On the Correspondence between Approximations and Similarity

Patrick Doherty[1] and Andrzej Szałas[1,2]

[1] Dept. of Computer and Information Science
Linköping University
SE-581 83 Linköping, Sweden
`patdo@ida.liu.se`

[2] The College of Economics and Computer Science,
Olsztyn, Poland
`andsz@ida.liu.se`

In Proceedings of 4th International Conference on Rough Sets and Current Trends in Computing, RSCTC-2004.

Abstract. This paper focuses on the use and interpretation of approximate databases where both rough sets and indiscernibility partitions are generalized and replaced by approximate relations and similarity spaces. Similarity spaces are used to define neighborhoods around individuals and these in turn are used to define approximate sets and relations. There is a wide spectrum of choice as to what properties the similarity relation should have and how this affects the properties of approximate relations in the database. In order to make this interaction precise, we propose a technique which permits specification of both approximation and similarity constraints on approximate databases and automatic translation between them. This technique provides great insight into the relation between similarity and approximation and is similar to that used in modal correspondence theory. In order to automate the translations, quantifier elimination techniques are used.

1 Introduction

There is a natural generalization of relational databases where one uses intuitions from rough set theory [14] and rather than storing and querying crisp relations, one stores and queries rough relations consisting of an upper and lower approximation of the implicit crisp relation whose definition one tries to approximate [4]. There is also a natural generalization of an indiscernibility relation used in rough set theory, where rather than partitioning the universe of discourse U into indiscernibility classes, one can consider a covering of U by similarity-based neighborhoods (see, e.g., [8]) with lower and upper approximations of relations defined via the neighborhoods. To mark the difference, we will use the terms approximate relations and approximate databases instead of rough relations and rough databases. Approximate databases and tolerance spaces have been shown to be quite versatile in many applications areas requiring the use of approximate knowledge structures [5, 7].

When taking this step and generalizing to approximate relations and databases, there are many choices that can be made as regards the constraints one might want to place on the similarity relation used to define upper and lower approximations. For example, we would not want the relation to have the property of transitivity since similar things don't naturally chain in a transitive manner. Many of these issues are discussed in the context of rough sets (see, e.g., [15–17]). Whatever choices are made, one wants to ensure that these constraints are enforced while querying an approximate database. In a similar manner, there are many constraints that are more naturally expressed in terms of upper and lower approximations which must remain consistent with the properties one assumes for the underlying similarity relation and which also have to be enforced while querying the approximate database. For example, for any relation in an approximate database, we would like to ensure that the lower approximation is a subset of the upper approximation. There are even constraints we might like to enforce that refer to the crisp relation of which we implicitly represent in terms of an upper and lower approximation. For example the lower approximation should be a subset of this crisp relation.

The goal of this paper is to study the interaction between constraints stated in terms of a language of approximate relations and constraints stated in terms of the underlying similarity relation which is used to define neighborhoods. To do this, we first define a language of set theoretical terms which permit us to represent boolean constraints on upper and lower approximations. We then introduce a first-order language and translation function which translates constraints in the set theoretical language into first-order formulas. These first-order formulas are then quantified over various relations in the formulas because we are interested in universal constraints. We then use quantifier elimination techniques to generate logically equivalent formulas in a first-order language but in this case, the resulting output only refers to the similarity relation if the elimination is successful.

This technique is analogous to techniques used in modal correspondence theory [19] where one studies the nature and expressiveness of modal axioms by viewing them as expressing constraints on the possible worlds alternative relation in the underlying Kripke frames. These constraints are represented as 1st- or higher-order logical formulas and correspondence theory is the study of the generation and relations between these correspondences. We approach the topic of the relation between approximate relation axioms and the formulas expressing constraints on the underlying similarity relation in a similar manner. In this case though, we can often automatically generate the resulting correspondence through the use of quantifier eliminations techniques [6] developed by the authors in another context.

