Fundamenta Informaticae ?? (2006) 1–15
IOS Press

A Correspondence Framework between Three-Valued Logics and Similarity-Based Approximate Reasoning

Patrick Doherty[*]
Department of Computer and Information Science
Linköping University, Linköping, Sweden
patdo@ida.liu.se

Andrzej Szałas[†]
Department of Computer and Information Science
Linköping University, Linköping, Sweden,
and
University of Economics and Computer Science
Olsztyn, Poland
andsz@ida.liu.se

Abstract. This paper focuses on approximate reasoning based on the use of similarity spaces. Similarity spaces and the approximated relations induced by them are a generalization of the rough set-based approximations of Pawlak [17, 18]. Similarity spaces are used to define neighborhoods around individuals and these in turn are used to define approximate sets and relations. In any of the approaches, one would like to embed such relations in an appropriate logic which can be used as a reasoning engine for specific applications with specific constraints. We propose a framework which permits a formal study of the relationship between approximate relations, similarity spaces and three-valued logics. Using ideas from correspondence theory for modal logics and constraints on an accessibility relation, we develop an analogous framework for three-valued logics and constraints on similarity relations. In this manner, we can provide a tool which helps in determining the proper three-valued logical reasoning engine to use for different classes of approximate relations generated via specific types of similarity spaces. Additionally, by choosing a three-valued logic first, the framework determines what constraints would be required on a similarity relation and the

Address for correspondence: Department of Computer and Information Science, Linköping University, Linköping, Sweden
[*]Supported in part by the Wallenberg Foundation.
[†]Supported in part by the MNiI grant 3 T11C 023 29.

approximate relations induced by it. Such information would guide the generation of approximate relations for specific applications.

Keywords: approximate reasoning, rough sets, similarity relation, three-valued logics

1. Introduction

There is a natural generalization of classical logical relations studied in rough set theory [17, 18], where a rough relation is characterized by two crisp relations, one representing an upper approximation to the original implicit relation and the other representing a lower approximation. Upper and lower approximations are generated using an indiscernibility relation which is an equivalence relation among individuals in the domain of discourse and which partitions the universe into indiscernibility classes relative to a set of features which parameterize the indiscernibility relation.

There is a natural generalization of this idea [3, 8, 10, 13, 20, 21, 22], where one considers a similarity relation among individuals instead of an indiscernibility relation and generates similarity-based neighborhoods instead of families of elementary sets. The similarity-based neighborhoods provide a covering of the domain of discourse and via such neighborhoods, one can define upper and lower approximations. To distinguish this generalization from that of classical rough set theory, we will consider and use the name *approximate relations* as a generalization of rough relations and *similarity spaces* [8, 10] as a generalization of of families of elementary sets. Approximate relations and similarity spaces have been shown to be quite versatile in many application areas requiring the use of approximate knowledge structures [4, 6, 7, 9].

Traditional applications involving the use of rough sets or relations are often table based. From tables of relational data one often generates classifiers represented as rough relations for use in applications. These classifiers are then often combined into rules which are used for reasoning about the application domain in an informal manner. The underlying logic for reasoning with such rule sets is often only implicitly defined. This would also be the starting point for using approximate relations, where one assumes a similarity relation as a basis for defining neighborhoods which in turn are used to define upper and lower approximations for each relation. Two important issues arise in the context of this generalization, one common with the use of traditional rough sets, and the other derivative of the generalization.

1. What is the appropriate logic and inference mechanism that should be used for reasoning with approximate relations?

2. Given that there is a wide spectrum of choice concerning constraints that could be placed on the underlying similarity relation used to generate approximate relations, how can a formal correspondence be set up and shown to hold between these constraints and the logic that will be used to reason with the approximate relations generated from the similarity relation?

When generalizing to approximate relations, it turns out that there are many choices that can be made concerning the constraints one might want to impose on the underlying similarity relation used to define upper and lower approximations for such relations. For example, one might not want the relation to be transitive since similar objects don't naturally chain in a transitive manner. Many of these issues are discussed in the context of rough sets (see, e.g., [20, 21, 22]). Whatever choices are made though, one wants to ensure that these constraints are guaranteed to hold in the inference machinery used to reason with such relations.

With regard to the appropriate logic and inference machinery to use for reasoning with approximate relations, there are also many choices. There is a nice correspondence though, between three-valued logics and approximate relations. If an individual is in the lower approximation of an approximate relation, we say that that relation holds for the individual; if the individual is in the set generated by the difference between the upper and lower approximation, we say that it is unknown whether the relation holds for the individual; if the individual is in the set generated by the difference between the domain of discourse and the upper approximation, we say that it is false that the relation holds for the individual. In fact, when it can only be ascertained that the individual is in the boundary region between the upper and lower approximation, thus its membership is unknown, there is a nonmonotonic flavor to its status since with additional information one may be able to determine whether it is in the lower approximation or outside the upper approximation. Based on this correspondence, we will focus on the space of three-valued logics as providing alternatives for appropriate logics for reasoning with approximate relations.

Just as we have a choice regarding constraints on an underlying similarity relation used to define approximate relations in an application domain, we will also have a choice as to the proper three-valued logic to embed our relations in for reasoning in the application. What we would like is a formal framework for studying the relation between approximate relations, similarity and three-valued logic. As a derivative of the framework, we would also like a means of choosing the most appropriate three-valued logic for reasoning with a set of approximate relations in an application given a similarity relation with associated constraints. In order to answer the questions above and to define the framework, we will approach the problem by using methods similar to those used for the analysis of modal logics using correspondence theory [24].

