Efficient Rough Set Theory Merging

Adam Grabowski

Institute of Informatics
University of Białystok
ul. Akademicka 2
15-267 Białystok, Poland
adam@math.uwb.edu.pl

Abstract. Theory exploration is a term describing the development of a formal (i.e. with the help of an automated proof-assistant) approach to selected topic, usually within mathematics or computer science. This activity however usually doesn't reflect the view of science considered as a whole, not as separated islands of knowledge. Merging theories essentially has its primary aim of bridging these gaps between specific disciplines.
As we provided formal apparatus for basic notions within rough set theory (as e.g. approximation operators and membership functions), we try to reuse the knowledge which is already contained in available repositories of computer-checked mathematical knowledge, or which can be obtained in a relatively easy way. We can point out at least three topics here: topological aspects of rough sets – as approximation operators have properties of the topological interior and closure; lattice-theoretic approach giving the algebraic viewpoint (e.g. Stone algebras); possible connections with formal concept analysis.
In such a way we can give the formal characterization of rough sets in terms of topologies or orders. Although fully formal, still the approach can be revised to keep the uniformity all the time.

Keywords: rough sets, knowledge management, formal mathematics.

1 Introduction

The era of extensive use of computers brought also an evolution of the mathematicians' work. Among new possibilities offered by computers we can point out the better transfer of knowledge between researchers via repositories of knowledge. Such computer algebra tools as Mathematica or MathCAD are very popular nowadays; researchers can also develop their own specialized software for computing relatively easier than before. The possibility of enhancing human work using automated proof assistants should be also underlined. We try to disscuss some issues concerned with the latter activity, concentrating on formalizing not only selected fields; but viewing specific disciplines from a wider perspective.

As we provided formal apparatus for basic notions within rough set theory (as e.g. approximation operators and membership functions), we try to reuse

the knowledge which is already contained in available repositories of computer-checked mathematical knowledge, or which can be obtained in a relatively easy way. We can point out at least three topics here: topological aspects of rough sets – as approximation operators have properties of the topological interior and closure; possible connections with formal concept analysis; lattice-theoretic approach giving the algebraic viewpoint (e.g. Stone algebras).

Our main aim is to develop (i.e. to describe in the formal computer language to be used within the repository of the existing mathematical knowledge) concrete examples of such formal knowledge reuse on the area of rough set theory. We also discuss some issues concerned with our implementation, but as we offer more than purely theoretical considerations (actual implementation is given), hence the word 'efficient' in the paper's title.

The structure of the paper is as follows: in the next section we present the overall methodological background for our work while in the third we focus on the activity of putting formal things together, called *merging theories*. Then we describe briefly the formal approach to rough sets we developed and some examples of successful, although not yet fully reused, bridging between various fields of formal mathematics.

2 Mathematical Knowledge Management

"Computer certification" is a relatively new term describing the process of the formalization via rewriting the text in a specific manner, usually in a rigorous language. Now this idea, although rather old (taking Peano, Whitehead and Russell as protagonists), gradually obtains a new life. As the tools evolved, the new paradigm was established: computers can potentially serve as a kind of oracle to check if the text is really correct. And then, the formalization is not *l'art pour l'art,* but it extends perspectives of knowledge reusing. The problem with computer-driven formalization is that it draws the attention of researchers somewhere at the intersection of mathematics and computer science, and if the complexity of the tools will be too high, only software engineers will be attracted and all the usefulness for an ordinary mathematician will be lost. But here, at this border, where there are the origins of MKM – Mathematical Knowledge Management, the place of fuzzy sets can be also. To give more or less formal definition, according to Wiedijk [26], *the formalization* can be seen presently as "the translation into a formal (i.e. rigorous) language so computers check this for correctness."

