Approximative Query Techniques for Agents with Heterogeneous Ontologies and Perceptive Capabilities

Patrick Doherty and **Andrzej Szałas**
Department of Computer and Information Science
Linköping University, Sweden
{patdo, andsz}@ida.liu.se

Witold Łukaszewicz
The College of Economics and Computer Science
Olsztyn, Poland
witlu@ida.liu.se

Abstract

In this paper, we propose a framework that provides software and robotic agents with the ability to ask *approximate* questions to each other in the context of heterogeneous ontologies and heterogeneous perceptive capabilities. The framework combines the use of logic-based techniques with ideas from approximate reasoning. Initial queries by an agent are transformed into approximate queries using weakest sufficient and strongest necessary conditions on the query and are interpreted as lower and upper approximations on the query. Once the base communication ability is provided, the framework is extended to situations where there is not only a mismatch between agent ontologies, but the agents have varying ability to perceive their environments. This will affect each agent's ability to ask and interpret results of queries. Limitations on perceptive capability are formalized using the idea of tolerance spaces.

Introduction

In this paper, we will propose a number of logic-based techniques combined with ideas from approximate reasoning that can provide software or robotic agents with the ability to ask *approximate* questions to each other in the context of heterogeneous ontologies and perceptive capabilities. Communication in the context of heterogeneous ontologies is a particularly pressing research issue and has applications in many areas such as the semantic web, distributed robotics and distributed databases.

For example, the next stage in the evolution of the WWW is to enhance the current infrastructure with support for explicit, machine accessible descriptions of information content on the Web. These machine accessible descriptions of information content should be usable and understandable by machines, in particular software agents. Tim Berners-Lee has used the term *Semantic Web* – a web of data that can be processed directly or indirectly by machines (Berners-Lee 2000), to describe this next phase in the evolution of the Web.

The meaning or semantics of diverse information content has to be accessible to software agents for use and reasoning if sophisticated knowledge intensive tasks are to be automated in the Web context. Most importantly, just as humans cooperate and communicate in a common language and conceptual space in order to achieve complex tasks, so will software agents, both with other software agents and with humans. There is a great deal of research activity in this area, particularly in providing the necessary tools to support communication among agents and construction and use of shared ontologies.

Many are of the opinion that it is wishful thinking to assume that there will be one common or global conceptual space on the WWW. Instead there will be many local ontological structures, some shared by agents and others not. A similar communication problem arises in the area of robotics, especially in distributed network-centric applications where there are many robots interacting with humans. It can also be expected that the players in these interactions share some ontological space, but not all. Finally, in the area of distributed databases, each database will have its own vocabulary, but one would like to ask questions in a seamless manner using ones own vocabulary.

The term "ontology" has a number of different interpretations which often depend upon the research community one belongs to. For the purposes of this paper, we will use the more traditional definition from artificial intelligence, more specifically knowledge representation, where the term is used somewhat more pragmatically to describe how we choose to "slice up" reality and represent these choices in representational structures used for reasoning about agent environments at various levels of abstraction. One common way of "slicing" or conceptualizing is to specify a base set of individuals, properties, relations and dependencies between them. This choice is particularly amenable to a straightforward use of logic as a representational tool and is the one we will focus on.

One of the problems we are interested in investigating is when two agents want to communicate with each other in their own unrestricted languages, yet their vocabularies (ontologies) only partially overlap. The question would then be what is the provably strongest question an agent can ask another given the mismatch in ontologies. The original query in the asking agent's language must be transformed into a query that the receiving agent understands. We will show that an approximate query can be generated. We assume that standard techniques from the literature such as those described in Bailin and Truszkowski (Bailin & Truszkowski

2002) can be used for alignment of the shared part of two agent's ontologies. The problem specification is depicted in Figure 1.