The correspondences considered in this paper are not surprising in the view of [19] as well as results more directly oriented towards rough set theory (for an overview of results see, e.g., [11]). However, the novelty of our approach is that we show a uniform, principled way to compute appropriate correspondences and focus on similarity spaces rather then Kripke structures.

2 Preliminaries

The starting point for our approach are tolerance spaces, as introduced in [8]. Technically, they allow us to classify a universe of individuals into indiscernibility or tolerance neighborhoods based on a parameterized tolerance relation. This is a generalization on the indiscernibility partitions used in rough set theory where instead of partitions, the neighborhoods provide a covering of the universe. In fact, tolerance functions are required to induce reflexive and symmetric neighborhood relations, while rough neighborhood relations are additionally transitive.

Tolerance spaces can still be generalized to represent even weaker notions of similarity in a universe of individuals. Consequently, we consider similarity spaces, where the definition of a similarity function has no initial constraints.

Definition 1. *By a* similarity function *on a set U we mean any function $\tau : U \times U \longrightarrow [0, 1]$.*

For $p \in [0, 1]$ by a similarity relation to a degree at least p, induced by τ, *we mean the relation $\sigma^p \stackrel{\mathrm{def}}{=} \{\langle x, y\rangle \mid \tau(x, y) \geq p\}$.* □

In what follows, we assume p is given and use $\sigma(x, y)$ to denote the characteristic function of σ^p.

Similarity relations are used to construct similarity neighborhoods.

Definition 2. *By a* neighborhood function wrt σ^p *we mean a function given by $n^{\sigma^p}(u) \stackrel{\mathrm{def}}{=} \{u' \in U \mid \sigma^p(u, u')$ holds$\}$. By a* neighborhood *of u wrt σ^p we mean the value $n^{\sigma^p}(u)$.* □

Definition 3. *A* similarity space *is defined as the tuple $S = \langle U, \tau, p\rangle$, consisting of a nonempty set U, called the* domain *of S, a* similarity function *τ, and a* similarity threshold *$p \in [0, 1]$.*

Let $A \subseteq U$. The lower and upper approximation of A wrt S, *denoted respectively by A_{S^+} and $A_{S^\oplus}$, are defined by $A_{S^+} = \{u \in U : n^{\sigma^p}(u) \subseteq A\}$, $A_{S^\oplus} = \{u \in U : n^{\sigma^p}(u) \cap A \neq \emptyset\}$.* □

We shall often provide similarity spaces as pairs $\langle U, \sigma\rangle$, where σ is a similarity relation induced by a given similarity function and threshold. This simplifies the presentation and is sufficient for the purposes of the current paper.

The following proposition provides us with an alternative way to define upper and lower approximations and will be used throughout the paper.

Proposition 1. *Let $S = \langle U, \sigma\rangle$ be a similarity space and let $A \subseteq U$. Then $A_{S^+} = \{a \in A \mid \forall b\, [\sigma(a, b) \rightarrow b \in A]\}$ and $A_{S^\oplus} = \{a \in A \mid \exists b\, [\sigma(a, b) \wedge b \in A]\}$.* □

As a basis for doing quantifier elimination, we will use the following lemma due to Ackermann [1] (see also, e.g., [6, 18]), where $\Psi\left[P(\overline{\alpha}) \leftarrow [\Phi]^{\overline{x}}_{\overline{\alpha}}\right]$ means that every occurrence of P in Ψ is to be replaced by Φ where the actual arguments $\overline{\alpha}$ of P, replaces the variables of $\overline{x}$ in Φ (and the bound variables are renamed if necessary).

Lemma 1. *Let P be a predicate variable and let Φ and $\Psi(P)$ be first–order formulae such that $\Psi(P)$ is positive w.r.t. P and Φ contains no occurrences of P at all. Then $\exists P\, \forall \overline{x}\, (P(\bar{x}) \rightarrow \Phi(\bar{x}, \bar{y})) \wedge \Psi(P) \;\equiv\; \Psi\left[P(\overline{\alpha}) \leftarrow [\Phi]^{\overline{x}}_{\overline{\alpha}}\right]$ and similarly if the sign of P is switched and Ψ is negative w.r.t. P.* $\square$

3 Languages for Expressing Similarity and Approximation Constraints

In order to specify constraints on approximate relations and similarity relations and to show correspondences between them, we will introduce a number of languages and translations between them. We begin by defining a language for approximation constraints.