Possible worlds semantics allows one to make interesting connections between intensional axioms in modal logic and properties of the accessibility relation between worlds in those frames which validate such axioms. The discovery of this connection clarified the relation between the many intensional axiomatic theories that had been proposed and provided an intuitive semantic basis for comparison between such theories. Correspondence theory arose from this context with the key observation that modal formulas express certain constraints on the accessibility relation in frames and these constraints could be represented in 1st-order or 2nd-order classical logical formulas.

For example,

$$\Box p \to p \quad \text{becomes} \quad \forall y(R(x,y) \to P(y)) \to P(x) \tag{1}$$

$$\begin{aligned} \Box\Diamond p \to \Diamond\Box p \quad \text{becomes} \quad & \forall y(R(x,y) \to \exists z(R(y,z) \land P(z))) \\ & \to \exists y(R(x,y) \land \forall z(R(y,z) \to P(z))) \end{aligned} \tag{2}$$

The parameter x refers to the current world where the formula is being evaluated and the unary predicate constant P corresponds to the propositional variable p and denotes the set of worlds where p holds.

Since the relational pattern of the accessibility relations is primary and the actual assignments to the current world and to propositional variables are irrelevant in studying these patterns, a further abstraction is made on the denotations of the associated predicate constants, where one quantifies over such unary predicate constants. For example, the 1st-order transcription of the McKinsey axiom (2) is replaced by a 2nd-order transcription,

$$\Box\Diamond p \to \Diamond\Box p \quad \text{becomes} \quad \forall P \forall x[\forall y(R(x,y) \to \exists z(R(y,z) \wedge P(z))) \\ \to \exists y(R(x,y) \wedge \forall z(R(y,z) \to P(z)))] \tag{3}$$

When moving from modal formulas to classical formulas, an obvious question of definability[1] arises: which modal formulas define 1st-order relational conditions?

The 2nd-order transcription of the McKinsey axiom is not reducible to a logically equivalent 1st-order formula while that of $\Box p \to p$ is, and correspondence theory studies the reasons why. In some cases, this question can be answered by using quantifier eliminations techniques [5, 12, 23]. In these cases one can automatically generate 1st-order equivalences of the 2nd-order transcriptions. Such elimination techniques will also be used in this paper.

Symmetrically, when moving in the other direction from classical formulas to modal formulas, another obvious question arises:

Which 1st-order relational conditions are modally definable?

These are some of the many questions studied in the area of correspondence theory for modal logics.

In a similar manner, we will view three-valued axioms in different three-valued logics as expressing certain constraints on the underlying similarity relation which is used to define the upper and lower approximations on a base set of approximate relations intended to be used in our applications. Just as one requires an accessibility relation based semantics in classical logic for transcribing modal formulas into, we will require a similarity relation based semantics for three-valued logics for transcribing three-valued axioms into classical formulas representing constraints on the underlying similarity relation.

In this manner, we will have a framework for studying correspondences between three-valued logical axioms and their equivalent forms in the similarity based relational language. Given a similarity relation in an application and the definitions of the upper and lower approximations of specific relations generated from the similarity relation, the constraints on the similarity relation can be expressed in an appropriate three-valued logic as axioms and the relations can be embedded in the logic to provide the proper inference machinery for reasoning correctly with the approximate relations. By correct, we mean that the inference machinery does not violate any of the constraints associated with the underlying similarity relation used to generate the approximate relations to be used in the application.

2. A Motivating Example

Let us assume an application domain where an autonomous robot is operating in a chemical plant that has been subjected to a terrorist attack with fires and chemical spills in different parts of the plant as a result of the attack. Different chemicals and gases are also stored in tanks and cannisters. The robot is moving around in the plant trying to determine whether different regions are safe enough to send in emergency services personnel. For the application, we can assume an underlying approximate model regarding safe situations has been generated which consists of a number of approximate relations generated from a similarity relation.

[1] As well as a related question of complexity.

Table 1. Truth tables for $\neg, \sim, \vee, \wedge, \rightarrow$ of strong Kleene logic $K_3^{\sim}$.

$\neg v, \sim v$

v	$\neg v$	$\sim v$
T	F	F
F	T	T
U	U	T

$u \vee w$

$u \backslash w$	T	F	U
T	T	T	T
F	T	F	U
U	T	U	U

$u \wedge w$

$u \backslash w$	T	F	U
T	T	F	U
F	F	F	F
U	U	F	U

$u \rightarrow w$

$u \backslash w$	T	F	U
T	T	F	U
F	T	T	T
U	T	U	U

The vocabulary of approximate relations includes, $Temperature$, $Pressure$, $LiquidLevel$, and $Safe$ parameterized with a situation variable s and other constants related to specific states of regions in the chemical plant. We can assume that when asking questions about these relations, the result may be true, false or unknown based on the following relations between partitions of an approximate relation and a three-valued semantics for atoms which will hold in any choice of logic in the three-valued logical space.

- If a tuple $\bar{x}$ is in the lower approximation of a relation R then $R(\bar{x})$ is true;
- if $\bar{x}$ is not a member of the upper approximation then $\neg R(\bar{x})$ is true ($R(\bar{x})$ is false);
- if $\bar{x}$ is not a member of the lower approximation and it is a member of the upper approximation then $R(\bar{x})$ is unknown.