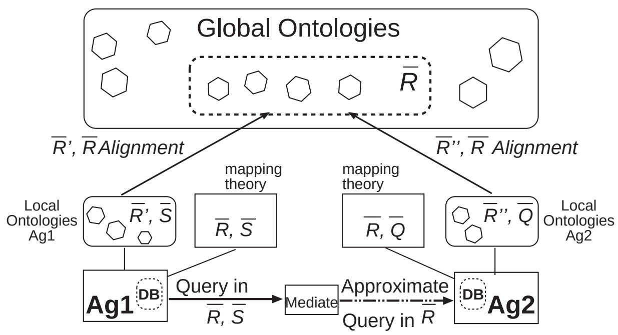

Figure 1: Problem Specification

Assume agent Ag_1 has a local ontology consisting of concepts/relations in $\bar{R}'$ and $\bar{S}$ and that concepts/relations in $\bar{R}'$ have previously been aligned with those in $\bar{R}$, a subset from a global ontology repository assumed accessible to the agent community in question. In addition, Ag_1's knowledge base contains a mapping theory which represents dependencies between various concepts/relations in $\bar{R}$ and $\bar{S}$. These are assumed to be logical formulas in a fragment of first-order logic representing some sufficient and necessary conditions for concepts/relations in $\bar{R}$ and $\bar{S}$. Ag_1's database can contain approximations of concepts/relations in $\bar{R}'$, $\bar{R}$ and $\bar{S}$.

Assume similarly, that agent Ag_2 has a local ontology consisting of concepts/relations in $\bar{R}''$ and $\bar{Q}$ and that concepts/relations in $\bar{R}''$ have previously been aligned with those in $\bar{R}$, the same subset that $Ag1$ has aligned R' with. In addition, Ag_2's knowledge base contains a mapping theory which represents dependencies between various concepts/relations in $\bar{R}$ and $\bar{Q}$ representing some sufficient and necessary conditions for concepts/relations in $\bar{R}$ and $\bar{Q}$. Ag_2's database can contain approximations of concepts/relations in $\bar{R}''$, $\bar{R}$ and $\bar{Q}$.

From an external perspective, agents Ag_1 and Ag_2 have concepts/relations in $\bar{R}$ in common and therefore a common language to communicate, but at the same time, Ag_1 has the additional concepts/relations $\bar{S}$ disjoint from Ag_2, and Ag_2 has the additional concepts/relations $\bar{Q}$ disjoint from Ag_1. When reasoning about the world and in asking questions to other agents, it is only natural that Ag_1 would like to use concepts from $\bar{R}'$, $\bar{R}$ and $\bar{S}$. In a similar manner, Ag_2 would like to use concepts from $\bar{R}''$, $\bar{R}$ and $\bar{Q}$. Since we assume alignment of both $\bar{R}'$ and $\bar{R}''$ with $\bar{R}$, and that both agents know they have $\bar{R}$ in common, the communication issue reduces to that between the two sub-languages using vocabularies $\bar{R}$, $\bar{S}$ and $\bar{R}$, $\bar{Q}$, respectively.

Suppose agent Ag_1 wants to ask agent Ag_2 a question in Ag_1's own language. We will assume that any first-order or fixpoint formula using concepts/relations from $\bar{R}$, $\bar{S}$ can be used to represent the question. To do this, Ag_1 will supply the query α to its mediation function in addition to its mapping theory $T(\bar{R}, \bar{S})$. The mediation function will return a new approximate query consisting of

- the weakest sufficient condition of α under theory $T(\bar{R}, \bar{S})$ in the sub-language consisting of concepts/relations from $\bar{R}$ and
- the strongest necessary condition of α under theory $T(\bar{R}, \bar{S})$ in the sub-language consisting of concepts/relations from $\bar{R}$.

Both these formulas can be understood by agent Ag_2 because they are formulated using concepts/relations that Ag_2 can understand and that can be used to query its relational database for a reply to Ag_1. More importantly, it can be formally shown that agent Ag_1 can not ask a question more informative, under the assumptions we have made.

The issue of heterogeneous perceptive capabilities of agents and their influence on agent communication has some similarity with heterogeneous ontologies, but demands an additional suite of representation and reasoning techniques. In addition to heterogeneous ontologies, the perceptive limitations of a robotic agent induced by its sensor suite or other contingencies should be taken into account not only when the robotics agent reasons about its external and internal environments, but also when one or more robotic agents communicate with each other by asking questions about each others knowledge about the world or themselves. In this case, two robotic agents communicating with each other can only ever ask queries of an approximative nature and receive answers of an approximative nature as seen through their respective filters of perceptive limitation. The issue at hand is to provide representations of perceptive limitations directly related to sensors and other data flows in robotic and agent systems and to integrate these with agent querying mechanisms.

In this paper, we also propose a technique that can provide software and robotic agents with the ability to ask *approximate* questions to each other in the context of heterogeneous perceptive capabilities. Even though they may have concepts in common, their ability to perceive individuals as having specific properties or relations can be distinct. The question then is how this affects the questions that can be asked and the replies that can be generated by agents with perception functions limited to varying degrees.

