

# Automated Comparative Study of Some Generalized Rough Approximations (Extended Abstract)

Adam Grabowski

Institute of Informatics, University of Białystok  
Konstantego Ciołkowskiego 1M, 15-245 Białystok, Poland  
adam@math.uwb.edu.pl

**Abstract.** The paper contains some remarks on building automated counterpart of the research presented during one of the previous *CS&P* workshops by A. Gomolińska. My main objective was the formal (and machine-checked) proof of Theorem 4.1 from her paper “A Comparative Study of Some Generalized Rough Approximations”, hence the title of the present paper is by no means accidental.

**Keywords:** rough approximation, formalization of mathematics, Mizar Mathematical Library

## 1 Motivation

During *Concurrency Specification & Programming International Workshop* in 2001 Anna Gomolińska presented a draft of her paper on the generalization of rough approximation operators. Essentially, the idea behind the research was to collect some standard properties of rough approximations and to cut off some ordinary properties of binary relations to get – a little bit in the spirit of reverse mathematics – a catalogue of equivalences between relations’ properties and corresponding formulas for rough sets. The final – revised and updated – version of the research appeared eventually in *Fundamenta Informaticae* under the title *A Comparative Study of Some Generalized Rough Approximations* [2] and contained also a brief review of various rough approximations, not necessarily classical ones.

During that time, I was also interested in computer-supported formalization of rough sets and I claimed that the paper could be a quite interesting testbed for my studies on the use of proof-assistants to support mathematicians in their work. Just to be a little more explicit, the system which was considered by me was Mizar, equipped with first order logic language and Tarski-Grothendieck set theory as a underlying set theory.

After a deeper study at that time, I decided to abandon the formalization work as it looked too much different from the one I followed as described in [5]. In the approach chosen by Gomolińska, rough inclusion function  $\kappa$  and two uncertainty mappings –  $I$  and  $\tau$  were quite fundamental at the very first first sight.

## 2 Approximations according to Gomolińska

Gomolińska started with – quite general – uncertainty map  $I$  from non-empty universe  $U$  into its powerset. Potentially, no additional assumptions at the beginning are made, with the exception of a kind of reflexivity – that any object  $u$  from the universe is an element of  $I(u)$ .

Theorem 4.1 is the core of the first part. We aimed at the full formalization of this item, so we quote it here using original notation from [2].

**Theorem 4.1** For any sets  $x, y \subseteq U$ , objects  $u, w \in U$ , and  $i = 0, 1$ , it holds that:

- a)  $f_0^d \leq id \leq f_0$ .
- b)  $f_1^d \leq id \leq f_1$ .
- c)  $f_0(x)$  is definable.
- d)  $\forall u \in f_1(x). \kappa(I(u), x) > 0$ .
- e)  $\forall u \in f_1^d(x). \kappa(I(u), x) = 1$ .
- f) If  $\tau(u) = \tau(w)$ , then  $u \in f_0(x)$  iff  $w \in f_0(x)$ ; and similarly for  $f_0^d$ .
- g) If  $I(u) = I(w)$ , then  $u \in f_1(x)$  iff  $w \in f_1(x)$ ; and similarly for  $f_1^d$ .
- h)  $f_i(\emptyset) = \emptyset$  and  $f_i(U) = U$ ; and similarly for  $f_i^d$ .
- i)  $f_i$  and  $f_i^d$  are monotone.
- j)  $f_i(x \cup y) = f_i(x) \cup f_i(y)$ .
- k)  $f_i^d(x \cup y) \supseteq f_i^d(x) \cup f_i^d(y)$ .
- l)  $f_i(x \cap y) \subseteq f_i(x) \cap f_i(y)$ .
- m)  $f_i^d(x \cap y) = f_i^d(x) \cap f_i^d(y)$ .

The approach is significantly different than those of W. Zhu [11] (but Zhu's paper was published later), where the starting point is – at least in my opinion – more natural. Moreover, in the second part of the paper under discussion the catalogue of various rough approximation operators is given (which is outside Zhu's research area).

## 3 Mizar State of the Art

In our formal approach to rough approximations, our choice was to have indiscernibility relation  $\rho$  defined as a binary relation on a non-empty universe  $U$ . Essentially then, this corresponds to an abstract relational structure

$$\mathcal{R} = \langle U, \rho \rangle$$

with all properties credited to the internal relation of  $\mathcal{R}$ .