The robot also has a number of decision rules in its database including the following which the robot can use to determine whether a situation s is safe or not:

$$Temperature(s, \mathsf{low}) \wedge Pressure(s, \mathsf{medium}) \wedge \rightarrow Safe(s) \tag{4}$$

$$Temperature(s, \mathsf{low}) \wedge LiquidLevel(s, \mathsf{medium}) \rightarrow Safe(s). \tag{5}$$

Assuming that all relations occurring in these rules are approximate and can take truth values T, F, U, how are we to determine the proper semantics for evaluating whether a particular situation is safe based on a context involving temperature, pressure and liquid level? In addition, how would we guarantee that the semantics is correct relative to the constraints placed on the underlying similarity relation used to define the base approximate relations?

For this example, let us choose the Strong Kleene Logic with external negation (see, e.g., [2]) which we will denote as $K_3^{\sim}$. $K_3^{\sim}$ truth tables for $\neg, \sim, \vee, \wedge, \rightarrow$ are provided in Table 1.

We could then calculate the truth value of $Safe(s)$ by first evaluating each rule, assigning to $Safe(s)$ the value equal to the truth value of the rule's body. We would then choose as the result for $Safe(s)$ the disjunction of all values obtained in all rules where $Safe(s)$ is a consequent. For example, suppose that the truth values of the following is asserted:

$Temperature(s, \mathsf{low}) = $ T, $LiquidLevel(s, \mathsf{medium}) = $ T and $Pressure(s, \mathsf{medium}) = $ U.
Then rule (4) assigns U to $Safe(s)$, while rule (5) assigns T . The disjunction U $\vee$ T is T, therefore we evaluate $safe(s)$ to be T.

The above calculations are straightforward once one has chosen the underlying three-valued logic. However the choice of strong Kleene logic is somehow arbitrary. There are many other choices in the space of three-valued logics (see, e.g., [2, 19]). This is a general problem when reasoning with three-valued logics, but in this case it is indeed additionally problematic due to the implicit use of approximate relations in the logic. One could, for example, choose a three-valued logic, where U $\rightarrow$ F is F. This choice is sometimes used as the basis for a semantics of deductive databases with negation (see, e.g., [11], or a discussion about alternatives provided in [19]).

Even though this choice might be intuitive for a semantics for deductive databases, it would not necessarily be an appropriate choice in other application domains. As an example, consider the formula:

$$\big(FourWheels(\mathsf{a}) \wedge OnRoad(\mathsf{a})\big) \rightarrow Car(\mathsf{a}). \tag{6}$$

Assume that an agent determines that the value of $Car(\mathsf{a})$ is F and is unable to determine whether a has four wheels, i.e., the value of $FourWheels(\mathsf{a})$ is U, or whether a is on the road, i.e., the value of $OnRoad(\mathsf{a})$ is U . The assumption that U $\rightarrow$ F is F would make formula (6), F. On the other hand, it is intuitive to assert that formula (6) is U in this example, as would be the case if $K_3^{\sim}$ was used as a basis for the semantics.

Even if the choice that U $\rightarrow$ F is F is doubtful in many applications, it is intuitive in other contexts and might reflect a particular form of approximate reasoning. The problem is to determine if such a choice is not only intuitive relative to the particular semantic intuitions of the three-valued logic, but also whether that three-valued logic reflects the underlying similarity constraints associated with the approximate relations used in the logic.

3. Preliminaries

3.1. Similarity Spaces and Approximations

An important basis for our approach involves the use of similarity spaces [10]. Technically, a similarity space defines a classification of a universe of individuals relative to neighborhoods generated through the use of a similarity relation among individuals. In order to provide the means to represent arbitrary notions of similarity in a universe of individuals, similarity relations initially have no initial constraints.

Definition 3.1. [2] By a *similarity space* we mean any pair $\langle U, \sigma\rangle$, where U is a set and $\sigma \subseteq U \times U$. By a *neighborhood* of u wrt σ we mean $n^{\sigma}(u) \stackrel{\text{def}}{=} \{u' \in U \mid \sigma(u, u') \text{ holds}\}$. For $A \subseteq U$, the *lower and upper approximation of* A *wrt* σ, denoted respectively by A^{+}_{σ} and $A^{\oplus}_{\sigma}$, are defined by

$$A^{+}_{\sigma} = \{u \in U : n^{\sigma}(u) \subseteq A\}, \quad A^{\oplus}_{\sigma} = \{u \in U : n^{\sigma}(u) \cap A \neq \emptyset\}.$$

[2] For the purposes of this paper, we simplify the definition of similarity spaces used in [10]. There, a similarity space is defined as a tuple $\langle U, \tau, p\rangle$ where τ is a similarity function, $\tau : U \times U \longrightarrow [0, 1]$ over the real interval, and $p \in [0, 1]$ is a similarity threshold used together with τ to define a parameterized similarity relation $\sigma^{p} \stackrel{\text{def}}{=} \{\langle x, y\rangle \mid \tau(x, y) \geq p\}$.