Definition 4. *Let U be a set, $\bar{A}$ be a tuple of set symbols*[3] *(respective sets are assumed to be included in U) and let $S = \langle U, \sigma \rangle$ be a similarity space.* Set-theoretical terms over vocabulary $\bar{A} \cup \{S\}$ *are defined as follows:*

- *for $A \in \bar{A}$, A is a set-theoretical term*
- *if α is a set-theoretical term then $-\alpha, \alpha_{S^+}, \alpha_{S^\oplus}$ are set-theoretical terms*
- *if α, β are set-theoretical terms then $\alpha \cup \beta$ is also a set-theoretical term.*

If α and β are set-theoretical terms over $\bar{A} \cup \{S\}$ then $\alpha \subseteq \beta$ is an atomic set-theoretical formula over $\bar{A} \cup \{S\}$. *The set of* set-theoretical formulas *is the least set which contains all atomic set-theoretical formulas and is closed under the classical propositional connectives.* $\square$

We also define $(\alpha \cap \beta) \stackrel{\text{def}}{=} -(-\alpha \cup -\beta)$ and $(\alpha = \beta) \stackrel{\text{def}}{\equiv} (\alpha \subseteq \beta \wedge \beta \subseteq \alpha)$.

Given approximation constraints in the language above, we would like to translate such constraints into formulas in a first-order language as follows.

Definition 5. *Let $U, \bar{A}, S$ be as in Definition 4. Let α be a set-theoretical term over $\bar{A} \cup \{S\}$ and x be a variable over U. Then the* translation $Tr(\alpha, x)$ of set-theoretical terms *into first-order formulas is defined inductively as follows:*

- $Tr(A, x) \stackrel{\text{def}}{=} A'(x)$*, where $A \in \bar{A}$ and A' is a fresh unary relation symbol*
- $Tr(-\alpha, x) \stackrel{\text{def}}{=} \neg Tr(\alpha, x)$

[3] Later on we will use the same symbols to denote sets and corresponding characteristic relations.

– $Tr(\alpha_{S+}, x) \stackrel{\text{def}}{=} \forall y\, [\sigma(x, y) \rightarrow Tr(\alpha, y)]$, *where* y *is a fresh variable*
– $Tr(\alpha_{S\oplus}, x) \stackrel{\text{def}}{=} \exists y\, [\sigma(x, y) \wedge Tr(\alpha, y)]$, *where* y *is a fresh variable*
– $Tr(\alpha \cup \beta, x) \stackrel{\text{def}}{=} Tr(\alpha, x) \vee Tr(\beta, x)$.

The translation $Tr(\gamma, x)$ of set-theoretical formulas *into first-order formulas is defined to satisfy* $Tr(\alpha \subseteq \beta, x) \stackrel{\text{def}}{=} \forall x\, (Tr(\alpha, x) \rightarrow Tr(\beta, x))$ *and to preserve the classical propositional connectives.* □

Example 1. Let a vocabulary consist of sets A,B,C and similarity space S. Then:

$$\begin{aligned} &Tr((A \cup B)_{S+} \subseteq C_{S\oplus}, x) = \\ &\quad = \forall x\, [\forall y\, (\sigma(x, y) \rightarrow (A(y) \vee B(y))) \rightarrow \exists z\, (\sigma(x, z) \wedge C(z))]. \end{aligned}$$ □

4 Computing Correspondences between Approximations and Similarity

For the purposes of this section, we will fix a similarity space $S = \langle U, \sigma \rangle$ and a tuple of sets $\bar{A}$, where all sets in $\bar{A}$ are included in U.

