Position papers of the Federated Conference on Computer Science and Information Systems, pp. 11–18
DOI: 10.15439/2017F553
ISSN 2300-5963 ACSIS, Vol. 12

Topological Structures as a Tool for Formal Modelling of Rough Sets

Adam Grabowski
Institute of Informatics
Department of Mathematics and Informatics
University of Białystok,
Konstantego Ciołkowskiego 1 M, 15-245 Białystok, Poland
Email: adam@math.uwb.edu.pl

Roland Coghetto
Rue de la Brasserie, 5
B-7100 La Louviére
Belgium
Email: roland_coghetto@hotmail.com

***Abstract*—In the paper, we present the topological counterpart supporting rich formal apparatus describing properties of rough sets within one of the largest repositories of computerized mathematical knowledge, the Mizar Mathematical Library. The paper develops third and final (after lattice theory and the theory of general binary relations) planned path designed to be linked (via mechanisms of theory merging) with the theory of structures described by Pawlak in the early seventies of the previous century. We propose the revision of the existing topological apparatus offered by Mizar, and give the outline of the formalization of uniform spaces, important objects allowing for further representation of approximation spaces.**

I. Introduction

GROWING popularity of computerized mathematical proof-assistants (Voevodsky who won the Fields medal in 2002 underlines the future of computer approach building new foundations of mathematics – univalent foundations) raises a number of new problems which should be solved in order to meet expectations of researchers. It is important that the formal approach should be flexible enough to be easily translated to writing, easily understood by people, and allow for further generalization. In recent years, traditional model of printed contribution fixed for years could be adjusted to take into account the possibilities given by contemporary media where such knowledge is stored.

We focus on the area where mathematical structures can be extensively used and their formal counterpart can be tuned accordingly. The examples were already formalized within machine-verified mathematical knowledge repository: we mean topological spaces certified with the help of the Mizar system [1].

The problem was translating these objects expressed in the natural language used by mathematicians into the formal language of Mizar. These topics are quite well represented in the Mizar Mathematical Library [15], and look promising for the mathematics as a whole – topology delivers tools for representing many other areas of mathematics (with Stone's representation theorem at the very beginning).

II. The Mizar System

The main aim of the Mizar system – the project steered by Andrzej Trybulec from early seventies of the previous century – was to develop a formal approach to mathematics which allows for faithful encoding of the definitions and theorems written in natural language in order to be verified for correctness by computers. This formal approach should be flexible enough to be understood by ordinary people without much pain, so one of the very basic points was to be as close as possible to mathematical vernacular. On the other hand one should have in mind the strictness and the relative simplicity of the grammar of the artificial language in order to be easily scanned by the parser of the Mizar system.

The second ingredient of the system is the repository of formal texts. The Mizar Mathematical Library (MML) [27] is based on Tarski-Grothendieck set theory, which is very close to the one used by the majority of mathematicians [30]. Hence it is not very strange that general topology is one of the widely represented parts of mathematics within this repository of knowledge (see Table I for details, general topology holds fifth position w.r.t. the number of lines of code implemented, but taking into account the number of Mizar articles is just third). Among the large formalization projects of the Mizar community, two were connected with topology. The first one was the formalization of Jordan curve theorem, resulting in many articles written in tight cooperation with Japanese Mizar group (the high position held by algebraic topology – AMS MSC 2010 category started with 14 is a result of this development). The second one, the formalization of the *Compendium of Continuous Lattices* by Gierz et al. [7], although originally meant to be placed within lattice theory, eventually was driven into the direction of category theory and topology. It was quite a lucky coincidence for us as we the first author was involved also in the part dealing with the properties of Scott-continuous functions. It should be mentioned that a few well-defined topological notions, as, for example, Aleksandrov topologies, obtained a new life just with the connection with continuous domains. Another formalization project, relatively recent one, was to formalize Engelking's *General Topology* [6], but as of now, the project seems to be not very dynamic.

Original motivation for our paper was to describe some of the issues raised in the process of formalizing important mathematical structures – topological spaces, connected with the theory of tolerance approximation spaces [10]. We realized that in order to do this properly (at least to use as much expressive power of the Mizar language as we can), we should lift both notions into the common ground – of the descendant of topological spaces merged with approximation spaces. We have observed that developing alternative background for already well-established area of formalized knowledge can cause many troubles. This paper is a contribution to the third large area of mathematics with which rough sets are strongly linked, with another two already formalized: lattice theory [13], and general theory of binary relations [35]. Unfortunately, modal logics are not a sufficiently developed area within the Mizar Mathematical Library, and we do not expect any significant future progress in this topic.

III. Topological Preliminaries

A topological space is a pair $(U, \mathcal{T})$ consisting of a set U and family $\mathcal{T}$ of subsets of U satisfying the following conditions:

- $\emptyset \in \mathcal{T}$ and $U \in \mathcal{T}$;
- $\mathcal{T}$ is closed under finite intersections, i.e., for all $A, B \in \mathcal{T}$ we have $A \cap B \in \mathcal{T}$;
- $\mathcal{T}$ is closed under arbitrary unions.