In this era of digital information anyone is free to choose his own way; to quote Vladimir Voevodsky, Fields Medal winner's words: "Eventually I became convinced that the most interesting and important directions in current mathematics are the ones related to the transition into a new era which will be characterized by the widespread use of automated tools for proof construction and verification". However he is focused as of now on the constructive Martin-Löf type theory many ordinary mathematicians aren't really familiar with. On the other hand, if we take into account famous Four Colour Theorem, automated

tools can really enable making some significant part of proofs, so hard to discuss with this opinion.

Among many available systems which serve as a proof-assistant we have chosen Mizar. The Mizar system [11] consists of three parts – the formal language, the software, and the database. The latter, called Mizar Mathematical Library (MML for short) established in 1989 is considered one of the largest repositories of computer checked mathematical knowledge. The basic item in the MML is called a Mizar article. It reflects roughly a structure of an ordinary paper, being considered at two main layers – the declarative one, where definitions and theorems are stated and the other one – proofs. Naturally, although the latter is the larger, the earlier needs some additional care.

As lattice theory (steered by Trybulec, Białystok, Poland) and functional analysis (led by Shidama, Nagano, Japan) are the most developed disciplines within the MML, further codification of rough sets, especially including their lattice-theoretic flavour, looks very promising. As a by-product, apart of readability of the Mizar language, we obtain also the presentation of the source accessible to ordinary mathematicians: pure HTML form with clickable links to corresponding notions and theorems.

3 Merging Theories

Theory exploration is a term describing the development of a formal (i.e. with the help of an automated proof-assistant) approach to selected topic, usually within mathematics or computer science. This activity however usually doesn't reflect the view of science considered as a whole, not as separated islands of knowledge. Merging theories essentially has its primary aim of bridging these gaps between specific disciplines. Of course, even digging deep in the area of selected discipline, eventually one have to use the apparatus from another field (usually category theory sheds some light), but this touches the informal layer, where interpretations can be somehat flexible.

In our CS&P 2012 paper [9] we have shown our translation of Zhu's paper about connections between ordinary properties of binary relations and underlying properties of rough approximation operators which proves some usefulness of proof assistants within a single area of research (essentially just the field of binary relations), but it is known that e.g., category theory gives nice interpretations for various questions; the same goes for modal logics. From another viewpoint, lattices can deliver similarly useful interpretations. Many fields can be reused depending on the author's preferred selected approach. Even in rough approach one can prefer either P-sets or I-sets (equivalence classes or just pairs) reflecting in the chosen language – either of ordinary sets (partitions) or subsets of Cartesian product.

We can consider merging on two levels:

Structures – when we inherit the overall signature of the object (as we can tell that groups are predecessors of rings or fields);

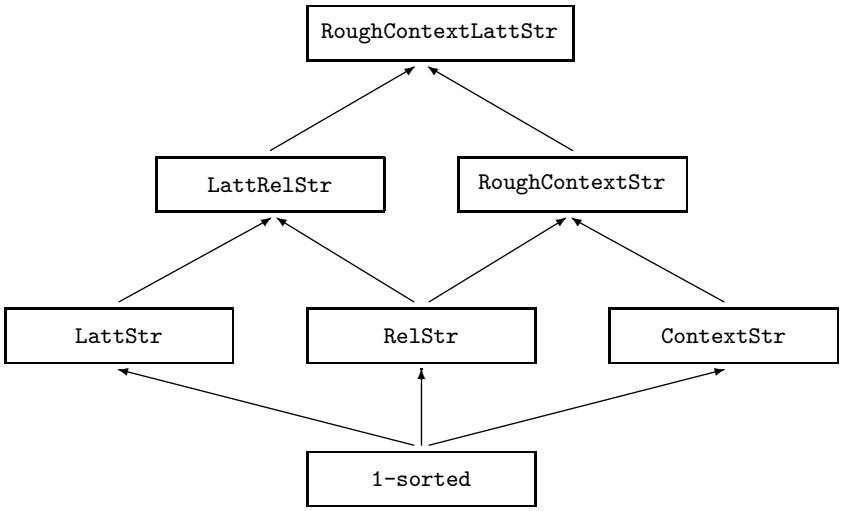

Fig. 1. The net of structures for chosen theories

Adjectives – when the hierarchy of axioms is described; here the example is that all Boolean algebras are Stone algebras.