4.1 The general technique

The general techniques used to compute correspondences between similarity constraints and approximation constraints are those described in [18] in the context of modal logics, but rather than working with the alternative relation on possible worlds, we will work with the similarity space S. The steps are as follows:

1. express the required property of approximations as a set-theoretical formula $\gamma(\bar{A})$ over vocabulary $\bar{A} \cup \{S\}$
2. compute the translation $Tr(\gamma(\bar{A}), x)$ of the formula obtained in step 1 according to Definition 5
3. consider the formula $\forall \bar{A}\, [Tr(\gamma(\bar{A}), x)]$ and eliminate second-order variables $\bar{A}$, if possible.

If the second-order quantifier elimination is successful then the resulting formula uses only the non-logical symbols σ and $=$ and is logically equivalent to the initial logical translation of the set-theoretical property. The quantifier elimination step can be automated using the algorithm given in [18] or its generalization known as the DLS algorithm of [6]. There are also other applicable methods which may be used (for an overview of known techniques see [12]).

In the remainder of this section, we will select a number of approximation constraints for analysis and show their correspondence with similarity constraints.

4.2 The meaning of inclusion $A_{S+} \subseteq A_{S\oplus}$

Consider the very basic requirement in rough set theory that the lower approximation of a set should be contained in its upper approximation, i.e., for any set

A we have $A_{S^+} \subseteq A_{S^\oplus}$. The translation of this approximation constraint results in the following first-order formula:

$$Tr(A_{S^+} \subseteq A_{S^\oplus}, x) = \forall x\, [Tr(A_{S^+}, x) \rightarrow Tr(A_{S^\oplus}, x)]$$
$$= \forall x\, [\forall y\, (\sigma(x, y) \rightarrow A(y)) \rightarrow \exists z\, (\sigma(x, z) \wedge A(z))].$$

We universalize over all relations A and get the following second-order formula:

$$\forall A\, \forall x\, [\forall y\, (\sigma(x, y) \rightarrow A(y)) \rightarrow \exists z\, (\sigma(x, z) \wedge A(z))].$$

To apply Ackermann's lemma, a number of syntactic transformations on the original formula are required. In this case, we first negate this formula and switch the order of initial existential quantifiers:

$$\exists x\, \exists A\, [\forall y\, (\sigma(x, y) \rightarrow A(y)) \wedge \forall z\, (\neg\sigma(x, z) \vee \neg A(z))].$$

Ackermann's lemma is then applied resulting in a logically equivalent first-order formula representing the following similarity constraint:

$$\exists x\, [\forall z\, (\neg\sigma(x, z) \vee \neg\sigma(x, z))].$$

After simplifying and negating again we find that the initial requirement is equivalent to $\forall x\, \exists z\, \sigma(x, z)$, i.e., to the seriality of σ.[4] Upon analysis, this leads to an interesting observation:

Proposition 2. *The condition that for any set A, the approximation constraint $A_{S^+} \subseteq A_{S^\oplus}$ holds, is equivalent to the seriality of σ, i.e., to $\forall x\, \exists z\, \sigma(x, z)$.* □

Seriality is a weaker requirement on σ than reflexivity, since reflexivity implies seriality. Assuming this is the only constraint placed on σ, what might this mean intuitively. In an epistemic context, one use of such a weak notion of similarity might be to represent a type of self-awareness, or lack of self-awareness in this case. Here is an example:

Example 2. Consider a society of (at least two) similar robots equipped with cameras and image processing software that allows a robot to recognize similar objects. Assume that because of its camera placement each robot can observe the whole environment except for itself. Assume any robot knows that it is similar to other robots. Based only on such knowledge no robot can verify that it is similar to itself. Here similarity is serial, but not reflexive. □

The following example shows another situation where similarity could be interpreted as serial, but not reflexive.

[4] This property reflects the axiom D of modal logics. The properties considered in consecutive subsections reflect modal axioms T, B and 4, respectively.