Let D be a partition of U. The collection of sets that can be written as union of some members of D together with the empty set is a topology for U – the partition topology generated by D. Obviously, every equivalence relation E generates a partition of U, namely U/E, hence it is connected with underlying topology on U. Such partition topology is usually denoted by τ_E, or just τ for fixed equivalence relation E (which is exactly the case, if we work in a given approximation space, and none other indiscernibilities are considered).

The partition topologies are characterized by the fact that every open set is also closed; every partition topology is an Alexandrov topology, in which the intersection of the members of each, not necessarily finite, collection of open sets is also open.

Let T be a tolerance relation in U and let E_T be the intersection of all equivalence relations in U that include T (extensions of T). It can be shown that E_T is again an equivalence relation, and the collection of T-definable sets is precisely the collection of E_T-definable sets. Hence, for tolerance relations T, the collection of T-definable sets is a partition topology. Essentially, the linking between an approximation space (U, E) and corresponding topological space (U, τ_E) can be established: X is definable if and only if X is open (or, respectively, closed) in the partition topology; the lower approximation of X is just the interior of X and the upper approximation of X – the closure of X. Hence X is definable if and only if its interior is equal to its closure.

The characterization of rough approximations can be also given in terms of maps between powersets of the universe U, and this was really the idea of Hammer [20]. For a binary relation R in U, the function

$$X \mapsto \{y \in U : (x, y) \in R \text{ for some } x \in X\}$$

is a mapping from 2^U into itself. Consequently then, similarly to Zhu [35], we can study the properties of approximations just by studying the properties of set-valued set-functions. In fact, the paper by Zhu [35] was fully translated into Mizar formalism and the details are to be presented at IJCRS 2017 [14].

For equivalence relation E on U a uniformity for U is defined as the collection ρ of subsets of U^2 in a following way:

$$\rho = \{R : R \subseteq U^2, E \subseteq R\}.$$

The topology for U induced by this uniformity coincides with topology τ_E. The connections between rough sets and uniform spaces [32] are as follows: Pawlak's approximation spaces are uniform spaces whose uniform topologies coincide with partition topologies; these topologies can be characterized by the fact that every open set is also closed, and hence, they are Aleksandrov topologies.

The relationship between the theory of rough sets and the theory of topological spaces is as follows: if the underlying indiscernibility relation is an equivalence relation, then the collection of definable sets is a uniformity whose topology is a partition topology (every open set is also closed and vice versa); if we deal with a tolerance relation, the collection of definable sets is a quasiuniformity whose topology is also a partition topology; if the underlying indiscernibility relation is a preorder, the collection of definable sets is a topology, but not necessarily a partition topology. In all cases however, we deal with an Alexandrov topology (arbitrary intersection of definable sets is a definable set).

IV. Towards Algebraic Hierarchy

All algebraic structures in Mizar are defined in similar manner: first we have to give a structure, where names of fields (called selectors) with their specification (the type and the arity) are given. In our concrete case there were

```
definition
  struct (1-sorted) addMagma
    (# carrier -> set,
          addF -> BinOp of the carrier #);
end;
```

and

```
definition
  struct (ZeroStr,addMagma) addLoopStr
  (# carrier -> set,
        addF -> BinOp of the carrier,
       ZeroF -> Element of the carrier #);
end;
```

Structures in Mizar can be used to model mathematical notions like groups, topological spaces, categories, etc. which are usually represented as tuples. A structure definition contains, therefore, a list of *selectors* to denote its fields, characterized by their name and type, e.g.:

```
definition
  struct multMagma
  (# carrier -> set,
```

TABLE I
TOP 10 DEVELOPED THEORIES IN MML BY AMS MSC 2010

No.	MSC	Topic	Number of articles	Lines of code
1.	03	Mathematical logic and foundations	146	311,083
2.	14	Algebraic geometry	84	251,809
3.	06	Order, lattices, ordered algebraic structures	110	234,413
4.	26	Real functions	91	225,634
5.	54	General topology	99	196,486
6.	68	Computer science	83	193,782
7.	11	Number theory	72	154,307
8.	15	Linear and multilinear algebra; matrix theory	61	149,941
9.	46	Functional analysis	69	132,741
10.	57	Manifolds and cell complexes	42	122,738