Although from the informal point of view both given examples seem to be just the correspondence between axiom sets, formally this issue should be considered more deeply.

First of all, there are automatic theorem provers operating on the form of an equational characterization (collection of identities) of the theory. Hence the formula binding distinct items from a given signature gives more possibilities than the axiom postulating the existence of an object (even if we don't take into account Birkhoff variety theorem; equationally definable classes of mathematical structures are hereditary, admit homomorphic images and admit products – they form a variety). Good illustrative example here is the treatment of Boolean rings and Boolean algebras; we can see them as subvarieties of each other but formallywe should cope somehow with different signatures both are defined on. The same problem apears in the case of lattices viewed on the one hand as structures with join and meet operations or posets, otherwise. One can freely define lattices as posets with the existence of binary joins and meets; hence we obtain the algebraic interpretation of a lattice used, e.g. in universal algebra. Obviously both definitions are equivalent, buth they are definitely not the same as the order-theoretic one uses the signature

$$\langle L, \leq \rangle$$

while the algebraic one takes

$$\langle L, \sqcup, \sqcap \rangle$$

with binary operations: join $\sqcup$ and meet $\sqcap$.

Taking into account the aforementioned two stages of merging – on the level of structures both have really little in common as only the carrier L can be identical (we can call it a kind of syntactical point of view). But the latter viewpoint (of universal algebra) can give a path to semilattices $\langle L, \sqcup \rangle$ and $\langle L, \sqcap \rangle$ and here the second level of merging (semantical) really makes sense. Namely, on the signature of lower semilattice we can give an axiom of $\sqcap$-commutativity or $\sqcap$-associativity which can be then used on all its descendants. Both identities can be expressed as adjectives binded with appropriate structures.

The extensive use of identities in the form of attributes is really close to standard manipulation of axioms, so the example of the connection between Boolean and Stone lattices is really illlustrative here: as we work on the common signature $\langle L, \sqcup, \sqcap, ', 0, 1 \rangle$, there is no need to extend the corresponding structures and the work really depends on the deductive power of proof assistant (and computers do some computations which is quite natural).

Of course, the term 'formal' or 'formally' is used in this paper in two threads: on the one hand, 'formal' means the strict description of the rules governing the theory – in common use, it is 'rigorous' method. But hence all mathematics should be called formal in this sense, and this adjective should not then be used at all. There is also another interpretation of this attribute, which stems from Hilbert's formalism. In the latter view, computer assistance is the recent emerging trend which can be really controversial from the pen-and-paper mathematician viewpoint as the mathematics developed without machines for ages. Many computer scientists and mathematical intuitionists really advocate this approach, as Voevodsky who was quoted before.[1]

4 Rough Sets

Originally, we dealt with the more often used and methodologically simpler approach, i.e. equivalence relations-based rough sets. One of the key issues was also the possibility of further reusing, but soon this was automatically generalized. The concept of an information system can be also formalized as the descendant of the approximation space in a natural way. At the first sight, the underlying Mizar structure is `RelStr`, which has two fields: the `carrier` and the `InternalRel`, that is a binary relation of the `carrier`. The theory of relational structures has been developed and improved mainly during formalization of the *Compendium of Continuous Lattices* (which is described in [1] in detail). While in this context `RelStr` was used with attributes `reflexive transitive` and `antisymmetric` to establish posets, we decided to reuse it in our own way. First, we defined two new attributes: `with_equivalence` and `with_tolerance` which state that the `InternalRel` of the underlying `RelStr` is an equivalence resp. a tolerance relation (where a tolerance relation is a total reflexive symmetric relation, see [20]). With such defined notions, the basic definitions are as follows:

[1] Thanks go the anonymous referee for pointing out this inconsequence.

```
definition
  mode Approximation_Space is with_equivalence non empty RelStr;
  mode Tolerance_Space is with_tolerance non empty RelStr;
end;
```

Formalized theories can be treated as objects (axioms, definitions, theorems) clustered by certain relations based on information flow. The more atomic the notions are, the more is their usefulness. Driven by this idea we tried to drop selected properties of the equivalence relations. Our first choice was transitivity – therefore the use of tolerance spaces – as it seemed to be less substantial than the other two. The generalization work went rather smoothly. As we discovered soon, similar investigations, but without any machine-motivations, were done by Järvinen [14].