At the very beginning, we do not assume any of standard properties of approximations (or tolerances) added to the type of  $\rho$ , although we introduce two basic Mizar types, called `Approximation_Space` and `Tolerance_Space`, to have them available in the Mizar Mathematical Library. More thorough discussion on this development is given in [4].

Faithfully following Gomolińska, we should be aware of the fact that we have

$$\mathcal{A} = \langle U, I, \kappa \rangle$$

as a formal approximation space. I had some doubts if I should mention  $\kappa$  at all – rough inclusion function as in the classical approach the membership function was considered. Also  $\rho$  as indiscernibility relation was rather preferred over the uncertainty mapping. My conviction was stronger after the formalization of Zhu's paper on correspondence between properties of binary relations and of rough sets.

In the literature [11] the authors skip over the theory of less known properties of binary relations (or at least treat them as widely known), and I was also in a dead point as the – contrary to the notion of serial or mediate which could be claimed as mathematical folklore – positive or negative alliance relations were really unknown for me.

Even if in considered paper standard lower approximation operator is defined as (10)

$$LOW(x) = \{u \in U : \kappa(I(u), x) = 1\},$$

happily its immediate consequence is available as (12):

$$LOW(x) = \{u \in U : I(u) \subseteq x\}.$$

Then we could be sure the function  $\kappa$  will not be needed at the moment.

## 4 Generalizing Functions

Gomolińska claimed the following name space for approximations operators (18):

$$f_0(x) = \{u \in U : \tau(u) \cap x \neq \emptyset\}$$

$$f_1(x) = \{u \in U : I(u) \cap x \neq \emptyset\}$$

with corresponding  $f_0^d$  and  $f_1^d$ .

Then we can think on the following definition for  $\tau$

**definition**

```

let R be non empty RelStr;
let tau be Function of the carrier of R, bool the carrier of R;
func f_0 R -> map of R means
  for x being Subset of R holds
    it.x = { u where u is Element of R : tau.u meets x };
end;
```

and similar, for uncertainty mapping  $I$ . Hence, it is quite straightforward to see that the common variable in both definitions is just arbitrary function (with proper domain and codomain), let us call it  $f$ , and then it is quite natural to use instead of the above formulation something like

```

definition
  let R be non empty RelStr;
  let f be Function of the carrier of R, bool the carrier of R;
  func ff_0 f -> map of R means
    for x being Subset of R holds
      it.x = { u where u is Element of R : f.u meets x };
end;

```

and then to use

```

definition
  let R be non empty RelStr;
  func f_0 R -> map of R equals
    ff_0 tau R;
  func f_1 R -> map of R equals
    ff_0 UncertaintyMap R;
end;

```

For a reader not very much acquainted with the Mizar system it is not very cryptic, but the use of a keyword `equals` turns automatic treatment of equalities on and then more reasonings can be justified automatically.

Furthermore, we have just a single – more general – theorem expressing (f) and (g) at the same time, instead of two:

```

theorem :: ROUGHS_5:31 :: 4.1 f) Flip
  for u,w being Element of R,
    x being Subset of R st
  f.u = f.w holds
  u in (Flip ff_0 f).x iff w in (Flip ff_0 f).x;

```

It can be noted at the first sight, that there is nothing on approximation operators there – we can think on quite general property of mappings as  $R$  is only non-empty relational structure.

The Mizar functor `Flip f`, which is the formal counterpart of  $f^d$  is defined accordingly as

```

definition let X be set;
  let f be Function of bool X, bool X;
  func Flip f -> Function of bool X, bool X means
    for x being Subset of X holds
      it.x = (f.x)';
end;

```

Obviously, no approximations are needed here, either.

## 5 Attributes

In our Mizar approach, *UPP* and *LOW* operators from [2] are classical upper and lower approximation operators, respectively. Theorem 3.1 from [2] lists them all, but it could be mentioned in this point, that all these – more or less standard – properties were already formulated and proved formally in Mizar script available in MML under identifier `ROUGH5_1` as a direct reflection of Pawlak’s paper [7].

In our study of fuzzy implications, we developed some techniques (*Mizar schemes*) which could decrease the number of technical steps in defining functions.

It should be pointed out that if we start with the uncertainty mapping rather than with indiscernibility relation, the core property is

$$\forall_{u \in U} u \in f(u),$$

which is essentially the formula (1) in [2].