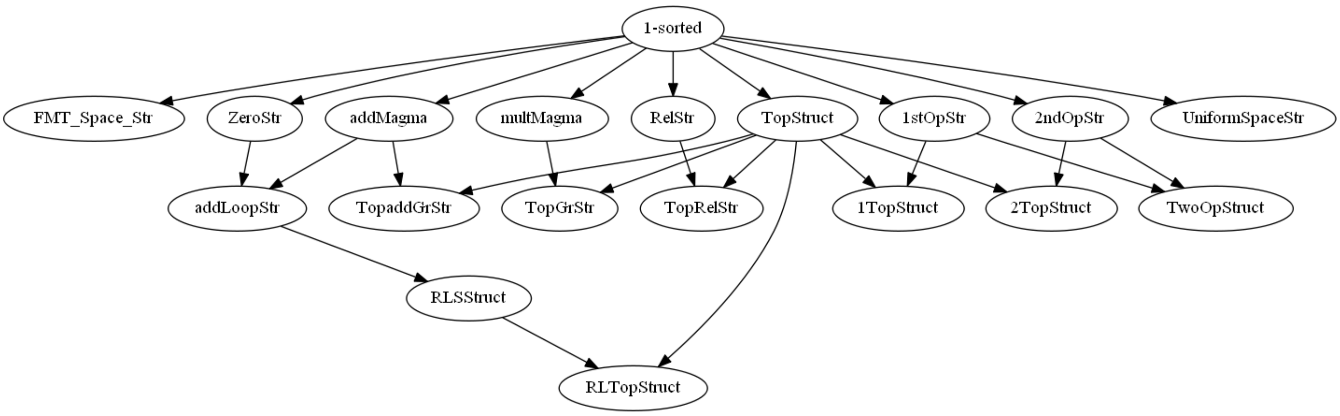

Fig. 1. The net of topological structures in MML

```
    multF -> BinOp of the carrier #);
end;
```

where `multMagma` is the name of a structure with two selectors: an arbitrary set called its `carrier` and a binary operation on it, called `multF`. This structure can be used to define a group, but also upper and lower semilattices, so in fact any notion that is based on a set and a binary operation on it. It should be noted that the above structure does not define a group yet (nor any other more concrete object), because there is no information on the properties of `multF`. The structure is just a basis for developing a theory. In practice, after introducing a required structure, a series of attributes is also defined to describe the properties of certain fields.

As mentioned before, the above `multMagma` structure can be used to define notions which are not only groups. Still, the operation in such structures inherit the name `multF`, because the current Mizar implementation does not provide a mechanism to introduce synonyms for selectors (or whole structures). Therefore, in cases when a different name is frequently used in standard mathematical practice, it may be better to introduce a different structure. For example, lattice operations are commonly called `meet` and `join`, so a lower semilattice may be better encoded as:

```
definition
  struct /\-SemiLattStr
  (# carrier -> set,
     L_meet -> BinOp of the carrier #);
end;
```

Mizar supports multiple inheritance of structures that makes a whole hierarchy of interrelated structures available in the Mizar library, with the `1-sorted` structure being the common ancestor of almost all other structures. For example, formalizing topological groups in Mizar can be done by independently defining and developing group theory and the theory of topological spaces, and then merging these two theories together based on a new structure, e.g.:

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

definition
  struct (multMagma, TopStruct) TopGrStr
  (# carrier -> set,
      multF -> BinOp of the carrier,
   topology -> Subset-Family
                 of the carrier #);
end;
```

The advantage of this approach is that all notions and facts concerning groups and topological spaces are naturally applicable to topological groups. Let us note that when introducing a new structure, the inherited selectors can be listed in any order, as far as relations between them are preserved. The list of names of ancestor structures is put in brackets before the name of the structure being defined. Figure 1 shows only the part of the net of all over 150 structures defined in MML which are used for formalizing topology: we can find there topological groups, topological relational structures, or real linear spaces equipped with a topology, just to mention a few important ones. Structures `RelStr` and `TopStruct` are in the middle as the most important ones, and crucial in the formalization of CCL. The right hand side of the diagram was recently fully developed by the authors; it is useful both for alternative formal approach to topological spaces which which will be shown in Section VII and the theory of uniform spaces given by the second author.

Concrete mathematical objects, e.g. the additive group of integers are introduced with so called *aggregates* – special term constructors defined automatically by the definition of a structure, e.g.: `multMagma(#INT,addint#)`, where `INT` is the set of integers, and `addint` represents the addition function. It is necessary that all terms used in the aggregate have the respective types declared in the structure's definition. In our example `INT` is obviously a set, and `addint` must be of type `BinOp of INT`.

Every structure defines implicitly a special attribute, strict. The corresponding adjective's meaning is that an object of a structure type contains nothing more, but the fields defined for that structure. For example, a term with structural type based on `TopGrStr` may be strict `TopGrStr`, but it is neither strict `multMagma`, nor strict `TopStruct`. Clearly, every term constructed using a structure's aggregate is strict.

Finally, the Mizar language has means to restrict a given term with a complex structure type to its well-defined subtype. This special term constructor, the *forgetful functor* also utilizes the structure's name, e.g. `the multMagma of G`, where `G` has a potentially wider type which inherits the `multMagma` structure. Again, such terms are strict, with respect to the given structure type. The (part of) hierarchy of algebraic structures deliver only a signature for corresponding algebras; the real semantics is given by axioms. In Mizar formalism, axioms are defined as adjectives (called also attributes). The details of the algebraic hierarchy in the Mizar Mathematical Library were presented at FedCSIS conference last year [17].

V. Topology Formalized

In this section we will describe the existing current definition of topological spaces within MML. Following Engelking [6], we can choose open sets as the basic notion and so it was decided to be the base in the MML: we have a structure of a topological space together with the only adjective of which name suggests its technical character. We can originally choose between point-free topology and that with points; in MML we deal with the earlier approach. Obviously, the backbone corresponding structure is `TopStruct` given in Section IV. Similarly, as in the algebraic case, structures can be understood as partial functions on the selectors (in the abovementioned example, the carrier which is a set on which a topology can be defined, and the topology, i.e. the family of open sets). But the real properties of the topology (both $\emptyset$ and the whole universe should be open; the family should be closed for finite intersections and arbitrary unions) is given by the Mizar attribute which is in fact an adjective (`TopSpace-like`).

