let us assume that there is an additional attribute color. The language of i.s.r. allowed to express the properties of the systems such as "All red boxes have length 35". Today's SQL systems do not offer the language for testing such integrity constraints, although the user can write a program testing for such properties using so-called embedded SQL.

The researchers of i.s.r. devoted a significant amount of attention to various aspects of possible implementation of such systems. While today storage is inexpensive and all sorts of data are collected with massive databases of immense size, the situation was different in the 1970ies. Storage was expensive and processing was slower. This lead to two important research topics; first, decomposing databases so that they required less space, second, organizing data on disk so that answers to *some* queries were computed in a simple manner. To give the example of this second issue, if the records of boxes with the length equal to 35 form a segment in the underlying organization of data then the answer of the query $||length : 35||$ is simple and requires minimal number of accesses to the disk. While the issue of decomposition of data went, essentially, away (we no longer require our students to normalize the data "to death", and normal forms beyond the so-called third normal form are not taught), the issue of organization of data did not go away and we are still concerned with minimizing the number of accesses to the disk. The

[1]Of course, *some* integrity constraints can be declared in SQL, but generally, SQL limits the capabilities of the database designer to specify the integrity constraints.

theory behind the organization of data is a well-established topic. It involves both combinatorics (here the mathematics comes into play) and data structures. The mathematical foundations of the technology of storage were based, at the time, on *interval graphs* ([3]) and on the theory of Boolean matrices with *consecutive* 1*s property*. Lipski and his collaborators (this included me ([4]), and generally was a subclass of Pawlak's research group) devoted a lot of attention to these issues. In the modern setting, today, the issue did not disappear. As the larger amount of data is stored and then processed, the issue of quick retrieval becomes even more important since moving the data through the network becomes a "chocking point". For that reason the researchers of so-called Cloud Computing pay significant attention to data organization.

The work on i.s.r. under the name of *information systems* concerned the group of researchers around Pawlak throughout 1970ies and resulted in a large body of research, eventually leading to studies of rough sets which I will report in the next section. One legacy of that research which slowly gained an acceptance in the mainstream database community was that the records can have duplicates. In theoretical terms this means that the tables are bags of records, not sets of records. This was obvious to Pawlak and his collaborators, because the language of i.s.r. naturally admitted a situation where two different objects had exactly same descriptions (certainly a common situation in databases of medical cases - one of the main motivation of Pawlak). To sum up, the investigations of i.s.r. prepared the ground for future related research on rough sets which will be discussed in the next section.

3 Rough Sets

The issue of the inadequacy of formal description language to describe desired families of sets of objects plagued (and still plagues) Computer Science. The nature of human natural language is such that when there is no adequate definition of some concept, we can invent an appropriate definition "on the fly". That is, the natural language constantly invents new concepts and vocabularies. With the formalized languages, for instance of predicate calculus, change of vocabulary is still possible, but with each change comes the change of semantics and, often, of processing algorithms. The question of the changing language used to describe i.s.r. concerned the researchers from the beginning. The formal means to describe the inadequacy of the language was, again, a certain natural equivalence relation that can be associated with a given i.s.r. $\mathcal{S}$. Namely, $\mathcal{S}$ induces an equivalence relation in the set X of objects. This relation $\sim$ is defined as follows: $x \sim_{\mathcal{S}} y$ if for all descriptors a, $x \in U(a) \equiv y \in U(a)$. We will drop the subscript $\mathcal{S}$ when the system $\mathcal{S}$ is fixed. Hence, $x \sim y$ holds when, from the point of view of the query language of $\mathcal{S}$, the objects x and y are undistinguishable. The equivalence classes (cosets) of $\sim$ are minimal units that the language of $\mathcal{S}$ allows to describe. Assuming there is finite number of equivalence classes of $\sim$ (one can theoretically think about infinite $\mathcal{S}$'s but these do not appear in reality), the subsets of X that are describable by means of the query language of $\mathcal{S}$ are *precisely* the unions of (finite collections) of these equivalence classes. Let us call, for the lack of a better word, these equivalence classes *monads*. If monads can have more than one element, then we face the following dilemma: What to do if that family of descriptions is inadequate to the needs of the user? There are several situations where we see this inadequacy. First, there may be situations where the monads are too big - in reality the descriptions should be finer, but we do not have the language good enough to describe the differences. This is common situation in medicine. Physicians strive to have adequate description of the underlying biological system (the patient) in objective terms. But before such description can be found, the less precise descriptions in forms of symptoms experienced by the patient is all that is available. But the same symptoms may show up in different medical conditions. In fact, the discovery of objective values is the subject of what is commonly known as medical tests, and the process of differentiation of description is practiced in medicine under the name of *differential diagnosis*. So, in this situation, which we call situation I the available query language is inadequate. But there is also another situation where the query language is adequate, but the shortest description of a set of interest is too big. To see what happens in this second situation (situation *II*) let us observe that, in principle, the number of monads is proportional to the product of the sizes of cosets of the relation R_I of the i.s.r. $\mathcal{S}$. Every set-theoretical union of monads is describable, but such descriptions may be very long! The question that Pawlak asked was how to handle both situations. His idea, first described in the paper [6], and then elaborated in details in his book [8] was to use approximations. But what approximations? It turned, eventually, out that more than one concept is involved. The question of inadequacy of the language can be treated as follows. With every subset $Y \subseteq X$ we can assign two definable subsets of X. Namely the *greatest* definable subset of Y and the *least* definable superset of Y. These sets are commonly denoted $\underline{Y}$ and $\overline{Y}$, respectively. Of course, $\underline{Y}$ is the union of all monads included in Y, while $\overline{Y}$ is the union of monads that have nonempty intersection with Y. Having the concepts of $\underline{Y}$ and $\overline{Y}$ allows to *measure* inadequacy of the language of i.s.r. to describe a set Y. A variety of measures is possible, for instance the ratio of sizes of $\underline{Y}$ and Y (which is defined whenever $Y \subseteq X$ is nonempty). But there are other measures, too. For instance the ratio of upper approximation to the lower one, or of the size of Y to its upper approximation.