To have faithful formal representation is the above, we introduced the new Mizar attribute

```
definition :: property (1) p. 105
  let R be non empty set;
  let I be Function of R, bool R;
  attr I is map-reflexive means
:: ROUGH5_5: def 1
  for u being Element of R holds u in I.u;
end;
```

As we already mentioned before, our core idea was to start with an approximation space  $\langle U, \rho \rangle$ , where  $U$  is non-empty universe and  $\rho$  is an indiscernibility relation defined on  $U$ . All rough operators, membership and uncertainty functions, and also rough inclusions, could be defined based within this framework. The proper choice of the formal background is especially important, because in the Mizar Mathematical Library, all statements should be proven based on classical logic and/or already proven lemmas.

In this paper, as a rule we do not quote correctness conditions (as they are needed also for definitions), although definitions should also obey the rule of the necessary justification. Even if sometimes of quite technical character, the proof is needed and the example of that is given below. We formulate a bunch of handy *Mizar schemes* which make this activity a little bit less painful for the user.

```
definition
  let R be non empty set;
  func singleton R -> Function of R, bool R means
:: ROUGH5_5: def 2
  for x being Element of R holds it.x = {x};
  existence
  proof
```

```

    deffunc U(object) = {$1};
A1: now
    let x be Element of R;
    {x} c= R;
    hence U(x) in bool R;
    end;
    thus ex f being Function of R, bool R st
    for x being Element of R holds f.x = U(x)
    from FUNCT_2:sch 8(A1);
    end;
    uniqueness; :: here the proof is not quoted
end;

```

Why this definition was really needed? Essentially, to use the Mizar type (and adjectives are important constructors for types) one should prove its non-emptiness, i.e., to construct at least one example of an object possessing this property, to show the usefulness of the attribute.

```

registration
    let R be non empty set;
    cluster singleton R -> map-reflexive;
end;

```

Another handy use of adjectives is given below:

```

registration
    let R, f;
    cluster ff_0 f -> c--monotone;
end;

```

This allows to obtain automatically the proof of Theorem 4.1(i) as follows:

```

registration
    let R be non empty RelStr; :: i)
    cluster f_0 R -> c--monotone;
    cluster f_1 R -> c--monotone;
end;

```

## 6 Conclusions and Further Work

Obviously, building proper formal framework for further work is crucial. Having said that, we can build the rest of functions defined in [2]:  $f_2, \dots, f_9$ , having ten mappings altogether. This allows for the possibility of automated reasoning on these operators, which could be made in the nearest future, as well as the reasoning on the – temporarily abandoned – rough inclusion function  $\kappa$ .

Even if this extended abstract is only a draft of the formal work done, the complete formalization (just accepted for inclusion into the Mizar Mathematical Library) is available at

<http://mizar.uwb.edu.pl/library/roughsets/>

Using 1386 lines of Mizar code (44 kilobytes) and formulating 7 definitions and proving 53 theorems we completed the first part of formalization work of Gomolińska's paper [2]. The `roughs_5.abs` file contains the abstract for the development while `roughs_5.miz` is the complete source file.

## References

1. Bryniarski E.: Formal conception of rough sets, *Fundamenta Informaticae*, 27(2/3), 109–136 (1996)
2. Gomolińska A.: A comparative study of some generalized rough approximations, *Fundamenta Informaticae*, 51, 103–119 (2002)
3. Grabowski A.: Mechanizing complemented lattices within Mizar system. *Journal of Automated Reasoning*, 55:211–221 (2015)
4. Grabowski A.: Efficient rough set theory merging, *Fundamenta Informaticae*, 135(4), 371–385 (2014)
5. Grabowski A.: Automated discovery of properties of rough sets, *Fundamenta Informaticae*, 128(1–2), 65–79 (2013)
6. Järvinen J.: Lattice theory for rough sets, *Transactions on Rough Sets VI*, LNCS (LNAI) 4374, 400–498 (2007)
7. Pawlak, Z.: *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer, Dordrecht (1991). doi: 10.1007/978-94-011-3534-4
8. Rasiowa H. and Sikorski R.: *The Mathematics of Metamathematics*. Polish Scientific Publishers (1970)
9. Skowron A., Stepaniuk J.: Tolerance approximation spaces, *Fundamenta Informaticae*, 27(2–3), pp. 245–253 (1996). doi: 10.3233/FI-1996-272311
10. Wiedijk F.: Formal proof – getting started, *Notices of the American Mathematical Society*, 55(11), 1408–1414 (2008)
11. Zhu W.: Generalized rough sets based on relations, *Information Sciences*, 177(22), pp. 4997–5011 (2007). doi: 10.1016/j.ins.2007.05.037