5 Formal Concept Analysis

Formal context analysis (FCA for short) has been introduced by Wille [27] as a formal tool for the representation and analysis of data. The main idea is to consider not only data objects, but to take into account properties (attributes) of the objects also. This leads to the notion of a *concept* which is a pair of a set of objects and a set of properties. In a concept all objects possess all the properties of the concept and vice versa. Thus the building blocks in FCA are given by both objects and their properties following the idea that we distinguish sets of objects by a common set of properties.

In the framework of FCA the set of all concepts (for given sets of objects and properties) constitutes a complete lattice. Thus based on the lattice structure the given data – that is its concepts and concept hierarchies – can be computed, visualized, and analyzed. In the area of software engineering FCA has been successfully used to build intelligent search tools as well as to analyze and reorganize the structure of software modules and software libraries. In the literature a number of extensions of the original approach can be found. So, for example, multi-valued concept analysis where the value of features is not restricted to two values (true and false). Also more involved models have been proposed taking into account additional aspects of knowledge representation such as different sources of data or the inclusion of rule-based knowledge in the form of ontologies.

Being basically an application of lattice theory FCA is a well-suited topic for machine-oriented formalization. On the one hand it allows to investigate the possibilities of reusing an already formalized lattice theory. On the other hand it can be the starting point for the formalization of the extensions mentioned above. In the following we briefly present the Mizar formalization of the basic FCA notions. The starting point is a formal context giving the objects and attributes of concern. Formally such a context consists of two sets of objects O and attributes A, respectively. Objects and attributes are connected by an incidence relation $I \subseteq O \times A$. The intension is that object $o \in O$ has property $a \in A$ if and only if $(o, a) \in I$. In Mizar [23] this has been modelled by the following structure definitions.

```
definition
  struct 2-sorted (# Objects, Attributes -> set #);
end;

definition
  struct (2-sorted) ContextStr
    (# Objects, Attributes -> set,
       Information -> Relation of the Objects,the Attributes #);
end;
```

Now a formal context is a non-empty `ContextStr`. To define formal concepts in a given formal context C two derivation operators `ObjectDerivation(C)` and `AttributeDerivation(C)` are used. For a set O of objects (A of attributes) the derived set consists of all attributes a (objects o) such that $(o, a) \in I$ for all $o \in O$ (for all $a \in A$). The Mizar definition of these operators is straightforward and omitted here.

A formal concept FC is a pair (O, A) where O and A respect the derivation operators: the derivation of O contains exactly the attributes of A, and vice versa. O is called the extent of FC, A the intent of FC. In Mizar this gives rise to a structure introducing the `extent` and the `intent` and an attribute `concept-like`.

```
definition let C be 2-sorted;
  struct ConceptStr over C
    (# Extent -> Subset of the Objects of C,
       Intent -> Subset of the Attributes of C #);
end;

definition let C be FormalContext;
           let CP be ConceptStr over C;
  attr CP is concept-like means  :: CONLAT_1:def 13
    (ObjectDerivation(C)).(the Extent of CP) = the Intent of CP &
    (AttributeDerivation(C)).(the Intent of CP) = the Extent of CP;
end;

definition let C be FormalContext;
  mode FormalConcept of C is concept-like non empty ConceptStr over C;
end;
```