```
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;
```

Making appropriate hierarchy for well-established notions is really crucial for the repository of formal texts; if we are interested only in pure predicates and computer-generated proofs, readability is something which does not really matters (and this is the case of the part of Isabelle's Archive of Formal Proofs [3] devoted to software verification), however from a viewpoint of reusability of adjectives, when large databases are involved, this is a question of efficiency. As a simple nontrivial example, we can mention the net of cross-linked properties of rough approximation operators under various conditions as reflexivity, symmetry, transitivity – as canonical examples, but also with seriality, positive and negative alliance as less straightforward ones.

We can see that essentially the whole series of Mizar articles dealing with topology uses more or less the type defined as the structure with the single adjective as described in this section – the Mizar mode `TopSpace` is not very convenient starting point for further generalizations. One can notice that we do not need in the abovementioned definition the assumption that the empty set is an element of the topology: the union of $\emptyset$ is just $\emptyset$, and the thesis is trivial as any topology is closed under arbitrary unions. Bourbakists define topological spaces just by means of finite intersections and arbitrary unions, but one the other hand the set $\bigcap \emptyset$ is not well-defined in Zermelo-Fraenkel set theory.

We can see a topological operator either from the view of Mizar functors, as it can be recognized now as a base; as they are typed, we can read that the closure of an arbitrary subset of given topological space T is again the subset of T. But alternatively, we can use another way around: first we can define a function which returns the closure for arbitrary argument. Of course, one should define for such a map the domain and the range properly; in our specific case this could be a (total) function defined on the boolean of the carrier of T. Among various approaches to topological spaces the two are especially important: the first one deals with the family of subsets of a given universe possessing certain properties; the other deals with closure operators in sense of Kuratowski.

```
definition
  let T be TopStruct,
```

```
      P be Subset of T;
  attr P is open means
:: PRE_TOPC:def 2
  P in the topology of T;
end;
```

Closed sets are precisely those, of which complements are open; similarly the closure of given subset A can be defined just as the minimal closed set containing A.

```
definition
  let GX be TopStruct, A be Subset of GX;
  func Cl A -> Subset of GX means
:: PRE_TOPC:def 7
  for p being set st p in the carrier of GX
      holds
    p in it iff for G being Subset of GX st
      G is open holds p in G
  implies A meets G;
end;
```

Of course, the above is definitely not the only possible definition – we can define the closure as the intersection of all closed supersets of A, but the obvious and important connection between the closures and closed sets is that closed subsets are fixed points with respect to the closure operators.

```
theorem :: PRE_TOPC:22
  for A being Subset of T holds
    (A is closed implies Cl A = A) &
    (T is TopSpace-like & Cl A = A implies
      A is closed);
```

As a consequence, the above theorem can be considered as an equivalent definition of a closed set as the fixed point under closure operator; this will be explained from another viewpoint (and reused) later.

We can mention here the outline of the formalization of the common generalization of topological groups and metric spaces. Uniform spaces, which are credited to Weil [33] and more systematic formal approach – to the group of Bourbakists (which is quite nice coincidence as the Mizar project implements main postulates of formalization of mathematics which were fundamental to Bourbaki group), appeared to be a useful framework explaining the concept of rough sets in terms of both equivalence and tolerance relations. Formally, uniform spaces are based on Mizar structures

```
definition
  struct (1-sorted)
     UniformSpaceStr
          (# carrier -> set,
          entourages -> Subset-Family of
            [:the carrier,the carrier:] #);
end;
```

where French *entourages* means surroundings. The real topological flavour of these pretty general constructions is given by defining an open subset O of X if and only if for every $x \in O$ there exists an entourage V such that $V[x]$ is a subset of O. For more details of fundamental systems of entourages treated formally, we refer to [4] and [5] containing thorough encoding of the theory – almost 7 thousand lines of code, i.e. about 90 pages of formal definitons, theorems, and proofs.

The essential notion is the uniformity induced by the general binary relation

