RIFs as the Formal Tool of Measuring Similarity between Sets

Adam Grabowski

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

Abstract. In my paper, I will present some results towards the formalization of Rough Inclusion Functions (RIFs) and their complementary mappings using the Mizar system. Following the lines established by Anna Gomolińska, we reuse the notions which are already present in the Mizar Mathematical Library, in the theory of metric spaces, in order to establish the connections between the theory of such metrics and the theory of rough sets.

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

1 Motivation

Rough sets discovered by Z. Pawlak [15] are a tool for knowledge discovery and modelling under imperfect information; this is especially the case nowadays, where we often face large databases of information gathered from various sources. Knowledge discovery by means of computerized tools seems to be important taking into account the amount of available information. Similar tools can be used in order to check of verify the correctness of mathematical theorems expressed in artificial (but still close to the the natural) language understandable for computers.

Rough Inclusion Functions (RIFs for short) establish the connection between classical set theory and theory of rough sets. Their quantitative nature shows its feasibility in large databases; on the other hand such a view is also elegant from the mathematical point of view for the theory of rough sets, at the same time opening paths of further research – based on this single predicate one can build the whole theory, as it was done in the case of Leśniewski mereology.

As the proof assistant used to formalize the theory we have chosen the Mizar system, relatively well-known system based on classical logic, together with its repository of texts formally verified by computer – the Mizar Mathematical Library (MML). At *Concurrency Specification & Programming Workshop* 2018 we presented the faithful translation of Gomolińska's paper [4] on rough inclusion functions. This time, extending the view presented in [4] and exploring the ideas from [3], we focus rather on the reuse of existing fields already present in the Mizar repository, namely complementary mappings, metric spaces, and similarity measures between subsets.

In our formal approach to rough approximations, our choice was to have indiscernibility relation ρ 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 ρ, 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 [10].

Our core testing scenario for the formalization process was that, starting with the mapping complementary to rough inclusion function κ_1, we can obtain an instance of the Marczewski-Steinhaus metric (or, more generally, we can automatically discover metric properties of defined operators). Namely, for subsets X, Y of the non-empty universe U, we can define the operator δ_1 as (see [3])

$$\delta_1(X, Y) = \bar{\kappa}_1(X, Y) + \bar{\kappa}_1(Y, X).$$

Essentially then, we have to build a path from approximation spaces on U to the theory of metric (or metrizable) spaces, where we can already find some well-known distance operators. In order to achieve this goal, a list of certain formalized notions should be available at hand.

2 Rough Inclusion Functions

In this section, we will briefly sketch the content of the Mizar formalization [11] collected in the file under MML identifier `ROUGHIF1`[1]. For a given universe U, rough inclusion functions (RIFs for short) are the mappings κ from $\wp U \times \wp U$ into unit interval which satisfy two properties:

$$\mathrm{rif}_1(\kappa) \Leftrightarrow \forall_{X,Y \subseteq U} \; (\kappa(X, Y) = 1 \Leftrightarrow X \subseteq Y)$$

$$\mathrm{rif}_2(\kappa) \Leftrightarrow \forall_{X,Y,Z \subseteq U} \; (Y \subseteq Z \Rightarrow \kappa(X, Y) \leq \kappa(X, Z))$$

These two characteristic properties were introduced in Mizar as attributes `satisfying_RIF1` and `satisfying_RIF2`. For example, the first property is as follows:

[1] The development is available at `http://mizar.uwb.edu.pl/library/roughif1/`

```
definition let R be non empty RelStr;
           let f be preRIF of R;
  attr f is satisfying_RIF1 means
    for X,Y being Subset of R holds f.(X,Y) = 1 iff X c= Y;
end;
```

The classical example of a RIF is the standard RIF, κ, defined formally in two steps. Firstly, the Mizar functor `kappa` is defined for two subsets of an approximation space (frankly, it was defined more generally):

```
definition let R be finite Approximation_Space;
           let X,Y be Subset of R;
  func kappa (X,Y) -> Element of [.0,1.] equals
     card (X /\ Y) / card X if X <> {}
     otherwise 1;
  correctness;
end;
```