We also define $A^-_\sigma \stackrel{\mathrm{def}}{=} -A^\oplus_\sigma$, $A^\ominus_\sigma \stackrel{\mathrm{def}}{=} -A^+_\sigma$ and $A^\pm_\sigma \stackrel{\mathrm{def}}{=} A^\oplus_\sigma \cap A^\ominus_\sigma$. ◁

Let $S = \langle U, \sigma \rangle$ be a similarity space and let $A \subseteq U$. Then an alternative way to define approximations on A is

$$\begin{aligned}
A^+_S &= \{a \in A \mid \forall b\,[\sigma(a,b) \rightarrow b \in A]\},\\
A^\oplus_S &= \{a \in A \mid \exists b\,[\sigma(a,b) \wedge b \in A]\},\\
A^-_S &= \{a \in A \mid \forall b[\sigma(a,b) \rightarrow b \notin A]\},\\
A^\pm_S &= \{a \in A \mid \exists b, c[\sigma(a,b) \wedge b \in A \wedge\ \sigma(a,c) \wedge c \notin A]\}.
\end{aligned}$$

In practice, it is assumed that one or more similarity spaces are provided as part of the application domain, where a similarity space may be associated with each predicate in the application vocabulary. For the purposes of this paper, we will assume a single similarity space for all predicates in the application vocabulary.

3.2. Elimination of Second-Order Quantifiers

Quantifier elimination is an important part of the framework. As a basis for doing this, we will use the following lemma due to Ackermann [1] (see also, e.g., [5, 23]), 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, replace the variables of $\overline{x}$ in Φ (and bound variables are renamed if necessary).

Lemma 3.1. 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. 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. ◁

The lemma above shows how certain types of existentially quantified 2nd-order formulas can be reduced to logically equivalent 1st-order formulas. In fact, universally quantified 2nd-order formulas can also be reduced by negating, reducing, and then negating the result. In order to transform formulas to one of the forms required in Lemma 3.1, one can apply the DLS algorithm [5] (for an online implementation of the algorithm see [14]).

More advanced correspondences can be computed by applying the fixpoint theorem of [16]. In this case, the algorithm would reduce a certain class of 2nd-order formulas to logically equivalent fixpoint formulas. For a discussion of various approaches to second-order quantifier elimination see [15].

4. Similarity-Based Semantics for Three-valued Logics

In this section we define a translation of three-valued logics into classical first-order logic and define a similarity based semantics for three-valued logics.

Definition 4.1. *Formulas of propositional three-valued logics* are defined inductively as the smallest set containing a denumerable set of *propositional variables* and closed under applications of *propositional connectives* $\neg$ (strong negation), $\sim$ (weak negation), $\vee$ (disjunction), $\wedge$ (conjunction) and $\rightarrow$ (implication). ◁

In what follows we will use three logical values: T (true), F (false) and U (the third value). The similarity-based semantics for three-valued logics is given by the following definition.

Definition 4.2. Let V be the set of propositional three-valued variables and let $\mathcal{S} = \langle U, \sigma \rangle$ be a similarity space. By a *valuation* of propositional three-valued variables wrt objects of U we understand any function $v : V \times U \longrightarrow \{\mathrm{T}, \mathrm{F}, \mathrm{U}\}$.

Given a valuation v of propositional variables, we define a valuation $v^{\mathcal{S}}$ for arbitrary three-valued formulas as follows:

$$v^{\mathcal{S}}(p, x) \stackrel{\text{def}}{=} \begin{cases} \mathrm{T} & \text{iff for all } y \in U, \text{ if } \sigma(x, y) \text{ then } v(p, y) = \mathrm{T} \\ \mathrm{F} & \text{iff for all } y \in U, \text{ if } \sigma(x, y) \text{ then } v(p, y) = \mathrm{F} \\ \mathrm{U} & \text{otherwise,} \end{cases}$$

where $p \in V$

$$v^{\mathcal{S}}(\neg\alpha, x) \stackrel{\text{def}}{=} \begin{cases} \mathrm{T} & \text{iff for all } y \in U, \text{ if } \sigma(x, y) \text{ then } v^{\mathcal{S}}(\alpha, y) = \mathrm{F} \\ \mathrm{F} & \text{iff for all } y \in U, \text{ if } \sigma(x, y) \text{ then } v^{\mathcal{S}}(\alpha, y) = \mathrm{T} \\ \mathrm{U} & \text{otherwise} \end{cases}$$

$$v^{\mathcal{S}}(\sim \alpha, x) \stackrel{\text{def}}{=} \begin{cases} \mathrm{T} & \text{iff } v^{\mathcal{S}}(\alpha, x) \in \{\mathrm{F}, \mathrm{U}\} \\ \mathrm{F} & \text{otherwise} \end{cases}$$

$$v^{\mathcal{S}}(\alpha \vee \beta, x) \stackrel{\text{def}}{=} \begin{cases} \mathrm{T} & \text{iff } v^{\mathcal{S}}(\alpha, x) = \mathrm{T} \text{ or } v^{\mathcal{S}}(\beta, x) = \mathrm{T} \\ \mathrm{F} & \text{iff } v^{\mathcal{S}}(\alpha, x) = \mathrm{F} \text{ and } v^{\mathcal{S}}(\beta, x) = \mathrm{F} \\ \mathrm{U} & \text{otherwise} \end{cases}$$

$$v^{\mathcal{S}}(\alpha \wedge \beta, x) \stackrel{\text{def}}{=} \begin{cases} \mathrm{T} & \text{iff } v^{\mathcal{S}}(\alpha, x) = \mathrm{T} \text{ and } v^{\mathcal{S}}(\beta, x) = \mathrm{T} \\ \mathrm{F} & \text{iff } v^{\mathcal{S}}(\alpha, x) = \mathrm{F} \text{ or } v^{\mathcal{S}}(\beta, x) = \mathrm{F} \\ \mathrm{U} & \text{otherwise} \end{cases}$$