If we take the minima of the measures described above over all nonempty subsets of X, we get adequacy measure for the language itself! In other words, approximations allow to measure inadequacy of the language.

We observe that the information theorists devoted a significant attention to this problem. Some of the proposals such as *minimum description length* can be found in [9]. Once one starts to *measure* adequacy, new problems come to mind: feature extraction, feature constriction etc. We then land in the world of *machine learning*.

As mentioned above, the language can be adequate to describe a set $Y \subseteq X$, but description may be too large! If this is the case then we would like to find approximations of the set Y using a weaker language than that available from $\mathcal{S}$. This situation, under the name of *attribute reduction* trades precision for conciseness. Namely, we are willing to accept a pair of imprecise, but concise descriptions instead of one precise but impossibly long description.

Generally, then we trade impossibility of *adequate* description (either because of nonexistence of such description, or inadequacy of such description because of its size) by moving to approximations. It turns out (as shown in [7]) that rough sets (Pawlak approximations) can be characterized (in some precise sense) as best possible approximations. It did not surprise us, as in many situations Pawlak's intuition turned out to be very strong and confirmed by adequate mathematics. It is also worth mentioning that while the presence of a i.s.r. (that provides the description language) is beneficial, it is not a necessary ingredient of the approximation – all we need for this is the indiscernibility relation $\sim$, the point of view commonly accepted by the rough sets researchers. This level of abstraction allows to tie (as done of many researchers) rough sets with the universal-algebraic concepts of Boolean algebras with operators [2], and also with finite topologies.

4 Conclusions

Zdzisław was a true renaissance man, with many interests besides computer science he produced artistic short movies, wrote poetry, but also did practical things. Whatever he did the same enthusiasm and pervasive optimism present in his scientific work demonstrated itself in his actions. I, of course, benefited when he decided to build a shower stall in my small cottage in the country. Certainly the contrast between an academician living in “ivory tower” and a mason with his bricklayer trowel could not be bigger. Among many passions Zdzisław had was antique restoration. Like everything he did, this passion was contagious. So, when he and I brought to my apartment in Warsaw a round table bought in a consignment store (and in obvious need of restoration) my family was not pleased, and I had a new occupation for few months. I recall discussing with Zdzisław heating water using solar energy (yes, this was in 1970ies!) and other innovations.

I left Poland in tumultuous year 1982 and in 1983 settled in Lexington, KY. Occasionally I looked at Rough Sets (I mentioned one of those revisits above, there were other returns to that area of research, as well) but focused on another area, called nonmonotonic logic. This area dealt with another inadequacy of commonsense logic: namely of tentative and defeasible conclusions. This article is not the place to tell the story of that research. But of course, I followed developments in Rough Sets theory and met Zdzisław both in Poland when it was again possible to visit after the revolutions of 1989 and during his visits in the States. In particular I went to Nashville, TN (not far away from Lexington, at least for American distances) when in 1995 Zdzisław made an invited presentation for the ACM which resulted in next year of Zadeh prize in Soft Computing.

As I am looking back, one thing is certain: working with Zdzisław was more than just science, it was life to the fullest.

Acknowledgements

We thank Professor Andrzej Skowron for initiating this paper and for a number of suggestions for improvements.

References

[1] S. Abiteboul, R. Hull and V. Vianu, *Foundations of Databases*, Addison-Wesley, 1995.

[2] B. Jonsson and A. Tarski. Boolean Algebras with Operators. *Am. J. Math. 73*:891–939, 1951.

[3] C. Lekkerkerker and D. Boland, Representation of finite graphs by a set of intervals on the real line. *Fund. Math* 51:45–64, 1962.

[4] W. Lipski and W. Marek. File organization, an application of graph theory, *Springer L.N. in Comp. Sci.* 14, pages 270–279, 1974.

[5] W. Marek and Z. Pawlak. Information Storage and Retrieval Systems: Mathematical Foundations. *Theor. Comput. Sci.* 1:331–354. 1976.

[6] W. Marek and Z. Pawlak. Rough Sets and Information Systems. Computer Science Institute of PAS, Technical Report 441, 1981.

[7] W. Marek and M. Truszczynski. Contributions to the theory of Rough Sets. *Fund. Inform.* 39:389–409, 1999.

[8] Z. Pawlak. *Rough Sets*. Kluwer, 1991.

[9] J. Rissanen.*Information and Complexity in Statistical Modeling*, Springer-Verlag, 2007.

Use your QR-barcode reader to get to the e-repository of my papers.