Example 3. On a daily basis, humans often use many different relations of similarity concurrently. In commonsense reasoning these relations are generally kept apart, because this would lead to invalid conclusions. For example, assume we consider a similarity between parents and children in the sense that a child is similar to it's parent. Suppose further that we do not want to mix this notion of similarity with other similarities, e.g. those of persons to themselves. More formally we can say that $\sigma(x, y)$ holds if x is a child of y. Since everybody has a parent, σ is serial. Obviously it is not reflexive, since no one is its own child. In this case it would not be symmetric or transitive. □

4.3 The meaning of inclusion $A_{S^+} \subseteq A$

The properties we consider in this section and the next two sections are well-known topological properties if one considers the lower approximation to be the interior operation and the upper approximation to be the closure operation. From a modal logic perspective, the lower and upper approximations can be considered analogous to modal necessity and possibility, respectively.

We first translate the approximation constraint $Tr(A_{S^+} \subseteq A, x)$ into:

$$\forall x\, [\forall y\, (\sigma(x, y) \to A(y)) \to A(x)].$$

A straightforward calculation, similar to one used for modal logics in [18] shows that the universal requirement

$$\forall A\, \forall x\, [\forall y\, (\sigma(x, y) \to A(y)) \to A(x)]$$

is equivalent to the similarity constraint $\forall x\, \sigma(x, x)$, i.e., to the reflexivity of σ.

Proposition 3. *The condition that for any set A, $A_{\sigma^+} \subseteq A$ holds is equivalent to the reflexivity of σ, i.e., to the requirement that $\forall x\, \sigma(x, x)$ holds.* □

4.4 The meaning of inclusion $A \subseteq (A_{S^\oplus})_{S^+}$

We first translate the approximation constraint $Tr(A \subseteq (A_{S^\oplus})_{S^+}, x)$ into a first-order formula:

$$\forall x\, [A(x) \to \forall y\, (\sigma(x, y) \to \exists z\, (\sigma(y, z) \wedge A(z)))].$$

A straightforward calculation, similar to one used for modal logics in [18] shows that the universal requirement

$$\forall A \forall x\, [\forall y\, (\sigma(x, y) \to A(y)) \to A(x)]$$

is equivalent to the similarity constraint $\forall x, y\, [\sigma(x, y) \to \sigma(y, x)]$, i.e., to the symmetry of σ.

Proposition 4. *The condition that for any set A, $A \subseteq (A_{S^\oplus})_{S^+}$ holds is equivalent to the symmetry of σ, i.e., to the requirement that $\forall x, y\, [\sigma(x, y) \to \sigma(y, x)]$ holds.* □

4.5 The meaning of inclusion $A_{S^+} \subseteq (A_{S^+})_{S^+}$

We first translate the approximation constraint $Tr(A_{S^+} \subseteq (A_{S^+})_{S^+}, x)$ into:

$$\forall x \, [\forall y(\sigma(x, y) \to A(y)) \to \forall z(\sigma(x, z) \to \forall u \, (\sigma(z, u) \to A(u)))].$$

A straightforward calculation, similar to one used for modal logics in [18] shows that the universal requirement

$$\forall A \forall x \, [\forall y(\sigma(x, y) \to A(y)) \to \forall z(\sigma(x, z) \to \forall u \, (\sigma(z, u) \to A(u)))]$$

is equivalent to the similarity constraint $\forall x, z, u \, [(\sigma(x, z) \wedge \sigma(z, u)) \to \sigma(x, u)]$, i.e., to the transitivity of σ.

Proposition 5. *The condition that for any set A, $A_{S^+} \subseteq (A_{S^+})_{S^+}$ holds is equivalent to the transitivity of σ, i.e., to $\forall x, z, u \, [(\sigma(x, z) \wedge \sigma(z, u)) \to \sigma(x, u)]$.* □

5 Approximate Database Considerations

Based on the results above, when working with approximate databases, it is important that the use of the database is consistent with the approximation and similarity constraints envisioned by the database engineer and required by the particular application. In some respects, the approximate and similarity constraints have the role of integrity constraints in standard database theory. Yet, enforcing these constraints is not as straightforward. We now consider this issue.