$$v^{\mathcal{S}}(\alpha \rightarrow \beta, x) \stackrel{\text{def}}{=} \begin{cases} \mathrm{T} & \text{iff } v^{\mathcal{S}}(\alpha, x) = \mathrm{F} \text{ or } v^{\mathcal{S}}(\beta, x) = \mathrm{T} \\ \mathrm{F} & \text{iff } v^{\mathcal{S}}(\alpha, x) = \mathrm{T} \text{ or } v^{\mathcal{S}}(\beta, x) = \mathrm{F} \\ \mathrm{U} & \text{otherwise.} \end{cases}$$

A satisfiability relation $\models_3$ is now defined by

$$\mathcal{S}, v, x \models_3 \alpha \text{ iff } v^{\mathcal{S}}(\alpha, x) = \mathrm{T},$$

where x represents an object of the domain of the similarity space $\mathcal{S}$.

By $\mathcal{S} \models_3 \alpha$ we mean that for all v and x, we have that $\mathcal{S}, v, x \models_3 \alpha$. ◁

In the rest of the paper we will use the symbols $\models_3$ and $\models$ to denote the satisfaction relation of a three-valued logic and the classical first (and sometimes second-order) logic, respectively.

Remark 4.1. For the results in this paper to apply, there is some flexibility in the definition of the valuation function. The following restrictions are necessary:

1. the definitions of internal or strong ($\neg$) or external or weak ($\sim$) negations provided in definition 4.2 must be used; and

2. the *truth* clauses for the connectives $\vee, \wedge, \rightarrow$ must also be used. One is free to define the falsity and unknown clauses in a manner reflecting a variety of semantic intuitions in the literature.

Note also that the true and falsity clauses reflect the semantics of classical (two-valued) propositional logic, and therefore frequently appear in applications. $\triangleleft$

The similarity-based semantics for three-valued logics is given via a translation function $T(\alpha, x)$, where α is a three-valued formula and $x \in U$, with the intuitive meaning that formula α is T for object x.

Definition 4.3. Let $\mathcal{S} = \langle U, \sigma \rangle$ be a similarity space and let α be a formula of three-valued logic. Let $\Sigma_{\mathcal{S}}$ be a first-order signature with domain U, unary relation symbols $P_1, \ldots P_k$, corresponding to propositional variables $p_1, \ldots p_k$ and a binary relation symbol σ corresponding to σ in $\mathcal{S}$.

The *translation* $T(\alpha, x)$ *of formulas of three-valued logic into classical first-order logic over signature* $\Sigma_{\mathcal{S}}$ is defined inductively as follows:

- $T(p, x) \stackrel{\text{def}}{=} \forall y[\sigma(x, y) \rightarrow P(y)]$, where p is a propositional variable and P is a unary relation symbol corresponding to p,
- $T(\neg\alpha, x) \stackrel{\text{def}}{=} \forall y[\sigma(x, y) \rightarrow \neg T(\alpha, y)]$,
- $T(\sim \alpha, x) \stackrel{\text{def}}{=} \neg T(\alpha, x)$,
- $T(\alpha \vee \beta, x) \stackrel{\text{def}}{=} T(\alpha, x) \vee T(\beta, x)$,
- $T(\alpha \wedge \beta, x) \stackrel{\text{def}}{=} T(\alpha, x) \wedge T(\beta, x)$,
- $T(\alpha \rightarrow \beta, x) \stackrel{\text{def}}{=} T(\alpha, x) \rightarrow T(\beta, x)$. $\triangleleft$

The intuitive meaning of $T(P, x)$, where P is atomic is that $x \in P^+_\sigma$, of $T(\neg P, x)$ is that $x \in P^-_\sigma$, while of $T(\sim P, x)$ is that $x \in P^\ominus_\sigma$. This observation relates the definition of the translation function in Definition 4.3 to the similarity-based partitions of sets in Definition 3.1.

Note also that this translation basically reflects, e.g., the one given in [2] (see Section 12.5 there) for the strong Kleene logic, $K_3^\sim$.

We now have the following proposition.

Proposition 4.1. let $\mathcal{S} = \langle U, \sigma \rangle$ be a similarity space, v be a valuation of propositional variables and α be a formula of three-valued logic. Let $M_{\mathcal{S}} = \langle U, P_1, \ldots P_k, \sigma \rangle$ be a first-order model over $\Sigma_{\mathcal{S}}$ where $\sigma \in M_{\mathcal{S}}$ is interpreted as $\sigma \in \mathcal{S}$, and the interpretations of the $P_1, \ldots P_k$ are defined by

$$P_i(x) \stackrel{\text{def}}{=} \begin{cases} \text{T} & \text{iff } v^{\mathcal{S}}(p_i, x) = \text{T} \\ \text{F} & \text{otherwise} \end{cases}$$

and the interpretations of any free variables are individuals in U. Then $\mathcal{S} \models_3 \alpha$ iff $M_{\mathcal{S}} \models T(\alpha, x)$. $\triangleleft$

5. Correspondences between Three-valued Logics and Similarities

5.1. The Method

As stated in the introduction, correspondences between three-valued logics and approximate relations based on a specific similarity relation can be given in both directions. The general techniques used to compute correspondences between modal logics and similarity constraints are those described in [23], but here we deal with three-valued rather than modal logics.