Then we can define appropriate mapping pointwise as

```
definition let R be finite Approximation_Space;
  func kappa R -> Function of
    [:bool the carrier of R, bool the carrier of R:], [.0,1.] means
    for x,y being Subset of R holds it.(x,y) = kappa (x,y);
end;
```

To assure the automatic understanding that such defined `kappa` has the needed properties of RIF, the *registration of a cluster* is formulated and proved.

```
registration let R be finite Approximation_Space;
  cluster kappa R -> satisfying_RIF1 satisfying_RIF2;
  coherence;
end;
```

Practically, all the content of a paper [4] devoted to RIFs is formalized already.

3 Complementary Mappings

Having defined rough inclusion function, we can go on with the modification of this operator, using methods known from fuzzy set theory (as fuzzy implication operators or triangular norms or conorms in `FUZIMPL1` or `FUZNORM1`).

With any mapping $f : \wp U \times \wp U \to [0, 1]$ we can associate a complementary mapping $\bar{f} : \wp U \times \wp U \to [0, 1]$ as

$$\bar{f}(X, Y) = 1 - f(X, Y).$$

Mizar version of the above is introduced as

```
definition let R be finite Approximation_Space;
  let f be preRoughInclusionFunction of R;
  func CMap f -> preRoughInclusionFunction of R means
    for x,y being Subset of R holds
      it.(x,y) = 1 - f.(x,y);
   correctness;
end;
```

Using this definition, we can, e.g. the following:

```
theorem PropEx3k:
  X <> {} implies (CMap kappa R).(X,Y) = card (X \ Y) / card X
  proof
    assume
A1: X <> {};
    X \ Y = X \ (X /\ Y) by XBOOLE_1:47; then
T1: card (X \ Y) = card X - card (X /\ Y) by XBOOLE_1:17,CARD_2:44;
    (CMap kappa R).(X,Y) = 1 - (kappa R).(X,Y) by CDef
      .= 1 - kappa (X,Y) by ROUGHIF1:def 2
      .= 1 - card (X /\ Y) / card X by A1,ROUGHIF1:def 1
      .= (card X / card X) - card (X /\ Y) / card X by A1,XCMPLX_1:60
      .= card (X \ Y) / card X by T1,XCMPLX_1:120;
    hence thesis;
  end;
```

In order to shorten the notations, and to enhance the formulation of corresponding properties, we introduced also the attribute `co-RIF-like`, stating that for every RIF f, the mapping complementary to f is again rough inclusion function. Obviously, every `CMap f` for arbitrary f, which is a RIF, has this property.

Independently, we can prove the following analogon of Proposition 6a):

```
theorem Prop6a: :: Proposition 6 a)
  for kap being RIF of R holds
    (CMap kap).(X,Y) = 0 iff X c= Y;
```

In similar manner, we completed the whole Proposition 6 from [4].

4 Metric Spaces

Even if the repository of Mizar texts strongly depend on Tarski-Grothendieck set theory, which is roughly equivalent with Zermelo-Fraenkel with the Axiom of Choice, so-called structures are important element of the language, especially useful to define abstract algebraic structures.

Metric spaces are excellent example of such a construction: we start with arbitrary set U and a real function from the Cartesian square of U.

```
definition
  struct (1-sorted) MetrStruct
   (# carrier -> set,
     distance -> Function of [:the carrier,the carrier:],REAL #);
end;
```

Additional adjectives of the `distance` function correspond to standard properties defining distance operators. For example, `triangle` means the following:

```
definition
  let A be set;
  let f be PartFunc of [:A,A:], REAL;
  attr f is triangle means
  for a, b, c being Element of A holds f.(a,c) <= f.(a,b) + f.(b,c);
end;
```

Following these lines, the theory of metric spaces is build upon the Mizar type `MetrSpace`:

```
definition
  mode MetrSpace is Reflexive discerning symmetric triangle MetrStruct;
end;
```

That is, metric spaces are structures `MetrStruct` under the aassumption that `distance` satisfies ordinary properties of the distance function.