Definition 6. *By a* (relational, crisp) database *we understand a tuple*

$$D = \left\langle U, \{R^j \mid j \in J\} \right\rangle,$$

where U is a finite set, called the domain *of D and $\{R^j \mid j \in J\}$ is a finite collection of relations over U. By an* approximate database *we understand a tuple*

$$D = \left\langle U, \left\{ R^j \mid R^j = \left\langle R^j_+, R^j_\oplus \right\rangle \text{ and } j \in J \right\} \right\rangle,$$

where R^j_+s and $R^j_\oplus$s are crisp relations of the same arity, satisfying $R^j_+ \subseteq R^j_\oplus$. □

Let R be a relation[5] with it's approximations R_{S^+} and $R_{S\oplus}$ represented in an approximate database D. Note that R is available only through its approximations and is not itself stored in D.

We assume a similarity space $S = \langle U, \sigma \rangle$ and the ability to verify whether $\sigma(x, y)$ holds for tuples x and y which are stored in D.

[5] Of course, any relation is a set of tuples, so our previous considerations apply here, too.

Consider first a simpler case of constraints referring to approximations only. Such constraints can be directly represented in the database since approximations are represented as database relations. The requirements $R_{S^+} \subseteq R_{S^\oplus}$ and $R_{S^+} \subseteq (R_{S^+})_{S^+}$ are examples of such constraints. In this case, both the lower and upper approximations, R_{S^+} and $R_{S^\oplus}$ can be computed according to Proposition 1, since we assume that σ can be verified on elements which are stored in D.

A more complicated case arises when an approximation constraint refers to R, the crisp relation being approximated, since R is not stored in D. In such cases, one will often need to enforce meta-constraints, i.e., constraints that have to be ensured by database designers, and which cannot explicitly be represented or computed in an approximate database.

Let us start with the requirement that $R_{S^+} \subseteq R$ holds. In order to preserve its meaning in D, one has to ensure the following meta-constraint:

> the lower approximation $R_{S^+} \subseteq R$ can only contain those tuples which are known to satisfy R.

The requirement $R \subseteq (R_{S^\oplus})_{S^+}$ is more problematic. In some cases it can be replaced by a constraint that does not refer to R directly. For example, using our analogy to modal logic ($R \rightarrow \Box\Diamond R$), it is well known that in the presence of reflexivity and transitivity, this axiom can be replaced by the property called 5.

The corresponding similarity relation for 5 is known to be Euclidean, i.e., it satisfies $\forall x, y, z[(\sigma(x, y) \wedge \sigma(x, z)) \rightarrow \sigma(y, z)]$ (see, e.g., [2, 3, 9]). In the language of approximate constraints, this would be expressed as $R_{S^\oplus} \subseteq (R_{S^\oplus})_{S^+}$, which refers to R via approximations only. In fact, in the presence of this property, transitivity is no longer required, since any Euclidean and reflexive relation is also both symmetric and transitive.[6]

In order to preserve the symmetry requirement on σ, one has to ensure that the following meta-constraint is preserved:

> all tuples satisfying R are to be included in the lower approximation of the upper approximation $R_{S^\oplus}$.

In general, it is difficult to ensure this meta-constraint. One alternative to the meta-constraint would be to store, for any relation R, R_{S^+} together with $(-R)_{S^+}$. For any database update adding to or deleting a tuple of type compatible with tuples in R from the database, one would then have to check whether the approximations still satisfy the conditions of Proposition 1.

This technique is tractable, but expensive, as any database update might cause an integrity check. One could also apply techniques based on the static verification of database transactions which, in many cases, would result in much

[6] This easily follows from the well-known fact that modal logic $KT5$ characterized with a reflexive and Euclidean accessibility relation is the same as $S5$, where the accessibility relation is reflexive, symmetric and transitive - see, e.g., [3].

more efficient solutions to this problem. A technique which can be applied in this context was developed in [10].