```
definition
  let X be set,
      R be Relation of X;
  func uniformity_induced_by(R) ->
    upper cap-closed strict UniformSpaceStr
      equals
:: UNIFORM3:def 21
  UniformSpaceStr (# X,rho(R) #);
end;
```

where `rho` is just ρ as described in Section III. Adding underlying properties to a binary relation, it turns out that we obtain axioms defining basic classes of (semi-)uniform spaces. The full connection between theory of uniform spaces and rough sets is expressed in two important corollaries:

```
definition
  let X be set,
      R be Tolerance of X;
  redefine func uniformity_induced_by(R)
    -> strict Semi-UniformSpace;
end;

theorem :: UNIFORM3:51
  for X being set,
       R being Equivalence_Relation of X
    holds
    uniformity_induced_by(R) is UniformSpace;
```

Even if usually uniform spaces are meant to be topological spaces with additional structure, this extension is absent in the above definition, as this time we presented purely topological properties in terms of Mizar adjectives (instead of fixed `topology` we use appropriate notions in terms of entourages, which is not very strange, as we can use the notion of a neighbourhood).

VI. The Issue of Equivalent Characterizations

In mathematics we often experience the situation when we have equivalent sets of axioms for the same mathematical object. The motivation of using them both in the same time can be manyfold: either the approaches were developed in as sense independently, without knowing each other, and after that they were proved to be equivalent definitions of the same notion, or just the newly proposed set is preferred because of its simplicity or usefulness. Such considerations are especially often in lattice theory, where we deal with the fixed set of operations (as the supremum, the infimum and the complementation). The situation gets slightly more complicated if the collections of operations are distinct. Of course, the canonical example here is delivered again in the world of lattices, where we have, among the ordinary binary operations $\sqcup$ and $\sqcap$ (or, to be more precise, instead of them at first) the ordering relation $\leq$. In this case, the original idea to show the correspondence was to define two Mizar functors transforming posets into lattices [12], [18], and vice versa.

When we consider things informally, it is enough to have such construction; but then, we cannot be in these two universes in the same time and we have to choose only a single framework to work with (and `redefine` construction really supports such approach). Some time ago, as a part of the formalization of Jordan curve theory, we did similar work: essentially we have shown that the notion of an open set defined for subset of the set of real numbers coincides with that of an open set in the natural topology of the real line. Of course, having basic properties proven in both settings is important, but soon we should face the problem of how much theory to be developed in parallel.

As an interesting direction of research in the area of topology [8] we can point out the beginnings of the so-called theory of finite topological spaces as defined by Imura and Eguchi in [22]. Based on relational structures, the authors define new operator which is just the set of all elements of the universe which are in the internal relation with the given point.

```
definition
  let FT be RelStr;
  let x be Element of FT;
  func U_FT x -> Subset of FT equals
:: FIN_TOPO:def 1
  Class (the InternalRel of FT,x);
end;
```

The Mizar functor `Class` meant originally the class of abstraction w.r.t. the given equivalence relation. In the process of generalizing notions all underlying attributes were removed from the assumptions of this definition, but the name remains the same. One of the basic properties of neighbourhoods states that any point should be a member of its neighbourhood. Although the above definition does not need any additional assumptions, now we have to add a variant of reflexivity of the relational structure, with the new synonimical name, `filled`. Of course, having just a new name for the old notion does not bring too much additional information; but now we can express the reflexivity in terms of neighbourhoods.

```
definition
  let IT be non empty RelStr;
  redefine attr IT is filled means
:: FIN_TOPO:def 4
  for x being Element of IT holds x in U_FT x;
end;
```

The series started with [22] is not really exhaustive; but the connections with other areas of mathematics are obvious.

VII. The New Approach

The first step in our proposed approach was to have the new naming scheme. We decided to use again a postfix `-like` to suggest that if a family of subsets satisfies the conjunction of properties, it can be treated as the family of open sets (i.e. it is a topology).

```
definition
  let X be set;
  let F be Subset-Family of X;
  attr F is topology-like means
    {} in F & X in F &
    F is union-closed cap-closed;
end;
```

Later, such adjectives were meant to be replaced by more selfexplaining names. But in fact, the first conjunct is just the negation of already present in MML `with_non-empty_elements`, and the second one can be named as `with_universe` or something similar. Observe that there are two main differences between the definition from Section V (`TopSpace-like`) and the current one. The first one is that the latter is on the concrete level, i.e. it does not use the notion of the structure. Of course, it is easy to lift such definition to the abstract (i.e. structural) level: one can define appropriate field to have such properties. The second difference is that the old one is the conjunction of three instead of four adjectives, as one of them can be deduced from the combination of remaining ones and in this sense the approach proposed here is similar to the one developed in the case of σ-fields of subsets. In such a manner, we deal with Čech preclosure and Kuratowski closure operators, respectively.

```
definition
  let X be set,
      O be Function of bool X, bool X;
  attr O is preclosure means
    O is extensive \/-preserving
         empty-preserving;
  attr O is closure means
    O is extensive idempotent
         \/-preserving empty-preserving;
end;
```

The crucial issue here is about the structure on which we can establish the connection between closed sets and fixed points w.r.t. maps. We decided not to use concrete relational structures, but we introduced new structures, `1TopStruct` which are ancestors of topological structures enriched by maps on X, i.e. functions from the set 2^X into itself.