5 Similarity Measures between Subsets

Additionally, some preparatory work was started to bridge the gap between the theory of (abstract) metric spaces, concrete metrics defined to measure the distance between the subsets. I focus on the Hausdorff distance which was defined by me in the Mizar article `HAUSDORF` available in the Mizar Mathematical Library as the following[2]:

```
definition
  let M be non empty MetrSpace, P, Q be Subset of TopSpaceMetr M;
  func HausDist (P, Q) -> Real equals
:: HAUSDORF:def 1
  max ( max_dist_min (P, Q), max_dist_min (Q, P) );
  commutativity;
end;
```

Such quite complicated definition using an auxiliary functor `max_dist_min` introduced previously allows me to show the ordinary metric-like properties of the above, e.g. the triangle inequality (so, it states that `HausDist` is `triangle`):

[2] All item from MML can be tracked via its MML identifier at `http://mizar.org/version/current/html/`

```
theorem :: HAUSDORF:38
  for M being non empty MetrSpace,
       P, Q, R being non empty Subset of TopSpaceMetr M st
    P is compact & Q is compact & R is compact holds
      HausDist (P, R) <= HausDist (P, Q) + HausDist (Q, R);
```

In order to show some rough set-specific connections, we defined relatively well-known Jaccard distance, starting with the number showing the similarity between subsets of arbitrary finite 1-sorted structure:

```
definition let R be finite 1-sorted;
           let A, B be Subset of R;
  func JaccardIndex (A,B) -> Element of [.0,1.] equals :JacInd:
    card (A /\ B) / card (A \/ B) if A \/ B <> {}
    otherwise 1;
  correctness;
end;
```

Using this functor, and previously formalizing its basic properties, we can conclude with the definition of Jaccard function:

```
definition let R be finite 1-sorted;
  func JaccardDistance R -> Function of
    [:bool the carrier of R, bool the carrier of R:], REAL means :JacDef:
  for A, B being Subset of R holds
    it.(A,B) = 1 - JaccardIndex (A,B);
  correctness;
end;
```

Some other distance operators are present in the MML, e.g. in the article `METRIC_3`. In the case of Jaccard distance, first three adjectives are rather simple to show (that it is `Reflexive`, `discerning`, and `symmetric`). To prove the remaining one, first we constructed δ_1 from Section 1, in a straightforward way:

```
definition let R;
  func delta_1 R -> preRIF of R means :Delta1:
  for x,y being Subset of R holds
    it.(x,y) = (CMap kappa_1 R).(x,y) + (CMap kappa_1 R).(y,x);
  correctness;
end;
```

Finally, we can show that such defined δ_1 and `JaccardDistance` coincide, and hence, as the earlier is a distance (and satisfies the triangle inequality), so is the latter.

```
theorem
  for A, B being Subset of R holds
    (JaccardDistance R).(A,B) = (delta_1 R).(A,B);
```

Although the formal proof in arbitrary setting is quite complex, our version, using the proven correspondence with rough sets, is immediate.

```
registration let R be finite 1-sorted;
  cluster JaccardDistance R -> triangle;
  coherence;
end;
```

Having all these formal notions at hand, our future work will be to discover automatically possible connections between these topics, as suggested in [3].

All formal developments described above are submitted for inclusion into the Mizar Mathematical Library and are available under the following URL:

`http://mizar.org/library/roughif2/`

References

1. Bancerek, G., Byliński, Cz., Grabowski, A., Korniłowicz, A., Matuszewski, R., Naumowicz, A., Pąk, K., and Urban, J.: Mizar: State-of-the-art and beyond. In Kerber, M. Carette, J. et al. (Eds.): Intelligent Computer Mathematics – International Conference, CICM 2015, Washington, DC, USA, July 13–17, 2015, Proceedings, *Lecture Notes in Computer Science*, 9150, Springer, pp. 261–279 (2015). doi:10.1007/978-3-319-20615-8_17
2. Gomolińska, A.: A comparative study of some generalized rough approximations, *Fundamenta Informaticae,* 51, pp. 103–119 (2002)
3. Gomolińska, A.: On certain rough inclusion functions, *Transactions on Rough Sets IX,* Lecture Notes in Computer Science, 5390, pp. 35–55 (2008). doi:10.1007/978-3-540-89876-4_3
4. Gomolińska, A.: On three closely related rough inclusion functions, Proceedings of RSEISP 2007, *Lecture Notes in Artificial Intelligence,* 4585, pp. 142–151 (2007). doi:10.1007/978-3-540-73451-2_16
5. Grabowski, A.: Mechanizing complemented lattices within Mizar type system, *Journal of Automated Reasoning,* 55(3), pp. 211–221 (2015). doi:10.1007/s10817-015-9333-5
6. Grabowski, A.: Binary relations-based rough sets – an automated approach, *Formalized Mathematics,* 24(2), pp. 143–155 (2016). doi:10.1515/forma-2016-0011
7. Grabowski, A.: Automated discovery of properties of rough sets, *Fundamenta Informaticae,* 128(1–2), pp. 65–79 (2013). doi:10.3233/FI-2013-933
8. Grabowski, A.: Efficient rough set theory merging, *Fundamenta Informaticae,* 135(4), pp. 371–385 (2014). doi:10.3233/FI-2014-1129
9. Grabowski, A.: Computer certification of rough sets based on relations, Proceedings of IJCRS 2017, *Lecture Notes in Artificial Intelligence,* 10313, pp. 83–94 (2017), doi:10.1007/978-3-319-60837-2_7
10. Grabowski, A.: Automated comparative study of some generalized rough approximations, *Proceedings of CS&P 2018,* pp. 39–45 (2018)
11. Grabowski, A.: Formal development of rough inclusion functions, *Formalized Mathematics,* 27(3), pp. 337–343 (2019), doi:10.2478/forma-2019-0027
12. Grabowski, A., Korniłowicz, A., Naumowicz, A.: Four decades of Mizar, *Journal of Automated Reasoning,* 55(3), pp. 191–198 (2015). doi:10.1007/s10817-015-9345-1
13. Grabowski, A.: Lattice theory for rough sets – a case study with Mizar, *Fundamenta Informaticae,* 147(2–3), pp. 223–240 (2016). doi:10.3233/FI-2016-1406

14. Grabowski, A.: On the computer-assisted reasoning about rough sets, in: Dunin-Kęplicz, B., Jankowski, A., Szczuka, M. (Eds.) *Monitoring, Security and Rescue Techniques in Multiagent Systems*, Advances in Soft Computing, 28, pp. 215–226 (2005). doi:10.1007/3-540-32370-8_15
15. Pawlak, Z.: *Rough Sets: Theoretical Aspects of Reasoning about Data,* Kluwer, Dordrecht (1991). doi:10.1007/978-94-011-3534-4
16. Polkowski, L.: *Approximate Reasoning by Parts. An Introduction to Rough Mereology.* Intelligent Systems Reference Library, vol. 20, Springer Verlag, Berlin-Heidelberg (2011). doi:10.1007/978-3-642-22279-5
17. Polkowski, L. and Skowron, A.: Rough mereology: A new paradigm for approximate reasoning, *International Journal of Approximate Reasoning*, 15(4), pp. 333–365 (1996). doi:10.1016/S0888-613X(96)00072-2
18. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces, *Fundamenta Informaticae,* 27(2–3), pp. 245–253 (1996). doi:10.3233/FI-1996-272311
19. Yao, Y.Y.: Two views of the rough set theory in finite universes, *International Journal of Approximate Reasoning,* 15(4), pp. 291–317 (1996). doi:10.1016/S0888-613X(96)00071-0
20. Zadeh, L.: Fuzzy sets, *Information and Control,* 8(3), pp. 338-353 (1965). doi:10.1016/S0019-9958(65)90241-X
21. Zhu, W.: Generalized rough sets based on relations, *Information Sciences,* 177(22), pp. 4997–5011 (2007). doi:10.1016/j.ins.2007.05.037