6 Conclusions

Assuming the use of approximate databases as our starting point which appeal to a generalization of indiscernibility relations to tolerance or similarity spaces, we have proposed a set of techniques which permit the expression of approximation constraints and similarity constraints. We provide a method to automatically translate between the two and show how intuitions from modal logic and modal correspondence theory can be put to good use, not only in acquiring insight as to the interaction between similarity and approximation, but in providing more efficient means of enforcing such constraints in approximate databases.

References

1. W. Ackermann.
   Untersuchungen über das eliminationsproblem der mathematischen logik.
   Mathematische Annalen, 110:390–413, 1935.
2. R. A. Bull and K. Segerberg.
   Basic modal logic.
   In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, pages 1–88. D. Reidel Pub. Co., 20198402.
3. B. F. Chellas.
   Modal Logic - an Introduction.
   Cambridge University Press, 1980.
4. P. Doherty, J. Kachniarz, and A. Szałas.
   Using contextually closed queries for local closed-world reasoning in rough knowledge databases.
   In *[13]*, 2003.
5. P. Doherty, W. Łukaszewicz, A. Skowron, and A. Szałas.
   Approximation transducers and trees: A technique for combining rough and crisp knowledge.
   In S. K. Pal, L. Polkowski, and A. Skowron, editors, *Rough-Neuro Computing: Techniques for Computing with Words*, Heidelberg, 2003. Springer–Verlag.
6. P. Doherty, W. Łukaszewicz, and A. Szałas.
   Computing circumscription revisited.
   Journal of Automated Reasoning, 18(3):297–336, 1997.
7. P. Doherty, W. Łukaszewicz, and A Szałas.
   Information granules for intelligent knowledge structures.
   In *Proc. 9th International Conference on rough sets, fuzzy sets, data mining and granular computing*, LNCS. Springer-Verlag, 2003.
8. P. Doherty, W. Łukaszewicz, and A Szałas.
   Tolerance spaces and approximative representational structures.
   In *Proceedings of 26th German Conference on Artificial Intelligence*. Springer-Verlag, 2003.

9. G. E. Hughes and M. J. Cresswell.
An Introduction to Modal Logic.
Methuen and Co. Ltd., London, New York, 1968.
10. J. Kachniarz and A. Szałas.
On a static approach to verification of integrity constraints in relational databases.
In E. Orłowska and A. Szałas, editors, *Relational Methods for Computer Science Applications*, pages 97–109. Springer Physica-Verlag, 2001.
11. C-J Liau.
An overview of rough set semantics for modal and quantifier logics.
Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 8(1):93–118, 2000.
12. A. Nonnengart, H.J. Ohlbach, and A. Szałas.
Elimination of predicate quantifiers.
In H.J. Ohlbach and U. Reyle, editors, *Logic, Language and Reasoning. Essays in Honor of Dov Gabbay, Part I*, pages 159–181. Kluwer, 1999.
13. S. K. Pal, L. Polkowski, and A. Skowron, editors.
Rough-Neuro Computing: Techniques for Computing with Words.
Springer–Verlag, Heidelberg, 2003.
14. Z. Pawlak.
Rough Sets. Theoretical Aspects of Reasoning about Data.
Kluwer Academic Publishers, Dordrecht, 1991.
15. A. Skowron and J. Stepaniuk.
Tolerance approximation spaces.
Fundamenta Informaticae, 27:245–253, 1996.
16. R. Słowiński and D. Vanderpooten.
Similarity relation as a basis for rough approximations.
In P. Wang, editor, *Advances in Machine Intelligence & Soft Computing*, pages 17–33, Raleigh NC, 1997. Bookwrights.
17. R. Słowiński and D. Vanderpooten.
A generalized definition of rough approximations based on similarity.
IEEE Trans. on Data and Knowledge Engineering, 12(2):331–336, 2000.
18. A. Szałas.
On the correspondence between modal and classical logic: An automated approach.
Journal of Logic and Computation, 3:605–620, 1993.
19. J. Van Benthem.
Correspondence theory.
In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, pages 167–247. D. Reidel Pub. Co., 1984.