In the similarity to three-valued logic direction:

1. We assume an application has a similarity space S and a set of approximate relations derived from the similarity space. Associated with the similarity relation σ in S is a set of constraints Δ such as seriality, transitivity, etc. Formulas in Δ are characterized in first- or second-order logic.
2. The idea is to choose a three-valued logic where the constraints in Δ can be translated into the three-valued logic as axioms and the the approximate relations can be embedded into the logic using the definitions in section 3.1. Note that it is not always the case that such a mapping exists.
3. The correspondence framework formally shows that the three-valued logic is correct relative to the similarity relation chosen and any inferences made in the logic using the defined approximate relations necessarily obeys the constraints in Δ if one has successfully mapped each formula in Δ into an axiom in the three-valued logic.

In the three-valued logic to similarity direction:

1. Given a three-valued logic with associated axioms, generate the constraints on the similarity relation, Δ, that would be required to enable the use of the logic.
2. To do this:
   (a) choose a three-valued axiom α and compute its translation $T(\alpha, x)$
   (b) consider the formula $\forall \bar{P} \forall x\, [T(\alpha, x)]$, where $\bar{P}$ are all unary relation symbols (corresponding to propositional variables) in $T(\alpha, x)$
   (c) eliminate second-order variables $\bar{P}$, if possible.

If the second-order quantifier elimination is successful in step 2c, then the resulting formula uses only the non-logical symbols σ, $=$ and is logically equivalent to the initial logical translation of the chosen axiom. The quantifier elimination step can be automated using the algorithm given in [23] or its generalization known as the DLS algorithm of [5]. An implementation of DLS, generalized by the fixpoint theorem of [16], is available online (see [14]). There are also other applicable methods, such as the SCAN algorithm provided in [12], which may also be used (for an overview of known techniques, see [15]).

5.2. Some Correspondences

Table 2 provides examples of correspondences between three-valued axioms and similarities.

The following examples illustrate calculations leading to some of the correspondences listed in Table 2. Note that such calculations can be made automatically using the DLS algorithm [14] quantifier elimination algorithm.

Table 2. Correspondences for some axioms.

Axiom	The corresponding constraint on σ
$\sim (p \wedge \neg p)$	$\forall x \exists y \sigma(x, y)$ – seriality
$p \rightarrow \sim \neg p$	$\forall x \exists y \sigma(x, y)$ – also seriality
$\sim \neg p \rightarrow \neg\neg p$	$\forall x, y, z[(\sigma(x, y) \wedge \sigma(x, z)) \rightarrow \sigma(y, z)]$ – euclidicity
$p \rightarrow \neg \sim p$	$\forall x, y, z[(\sigma(x, y) \wedge \sigma(y, z)) \rightarrow \sigma(x, z)]$ – transitivity
$\sim \neg \sim\sim p \rightarrow \neg\neg p$	$\forall x, y, z[(R(x, y) \wedge R(x, z)) \rightarrow \exists w[R(y, w) \wedge R(z, w)]]$ – directedness
$p \rightarrow \neg\neg p$	$\forall x, y[\sigma(x, y) \rightarrow \exists z(\sigma(x, z) \wedge \sigma(y, z))]$
$p \vee \neg p$	$\forall x, y, z[(\sigma(x, y) \wedge \sigma(x, z)) \rightarrow y = z]$ – determinism
$p \vee \sim p$	T – neutral wrt σ

Example 5.1. Consider $\sim (p \wedge \neg p)$. According to the method discussed in Section 5, we first consider

$$\forall P \forall x[T(\sim (p \wedge \neg p), x)], \tag{7}$$

where T is defined as in Definition 4.3. According to Definition 4.3,

$$T(\sim (p \wedge \neg p), x) = \neg T(p \wedge \neg p, x) = \neg\{T(p, x) \wedge T(\neg p, x)\} = \\ \neg\{\forall y[\sigma(x, y) \rightarrow P(y)] \wedge \forall z[\sigma(x, z) \rightarrow \neg P(z)]\}.$$

Consequently, formula (7) is equivalent to

$$\forall P \forall x \neg\{\forall y[\sigma(x, y) \rightarrow P(y)] \wedge \forall z[\sigma(x, z) \rightarrow \neg P(z)]\},$$

i.e., to $\neg\exists x \exists P\{\forall y[\sigma(x, y) \rightarrow P(y)] \wedge \forall z[\sigma(x, z) \rightarrow \neg P(z)]\}$ and further to

$$\neg\exists x \exists P\{\forall y[\sigma(x, y) \rightarrow P(y)] \wedge \forall z[P(z) \rightarrow \neg\sigma(x, z)]\}.$$

Applying Lemma 3.1 we conclude that this formula is equivalent to

$$\neg\exists x \forall y[\sigma(x, y) \rightarrow \neg\sigma(x, y)], \text{i.e., to } \forall x \exists z \sigma(x, z),$$

which asserts the seriality of σ.