```
theorem :: ROUGHS_4:2
  for T being with_properly_defined_topology
    1TopStruct,
      A being Subset of T holds
    A is op-closed iff A is closed;
```

Under such defined attributes, showing that if the operator satisfies the properties of preclosure, it generates an abstract topological space.

```
registration
  cluster with_preclosure -> TopSpace-like for
    with_properly_defined_topology 1TopStruct;
end;
```

The question of defining the family of open sets (i.e. the most usual definition of topology) might arise; the answer is immediate – as the family of fixed points under the closure operator. So the topology is collected from these objects which are subsets of the considered universe which are `f-closed`, where `f` stands for the map under consideration (an abstract closure operator).

```
definition
  let X be set,
      f be Function of bool X, bool X;
  func GenTop f -> Subset-Family of X means
```

```
:: ROUGHS_4:def 25
    for x being object holds
      x in it iff ex S being Subset of X st
      S = x & S is f-closed;
end;
```

In fact, this is another formulation of the property expressed by the attribute `with_properly_defined_topology`.

```
theorem :: ROUGHS_4:5
  for X being set,
      f being Function of bool X, bool X st
    f is preinterior holds
      GenTop f is topology-like;

registration
  let C be set, I be (Relation of C),
      f be topology-like Subset-Family of C;
  cluster TopRelStr (#C,I,f#) -> TopSpace-like;
end;
```

Finally, composing the above theorem and functor registration, we deduce that if the map which is the field in the merged structure had the properties of preclosure, generated space has all the properties of topological spaces.

VIII. Merging Topologies and Rough Sets

The notion of a rough set was defined by Pawlak [28] to reflect the situation of an incomplete knowledge about the universe of objects. We formalized the notion in Mizar [11] and pretty recently observed that this is almost identical to the approach described in Section VI. Any element of the universe can be viewed through a binary relation which can unify potentially distinct objects if the available information about their properties is the same. Such relation, called indescernibility relation, can possess basic mathematical properties of relations: if we assume R to be reflexive, symmetric, and transitive (so it is an equivalence relation), we have the original approach of Pawlak.

```
definition let T be non empty TopRelStr;
  attr T is naturally_generated means
:: ROUGHS_4:def 28
    the topology of T = GenTop LAp T;
end;

theorem :: ROUGHS_4:10
  for T being naturally_generated
    non empty with_equivalence TopRelStr,
      A being Subset of T holds
    A is closed iff UAp A = A;
```

As both notions coincide (the upper approximation operator in rough sets and closure operator in underlying topological spaces and similarly in the dual case), reusing these areas of mathematics we have obtained concrete results: the characterization of rough sets in terms of Isomichi classification of domains, and the view for rough sets from the viewpoint of Kuratowski closure-complement problem (known also as fourteen sets of Kuratowski) [9].

As always, we can be skeptical about defining the mathematical object as one of the fields in the structure: it could be well illustrated based on the notion of the complementation operator in the lattice structure. On the one hand, it is really natural to have it as a separate field, as it was in case of ortholattices. When it is just a part of the language's signature, it reflects the ordinary mathematical definition [16].

```
registration
  let T be naturally_generated
    non empty with_equivalence TopRelStr,
      A be Subset of T;
  cluster UAp A -> closed;
end;