Formal concepts over a given formal context can be easily ordered: a formal concept FC_1 is more specialized (and less general) than a formal concept FC_2 iff the extent of FC_1 is included in the extent of FC_2 (or equivalently iff the intent of FC_2 is included in the intent of FC_1). With respect to this order the set of all concepts over a given formal context C forms a complete lattice, the concept lattice of C.

```
theorem
  for C being FormalContext holds ConceptLattice(C) is complete Lattice;
```

This theorem, among others, has been proven in [23]. The formalization of FCA in Mizar went rather smoothly, the main reason being that lattice theory has already been well developed. Given objects, attributes and an incidence relation between them, this data can now be analyzed by inspecting the structure of the (concept) lattice; see [27, 7] for more details and techniques of formal concept analysis.

6 Rough Concept Analysis

In this section we present issues concerning the merging of concrete theories in the Mizar system. We will illustrate them by living examples from Rough Concept Analysis done in Mizar and skipping most technical details (this part is an extension of [12]). For details of used type system, see [11, 2]. We like to mention that in the course of FCA formalization the formal apparatus yet existing in the Mizar Mathematical Library also had to be improved and cleaned up.

A basic structure for the merged theory should inherit fields from its ancestors, which would be hard to implement if structures were implemented as ordered tuples (multiple copies of the same selector, inadequate ordering of fields in the result). The more feasible realization is by partial functions rather, and that is the way Mizar structures work.

```
definition
  struct (ContextStr, RelStr) RoughContextStr
    (# carrier, carrier2 -> set,
       Information -> Relation of the carrier, the carrier2,
       InternalRel -> Relation of the carrier #);
end;
```

As it often happens, an extension of the theory to another need not be unique. There are at least three different methods of adding roughness to formal concepts [15, 22]. The question which approach to choose depends on the author. The notion of a free structure in a class of descendant type conservative with respect to the original object is very useful.

```
definition let C be ContextStr;
  mode RoughExtension of C -> RoughContextStr means
    the ContextStr of it = the ContextStr of C;
end;
```

Now, if C is a given context, we can introduce roughness in many different ways by adjectives.

Up to now, we described only mechanisms of independent inheritance of notions. Within the merged theory it is necessary to define connections between its source ingredients. Here the attributes describing mutual interferences between selectors from originally disjoint theories proved their enormous value. They may determine the set of properties of a free extension.

```
definition let C be RoughFormalContext;
  attr C is naturally_ordered means
    for x, y being Element of C holds
    [x,y] in the InternalRel of C iff
      (ObjectDerivation C).{x} = (ObjectDerivation C).{y};
end;
```

Since the relation from the definiens above is an equivalence relation on the objects of C and hence determines a partition of the set of objects of C into the so-called *elementary sets*, it is a constructor of an approximation space induced by given formal context.

Theory merging makes no sense, if proving the same theorem would be necessary within both source and target theory. Since a new Mizar type called `RoughFormalContext` is defined analogously to the notion of `FormalContext`, as `non quasi-empty RoughContextStr`, the following Fundamental Theorem of RCA is justified only by the Fundamental Theorem of FCA. Even more, clusters providing automatic acceptance of the original theorems do it analogously within target theory. That is also a workplace for clusters `rough` and `exact` from the core rough set theory.

```
for C being RoughFormalContext holds
  ConceptLattice(C) is complete Lattice by CONLAT_1:48;
```

7 Topological Spaces and Partitions

Of course, there are cases we shouldn't even change the language when switching between various fields of mathematics. An illustrative example here is again the notion of rough sets in its primal setting. When we see at the approximation space given by an equivalence relation, it is quite natural to consider just classes of abstractions forgetting about original relation. Hence, the lattice of such objects can be defined:

```
definition
  let X be set;
  func EqRelLatt X -> strict Lattice means
:: MSUALG_5:def 2
  the carrier of it = { x where x is Relation of X,X :
    x is Equivalence_Relation of X } &
  for x,y being Equivalence_Relation of X holds
    (the L_meet of it).(x,y) = x /\ y &
    (the L_join of it).(x,y) = x "\/" y;
end;
```