It is interesting to note that the three-valued logic $K_3^{\sim}$, in fact corresponds to this requirement (see, e.g., [2]). Seriality is interesting in the sense that it corresponds to the requirement that the lower approximation of a relation is included in its upper approximation (see [10]), which would appear to be the weakest requirement one would associate with approximate relations. Therefore, $K_3^{\sim}$ can be considered as a base logic for approximate reasoning in a manner similar to the role the modal logic K serves as a base logic for all normal modal logics. ◁

Example 5.2. Consider the axiom $p \rightarrow \neg \sim p$. According to the method discussed in Section 5, we first consider

$$\forall P \forall x[T(p \rightarrow \neg \sim p, x)], \tag{8}$$

where T is defined as in Definition 4.3. According to Definition 4.3,

$$\begin{aligned}
&T(p \rightarrow \neg\sim p, x) = T(p, x) \rightarrow T(\neg\sim p, x) = \\
&\quad \forall y[\sigma(x, y) \rightarrow P(y)] \rightarrow \forall z[\sigma(x, z) \rightarrow \neg T(\sim p, z)] = \\
&\quad \forall y[\sigma(x, y) \rightarrow P(y)] \rightarrow \forall z[\sigma(x, z) \rightarrow \neg\neg T(p, z)] = \\
&\quad \forall y[\sigma(x, y) \rightarrow P(y)] \rightarrow \forall z[\sigma(x, z) \rightarrow \forall w[\sigma(z, w) \rightarrow P(w)]].
\end{aligned}$$

Consequently, formula (8) is equivalent to

$$\forall P \forall x \{\forall y[\sigma(x, y) \rightarrow P(y)] \rightarrow \forall z[\sigma(x, z) \rightarrow \forall w[\sigma(z, w) \rightarrow P(w)]]$$

i.e., to $\neg\exists P \exists x \forall y[\sigma(x, y) \rightarrow P(y)] \wedge \neg\forall z[\sigma(x, z) \rightarrow \forall w[\sigma(z, w) \rightarrow P(w)]]$ and applying the dual form of Lemma 3.1 to $\neg\exists x \neg\forall z[\sigma(x, z) \rightarrow \forall w[\sigma(z, w) \rightarrow \sigma(x, w)]]$, which is equivalent to

$$\forall x \forall z[\sigma(x, z) \rightarrow \forall w[\sigma(z, w) \rightarrow \sigma(x, w)]],$$

i.e., to the transitivity of σ, as indicated in Table 2. $\lhd$

5.3. Adding an I Operator to the Correspondence Framework

Observe that the language of three-valued logic and its translation allows one to refer to approximations of relations only. This choice is intuitively correct when we do not have access to crisp relations themselves. This is frequently the case with approximate databases, where one stores only approximations of relations.

On the other hand, referring only to approximations, we cannot capture important properties of similarities, such as reflexivity or symmetry. Therefore it is useful to introduce a new operator Ip, meaning "p itself", translated as $T(Ip, x) \stackrel{\text{def}}{=} P(x)$, where P is the relation corresponding to p. In the extended language we can easily express reflexivity or symmetry, respectively by $p \rightarrow Ip$ and $Ip \rightarrow \neg\neg p$. Of course, use of the I operator would require additional reasoning machinery to extend any three-valued logic used. This however, is a topic we will leave for another paper.

6. A Choice of a Three-Valued Logic Based on Similarity Constraints

It was mentioned previously that it is not always possible to provide a one-one mapping between constraints on similarity relations and axioms in a three-valued logic of choice, since there may be constraints on the similarity relation which can not be expressed solely using three-valued axioms. The following criteria may instead be used to guarantee weaker forms of correspondence when determining a choice of three-valued logic.

Definition 6.1. We shall say that a three-valued axiom α *is correct wrt a constraint* β on a similarity relation provided that $\models \beta \rightarrow [\forall \bar{P} \forall x T(\alpha, x)]$, where $\bar{P}$ consists of all unary relation symbols in $T(\alpha, x)$. It *captures the constraint* β provided that $\models [\forall \bar{P} \forall x T(\alpha, x)] \rightarrow \beta$. It *corresponds to the constraint* β iff it both captures and is correct wrt β. $\lhd$

Table 3. Correspondences for some approximate reasoning paradigms.

Axioms	Approximate reasoning
$p \rightarrow \sim \neg p$	approximate reasoning based on seriality only
$p \rightarrow Ip; Ip \rightarrow \neg\neg p$	tolerance spaces
$Ip \rightarrow \neg\neg p; \sim \neg p \rightarrow \neg\neg p; p \rightarrow \neg \sim p$	rough set based reasoning

Full correspondence is an ideal situation. However, in practice one need only require that axioms are correct wrt constraints on a similarity relation. It is desirable that they capture those constraints[3].

Table 3 provides some additional correspondences between three-valued axioms and the underlying similarity constraints used to characterize a number of approximate reasoning methods.

7. Conclusions

We have proposed a framework which permits a formal study of the relationship between approximate relations, similarity spaces and three-valued logics. This framework can be used in a number of ways. In the case where approximate relations are generated for applications using a base similarity or indiscernibility relation, we show how the framework could be used to choose an appropriate three-valued logic for use as a reasoning engine for such relations. The framework guarantees that the logic is sound (and sometimes complete) relative to the constraints associated with the underlying similarity or indiscernibility relation. Another way to use the framework is to first choose a three-valued logic which captures the intuitions one wishes to associate with a reasoning engine for a particular application, and then to automatically generate the constraints that a similarity relation would have to meet in order to use the logic. Generation of approximate relations for the application would then have to obey these constraints.