registration
  let T be with_equivalence
    naturally_generated non empty TopRelStr;
  let A be Subset of T;
  identify Cl A with UAp A;
end;
```

The latter registration would allow for mixed use of the lower approximation instead of interior operator and vice versa. The only drawback of this approach is that to obtain pure context of uniform space (i.e. `strict` topological space or `strict` tolerance approximation space), we have to use Mizar forgetful functor `the`.

The above unification of the world of topological spaces and of rough sets allowed us to fully benefit from the results placed in the area of general topology, previously obtained: we can easily observe the connection of approximation spaces with the classification of domains proposed by Isomichi or the problems of Kuratowski sets, giving the combination of closure, interior, and complementation operators [11], without explicit reference to those theories.

IX. Conclusion and Future Work

In the paper we tried to show how theoretically straightforward examples can lead to difficult problems during their translation from informal presentation in natural human language into formalism of the Mizar language, a variant of mathematical vernacular. Based on the example of topological spaces we could observe that even if the approach is given in a not satisfactory way, it can be corrected in a process of the so-called revision [19]. The part of the work could be less painful – the splitting of the original definition as we proposed and automatic replacement of the references into new ones. The level of generality is obviously higher in our approach, so we hope to open some new paths in the formalization of general topology, especially in more abstract form.

The second part, which could be done gradually and with the possible use of automatic tools, is that this proposed new version should be consumed in the MML – the theorems and definitions which can be formalized in the more general way, should be formulated so. This would also enable reusing purely topological constructions in another areas of mathematics – for example, fourteen Kuratowski sets can be expressed in the language of group theory and abstract maps with accompanying properties. This also opened the way for explaining rough sets in topological terms and will not be restricted for the Mizar library only, as the translation from the Mizar formalism into other formal languages are available [21]. Additionally we hope to unify the existing approach with newly developed theory of uniform spaces.

In the informal form of a mathematical publication written by people in natural language, such process could (and eventually led in real life, as it was in the world of rough sets) to the sequence of papers generalizing the approach gradually. Hence it is also a kind of a problem for repository storing the knowledge. In our case, the Mizar Mathematical Library allows for some automatic enhancements. We removed repetition, compressed the files, and cleared the path to improve the overall algebraic framework available in the Mizar Mathematical Library. Although natural language is rather flexible, we believe that formal counterpart benefits from the relative coherence of the existing approaches.

REFERENCES

[1] Bancerek, G., Byliński, C., Grabowski, A., Korniłowicz, A., Matuszewski, R., Naumowicz, A., Pąk, K., and Urban, J. (2015). Mizar: State-of-the-art and beyond. In Kerber, M., Carette, J., Kaliszyk, C., Rabe, F., and Sorge, V., editors. *Intelligent Computer Mathematics – International Conference, CICM 2015, Washington, DC, USA, July 13–17, 2015, Proceedings*, volume 9150 of *Lecture Notes in Computer Science*. Springer, pages 261–279. doi:10.1007/978-3-319-20615-8_17

[2] Bancerek G, Rudnicki P. (2002). A Compendium of Continuous Lattices in Mizar (formalizing recent mathematics). *Journal of Automated Reasoning,* 29(3/4):189–224. doi:DOI: 10.1023/A:1021966832558

[3] Blanchette, J., Haslbeck, M., Matichuk, D., and Nipkow, T. (2015). Mining the Archive of Formal Proofs, in Kerber, M., editor. *Conference on Intelligent Computer Mathematics (CICM 2015),* Lecture Notes in Computer Science, 9150, pages 3–17, Springer. doi:10.1007/978-3-319-20615-8_1

[4] Coghetto, R. (2016). Quasi-uniform space. *Formalized Mathematics,* 24(3):215–226. doi:10.1515/forma-2016-0017

[5] Coghetto, R. (2016). Uniform space. *Formalized Mathematics,* 24(3):205–214. doi:10.1515/forma-2016-0018

[6] Engelking, R. (1977). *General Topology*, volume 60 of *Monografie Matematyczne*. PWN – Polish Scientific Publishers, Warsaw.

[7] Gierz, G., Hofmann, K., Keimel, K., Lawson, J., Mislove, M., and Scott, D. (1980). *A Compendium of Continuous Lattices*. Springer-Verlag, Berlin, Heidelberg, New York.

[8] Grabowski, A. (2004). Solving two problems in general topology via types. In Filliâtre, J., Paulin-Mohring, C., and Werner, B., editors, *Types for Proofs and Programs, International Workshop, TYPES 2004, Jouy-en-Josas, France, December 15–18, 2004, Revised Selected Papers*, volume 3839 of *Lecture Notes in Computer Science*, pages 138–153. Springer. doi:10.1007/11617990_9

[9] Grabowski, A. (2005). On the computer-assisted reasoning about rough sets. In Dunin-Kęplicz, B., Jankowski, A., Skowron, A., and Szczuka, M., editors, *International Workshop on Monitoring, Security, and Rescue Techniques in Multiagent Systems Location*, volume 28 of *Advances in Soft Computing*, pages 215–226, Berlin, Heidelberg. Springer-Verlag. doi: 10.1007/3-540-32370-8_15