Among many interesting properties which were proven about this structure we can quote its completeness, for example:

```
registration
  let A be set;
  cluster EqRelLATT A -> complete;
end;
```

The natural definition of the topological space is that we have a family of open sets called the topology, τ. Then a topological space can be considered as a pair consisting of the universe X and the topology τ defined on the subsets of X if τ satisfies the axioms of topology. As they are widely known, informally, we quote below only a formal counterpart of it:

```
definition
  struct (1-sorted) TopStruct
    (# carrier -> set,
      topology -> Subset-Family of the carrier
     #);
end;
```

reflecting the bare $\langle X, \tau \rangle$ tuple and

```
definition
  let IT be TopStruct;
  attr IT is TopSpace-like means
:: PRE_TOPC:def 1
  the carrier of IT in the topology of IT &
  (for a being Subset-Family of IT st a c= the topology of IT holds
    union a in the topology of IT) &
   for a,b being Subset of IT st
     a in the topology of IT & b in the topology of IT holds
      a /\ b in the topology of IT;
end;
```

as axiomatic description of τ.

Then a topological space is just the structure `TopStruct` to which the adjective `TopSpace-like` can be added. As usual, with every such object we can associate the closure and the interior operators, with axioms in Kuratowski style and then the existing apparatus of topological spaces (`Cl` and `Int` for the closure and interior, respectively) can be reused.

8 Conclusions

Even if we are aware that this paper is really an emerging work and most technicalities were really skipped (but they can of course tracked in corresponding Mizar source files freely available from the project homepage), there are some its clear advantages – considering the repository of formalized mathematical knowledge as a whole extends our knowledge. Some of the ideas contained in this paper

are dated back to 2004 and our paper [12] presented at the International Conference in Mathematical Knowledge Management where some of the problems were only identified, but until now many new tools were developed and many interesting new topics were formalized.

Quoting Pawlak's own words about the role of computers (or *mathematical machines* as they were called):

> *"One can formulate a risky opinion that almost all contemporary mathematical theories in their current state cannot be automatically treated. Reformulating them is not an easy task. So, the question arises, to which extent the amount of work done can be justified by the importance of obtained results. (...) Automated discovery of new important results seems to us rather unlikely."*

Even if Pawlak's doubts about finding new theorems were clearly expressed, he was convinced that computers can help in a bit different way:

> *"(...) the view for theories which are already known, but from another viewpoint can shed some new light for the structure of mathematical theories and improve human creativity."*
> ([18], p. 142, translation ours).

We try to argue that the formalization (still having in mind the discussion on the (over)use of the word 'formal' from the end of the third section) of knowledge in the way accessible by computers is not the question of the sense; it is the question of time. Real efficiency of this activity will be shown by much more examples, much more work, and definitely by much more automation many proof assistants offer. We implemented in Mizar already three paths of rough set theory merging: with topology, formal concepts and lattices (including interval sets, which is formalized in [10]). Hence preliminary steps were already done and as this work makes no sense in the island of isolated knowledge, anyone is invited to contribute.

References

1. G. Bancerek, Development of the theory of continuous lattices in Mizar, in: M. Kerber and M. Kohlhase (eds.), *The Calculemus-2000 Symposium Proceedings,* pp. 65–80, 2001.
2. G. Bancerek, On the structure of Mizar types, in: H. Geuvers and F. Kamareddine (eds.), *Proc. of MLC 2003, ENTCS* 85(7), 2003.
3. B. Buchberger, Mathematical Knowledge Management in Theorema, in: B. Buchberger and O. Caprotti (eds.), *Proc. of MKM 2001,* Linz, Austria, 2001.
4. L. Cruz-Filipe, H. Geuvers, and F. Wiedijk, C-CoRN, the Constructive Coq Repository at Nijmegen, `http://www.cs.kun.nl/~freek/notes/`.
5. D. Dubois, H. Prade, Rough fuzzy sets and fuzzy rough sets, *International Journal of General Systems,* 17(2–3), 191–209, 1990.
6. W. Farmer, J. Guttman, and F. Thayer, Little theories, in: D. Kapur (ed.), *Automated Deduction – CADE-11, LNCS* 607, pp. 567–581, 1992.