Much remains to be investigated. It would be interesting to do a thorough study of additional classes of three-valued logics and the constraints these would place on similarity relations. In the opposite direction, it would be interesting to investigate what common sets of constraints on similarity are used in existing applications and whether there are sound and complete three-valued logics that could be used as reasoning engines for such applications.

References

[1] Ackermann, W.: Untersuchungen über das Eliminationsproblem der mathematischen Logik, *Mathematische Annalen*, **110**, 1935, 390–413.

[2] Busch, D.: Sequent Formalizations of Three-Valued Logics, in: *Partiality, Modality and Nonmonotonicity* (P. Doherty, Ed.), CSLI Publications, 1996, 45–75.

[3] Demri, S., Orłowska, E.: *Incomplete Information: Structure, Inference, Complexity*, EATCS Monographs, Springer, 2002.

[3] This, however, sometimes cannot be achieved.

[4] Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Approximation Transducers and Trees: A Technique for Combining Rough and Crisp Knowledge, *Rough-Neuro Computing: Techniques for Computing with Words* (S. Pal, L. Polkowski, A. Skowron, Eds.), Cognitive Technologies, Cognitive Technologies, Springer–Verlag, 2003, 189–218.

[5] Doherty, P., Łukaszewicz, W., Szałas, A.: Computing Circumscription Revisited, *Journal of Automated Reasoning*, **18**(3), 1997, 297–336.

[6] Doherty, P., Łukaszewicz, W., Szałas, A.: Information Granules for Intelligent Knowledge Structures, *Proc. 9th Conf. RSFDGrC* (WangG., Q. Liu, Y. Yao, A. Skowron, Eds.), LNCS, vol. 2639 of *LNCS*, Springer-Verlag, 2003, 405–412.

[7] Doherty, P., Łukaszewicz, W., Szałas, A.: Similarity, Approximations and Vagueness, *Proc. RSFDGrC'05* (e. a. Slezak D., Ed.), LNAI, number 3641 in LNAI, Springer-Verlag, 2003, 541–550.

[8] Doherty, P., Łukaszewicz, W., Szałas, A.: Tolerance Spaces and Approximative Representational Structures, *Proc. 26th German Conf. on AI, KI'2003* (A. Günter, R. Kruse, B. Neumann, Eds.), LNAI, vol. 2821 of *LNAI*, Springer-Verlag, 2003, 475–489.

[9] Doherty, P., Łukaszewicz, W., Szałas, A.: Approximate Databases and Query Techniques for Agents with Heterogenous Perceptual Capabilities, *Proceedings of the 7th Int. Conf. on Information Fusion, FUSION'2004*, 2004, 175–182.

[10] Doherty, P., Szałas, A.: On the Correspondence between Approximations and Similarity, *Proc.of 4th Int. Conf. RSCTC'2004* (S. Tsumoto, R. Slowinski, J. Komorowski, J. Grzymala-Busse, Eds.), LNAI, vol. 3066 of *LNAI*, Springer-Verlag, 2004, 143–152.

[11] Fitting, M.: A Kripke-Kleene Semantics for Logic Programs, *Journal Logic Programming*, **2**(4), 1985, 295–312.

[12] Gabbay, D., Ohlbach, H.: Quantifier Elimination in Second-Order Predicate Logic, *South African Computer Journal*, **7**, 1992, 35–43.

[13] Konikowska, B.: A logic for reasoning about relative similarity, *Studia Logica*, **58**(1), 1997, 185–226.

[14] Magnusson, M.: DLS*, *http://www.ida.liu.se/labs/kplab/projects/dlsstar/,*, 2005.

[15] Nonnengart, A., Ohlbach, H., Szałas, A.: Elimination of Predicate Quantifiers, *Logic, Language and Reasoning. Essays in Honor of Dov Gabbay, Part I* (H. Ohlbach, U. Reyle, Eds.), Kluwer, 1999, 159–181.

[16] Nonnengart, A., Szałas, A.: A Fixpoint Approach to Second-Order Quantifier Elimination with Applications to Correspondence Theory, *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa* (E. Orłowska, Ed.), Studies in Fuzziness and Soft Computing, vol. 24 of *Studies in Fuzziness and Soft Computing*, Springer Physica-Verlag, 1998, 307–328.

[17] Pawlak, Z.: Information systems – theoretical foundations, *Information Systems*, **6**, 1981, 205–218.

[18] Pawlak, Z.: *Rough Sets. Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[19] Przymusinski, T., Przymusinska, H.: Semantic issues in deductive databases and logic programs, *Formal Techniques in AI* (R. Banerji, Ed.), North Holland, 1990, 321–367.

[20] Skowron, A., Stepaniuk, J.: Tolerance Approximation Spaces, *Fundamenta Informaticae*, **27**, 1996, 245–253.

[21] Słowiński, R., Vanderpooten, D.: Similarity relation as a basis for rough approximations, *Advances in Machine Intelligence & Soft Computing* (P. Wang, Ed.), Bookwrights, Raleigh NC, 1997, 17–33.

[22] Słowiński, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity, *IEEE Trans. on Data and Knowledge Engineering*, **12(2)**, 2000, 331–336.

[23] Szałas, A.: On the Correspondence between Modal and Classical Logic: An Automated Approach, *Journal of Logic and Computation*, **3**, 1993, 605–620.

[24] Van Benthem, J.: Correspondence Theory, *Handbook of Philosophical Logic* (D. Gabbay, F. Guenthner, Eds.), vol. 2, D. Reidel Pub. Co., 1984, 167–247.