[10] Grabowski, A. (2013). Automated discovery of properties of rough sets. *Fundamenta Informaticae*, 128(1-2):65–79. doi:10.3233/FI-2013-933

[11] Grabowski, A. (2014). Efficient rough set theory merging. *Fundamenta Informaticae*, 135(4):371–385. doi:10.3233/FI-2014-1129

[12] Grabowski, A. (2015). Mechanizing complemented lattices within Mizar type system. *Journal of Automated Reasoning*, 55(3):211–221. doi:10.1007/s10817-015-9333-5

[13] Grabowski, A. (2016). Lattice theory for rough sets – a case study with Mizar, *Fundamenta Informaticae,* 147(2–3):223-240. doi:10.3233/FI-2016-1406

[14] Grabowski, A. (2017). Computer certification of generalized rough sets based on relations. Accepted to International Joint Conference on Rough Sets, IJCRS 2017. doi:10.1007/978-3-319-60837-2_7

[15] Grabowski, A., Korniłowicz, A., and Naumowicz, A. (2015). Four decades of Mizar. *Journal of Automated Reasoning*, 55(3):191–198. doi:10.1007/s10817-015-9345-1

[16] Grabowski, A., Korniłowicz, A., and Schwarzweller, C. (2015). Equality in computer proof-assistants. In Ganzha, M., Maciaszek, L. A., and Paprzycki, M., editors, *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, volume 5 of *Annals of Computer Science and Information Systems*, pages 45–54. IEEE. doi:10.15439/2015F229

[17] Grabowski, A., Korniłowicz, A., Schwarzweller, C. (2016). On algebraic hierarchies in mathematical repository of Mizar. In: M. Ganzha, L.A. Maciaszek, M. Paprzycki, editors, *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, volume 8 of *Annals of Computer Science and Information Systems*, pages 363–371. IEEE. doi:10.15439/2016F520

[18] Grabowski, A. and Moschner, M. (2004). Managing heterogeneous theories within a mathematical knowledge repository. In Asperti, A., Bancerek, G., and Trybulec, A., editors, *Mathematical Knowledge Management, Third International Conference, MKM 2004, Bialowieza, Poland, September 19–21, 2004, Proceedings*, volume 3119 of *Lecture Notes in Computer Science*, pages 116–129. Springer. doi:10.1007/978-3-540-27818-4_9

[19] Grabowski, A. and Schwarzweller, C. (2007). Revisions as an essential tool to maintain mathematical repositories. In *Proceedings of the 14th Symposium on Towards Mechanized Mathematical Assistants: 6th International Conference*, Calculemus '07 / MKM '07, pages 235–249, Berlin, Heidelberg. Springer-Verlag. doi:10.1007/978-3-540-73086-6_20

[20] Hammer, P.C. (1963). Extended topology: the continuity concept. *Mathematics Magazine,* 36(2):101–105.

[21] Iancu, M., Kohlhase, M., Rabe, F., and Urban, J. (2013). The Mizar Mathematical Library in OMDoc: Translation and applications. *Journal of Automated Reasoning*, 50(2):191–202. doi:10.1007/s10817-012-9271-4

[22] Imura, H. and Eguchi, M. (1992). Finite topological spaces. *Formalized Mathematics*, 3(**2**):189–193.

[23] Järvinen, J. (2007). Lattice theory for rough sets, *Transactions on Rough Sets VI,* Lecture Notes in Computer Science (LNAI) 4374:400–498. doi:10.1007/978-3-540-71200-8_22

[24] Korniłowicz, A. (2015). Definitional expansions in Mizar. *Journal of Automated Reasoning*, 55(3):257–268. doi:10.1007/s10817-015-9331-7

[25] Korniłowicz, A. (2015). Flexary connectives in Mizar. *Computer Languages, Systems & Structures*, 44:238–250. doi:10.1016/j.cl.2015.07.002

[26] Naumowicz, A. (2015). Automating Boolean set operations in Mizar proof checking with the aid of an external SAT solver. *Journal of Automated Reasoning*, 55(3):285–294. doi:10.1007/s10817-015-9332-6

[27] Naumowicz, A. (2015). Tools for MML environment analysis. In Kerber, M., Carette, J., Kaliszyk, C., Rabe, F., and Sorge, V., editors (2015). *Intelligent Computer Mathematics – International Conference, CICM 2015, Washington, DC, USA, July 13–17, 2015, Proceedings*, volume 9150 of *Lecture Notes in Computer Science*. Springer, pages 348–352. doi:10.1007/978-3-319-20615-8_26

[28] Pawlak, Z. (1982). Rough sets. *International Journal of Parallel Programming*, 11:341–356.

[29] Pąk, K. (2015). Improving legibility of formal proofs based on the close reference principle is NP-hard. *Journal of Automated Reasoning*, 55(3):295–306. doi:10.1007/s10817-015-9337-1

[30] Trybulec, A., Korniłowicz, A., Naumowicz, A., and Kuperberg, K. (2013). Formal mathematics for mathematicians. *Journal of Automated Reasoning*, 50(2):119–121. doi:10.1007/s10817-012-9268-z

[31] Urban, J., Rudnicki, P., and Sutcliffe, G. (2013). ATP and presentation service for Mizar formalizations. *Journal of Automated Reasoning*, 50(2):229–241. doi:10.1007/s10817-012-9269-y

[32] Vlach, M. (2008). Topologies of approximation spaces of rough set theory, In *Interval/Probabilistic Uncertainty and Non-Classical Logics*, pp. 176–186, *Advances in Soft Computing,* 46, Springer. doi: 10.1007/978-3-540-77664-2_14

[33] Weil, A. (1937). Sur les espaces a structure uniforme et sur la topologie generale. *Act. Sci. Ind.*, 551, Paris.

[34] Yao, Y., Yao, B. (2012). Covering based rough set approximations, *Information Sciences,* 200:91–107. doi:10.1016/j.ins.2012.02.065

[35] Zhu, W. (2007). Generalized rough sets based on relations, *Information Sciences,* 177(22):4997–5011. doi:10.1016/j.ins.2007.05.037

[36] Zhu, W. (2007). Topological approaches to covering rough sets, *Information Sciences,* 177(6):1499–1508. doi:10.1016/j.ins.2006.06.009