7. B. Ganter and R. Wille, *Formal concept analysis – mathematical foundations,* Springer Verlag, 1998.
8. A. Grabowski, Basic properties of rough sets and rough membership function, *Formalized Mathematics,* 12(1), 21–28, 2004; can be tracked also under `http://mizar.org/version/current/html/roughs_1.html`.
9. A. Grabowski, Automated discovery of properties of rough sets, to appear in *Fundamenta Informaticae,* 2013.
10. A. Grabowski, M. Jastrzębska, On the lattice of intervals and rough sets, *Formalized Mathematics,* 17(4), 237–244, 2009.
11. A. Grabowski, A. Korniłowicz, A. Naumowicz, Mizar in a nutshell, *Journal of Formalized Reasoning,* 3(2), 153–245, 2010.
12. A. Grabowski, Ch. Schwarzweller, Rough Concept Analysis – theory development in the Mizar system, MKM 2004 Proceedings, LNCS, 3119, pp. 130–144, 2004.
13. A. Grabowski, Ch. Schwarzweller, Towards automatically categorizing mathematical knowledge, M. Ganzha, L. Maciaszek, and M. Paprzycki (Eds.), FedCSIS 2012 Proceedings, 63–68, 2012.
14. J. Järvinen, Approximations and rough sets based on tolerances, in: W. Ziarko and Y. Yao (eds.), *Proc. of RSCTC 2000, LNAI* 2005, pp. 182–189, 2001.
15. R.E. Kent, Rough Concept Analysis: a synthesis of rough sets and formal concept analysis, *Fundamenta Informaticae* 27(2–3), pp. 169–181, 1996.
16. T. Nipkow, L. Paulson, and M. Wenzel, Isabelle/HOL – a proof assistant for higher-order logic, *LNCS* 2283, 2002.
17. S. Owre and N. Shankar, Theory interpretations in PVS, Technical Report, NASA/CR-2001-211024, 2001.
18. Z. Pawlak: *Automatyczne dowodzenie twierdzeń,* Warsaw, PZWS, 1965 (Eng. *Automated theorem proving,* in Polish).
19. Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning about Data,* Kluwer, Dordrecht, 1991.
20. K. Raczkowski and P. Sadowski, Equivalence relations and classes of abstraction, *Formalized Mathematics,* 1(3), pp. 441–444, 1990.
21. P. Rudnicki and A. Trybulec, Mathematical Knowledge Management in Mizar, in: B. Buchberger and O. Caprotti (eds.), *Proc. of MKM 2001,* Linz, Austria, 2001.
22. J. Saquer and J.S. Deogun, Concept approximations based on rough sets and similarity measures, *International Journal on Applications of Mathematics in Computer Science,* 11(3), pp. 655–674, 2001.
23. C. Schwarzweller, Introduction to concept lattices, *Formalized Mathematics,* 7(2), pp. 233–242, 1998.
24. M. Strecker, Formal verification of a Java compiler in Isabelle, *Lecture Notes in Computer Science,* 2392, 63–77, 2002.
25. J. Urban, G. Sutcliffe, Automated reasoning and presentation support for formalizing mathematics in Mizar, *Lecture Notes in Computer Science,* 6167, 132–146, 2010.
26. F. Wiedijk, Formal proof – getting started, *Notices of the American Mathematical Society,* 55(11), 1408–1414, 2008.
27. R. Wille, Restructuring lattice theory: an approach based on hierarchies of concepts, in: I. Rival (ed.), *Ordered Sets,* Reidel, Dordrecht-Boston, 1982.
28. Y.Y. Yao, A comparative study of fuzzy sets and rough sets, *Information Sciences,* 109(1-4), 227–242, 1998.