In order to provide the proper level of detail for the specific framework in question, the following set of abstractions will be used in the paper. Each robotic agent will have access to the following functionalities and representations:

- An abstraction called a tolerance space which is used to represent similarity of data points for both basic and complex data domains.
- One or more databases capable of holding relational data. These databases may contain representations of crisp relations or approximate relations. The approximate relations will be represented using intuitions from the discipline of rough set theory. The intention is that sensor data is used in the generation of some of these approximate relations stored in the databases. Tolerance spaces again play a role in the generation of approximate relations from spe-

cific attributes in vectors or arrays of attributes representing sensors.

- A query mechanism which permits each agent to ask questions about knowledge in its own databases or in the databases of other agents. These queries will be approximate in nature due to the approximate nature of the knowledge stored in the databases. They will also be contextualized by perceptive limitations represented as tolerance spaces on more complex data domains.

In the remainder of the paper, we will provide the details for communicative functionalities for software and robotics agents in the context of heterogeneous ontologies and also perceptive capabilities. We will consider heterogeneous ontologies first and perceptive capabilities afterwards. Examples will be provided for each of the techniques in addition to an example where both techniques are used in an integrated manner.

The paper is structured as follows. In the first section, representations of approximate relations and queries are introduced using intuitions from rough set theory. In the second section, the central concepts of weakest sufficient and strongest necessary conditions, and their use in representing approximate queries are considered. In the third section, a formal framework for modeling agent communication in the context of heterogeneous ontologies is proposed and clarified using examples from an unmanned aerial vehicle scenario. In the fourth section, the important concept of a tolerance space is introduced. These spaces are used to represent indiscernibility, uncertainty and similarity between data. A formal framework for modeling agent communication with heterogeneous perceptive capabilities is then introduced in the fifth section with examples. We then conclude with a discussion. Some of these ideas were originally presented separately in (Doherty, Łukaszewicz, & Szałas 2001) and (Doherty, Łukaszewicz, & Szałas 2003). This paper combines and extends the two.

Set Approximation

The methodology we propose in this paper uses a generalization of a number of ideas associated with rough set theory which was introduced by Pawlak (Pawlak 1991). In many AI applications one faces the problem of representing and processing incomplete, imprecise, and approximate data. Many of these applications require the use of approximate reasoning techniques. The assumption that objects can be observed only through the information available about them leads to the view that knowledge about objects in themselves, is insufficient for characterizing sets or relations precisely since the knowledge about objects is incomplete. We thus assume that any imprecise concept is replaced by a pair of precise concepts called the lower and the upper approximation of the imprecise concept, as defined below. A graphical depiction of concept regions is provided in Figure 2.

Definition 1 An *approximation structure* is a tuple

$$S = \langle U, {}_{S^+}, {}_{S^\oplus} \rangle ,$$

where ${}_{S^+}, {}_{S^\oplus} : 2^U \longrightarrow 2^U$ are operations on sets such that, for any set $Z \subseteq U$, $Z_{S^+} \subseteq Z \subseteq Z_{S^\oplus}$.

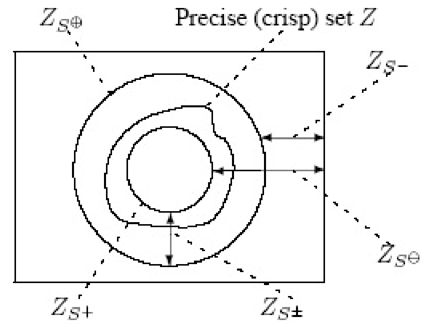

Figure 2: Approximations of a set

An *approximate set* wrt S is a pair $\langle Z_{S^+}, Z_{S^\oplus}\rangle$, where $Z \subseteq U$. In such a case:

- Z_{S^+} is called the *lower approximation* of Z (wrt S)
- $Z_{S^\oplus}$ is called the *upper approximation* of Z (wrt S)
- $Z_{S^\pm} \stackrel{\text{def}}{=} Z_{S^\oplus} - Z_{S^+}$, is the *boundary region* of Z (wrt S)
- $Z_{S^-} \stackrel{\text{def}}{=} -Z_{S^\oplus}$, is the *negative region* of Z (wrt S). ■

Intuitively,

- Z_{S^+} consists of objects that with certainty belong to Z
- $Z_{S^\oplus}$ consists of objects that might belong to Z
- $Z_{S^\pm}$ consists of objects for which it is unknown whether they belong to Z
- Z_{S^-} consists of all objects which with certainty do not belong to Z.

Approximate databases are generalizations of relational databases where relations are represented as having both lower and upper approximations. One can then associate additional integrity constraints on the lower and upper approximations. For the purposes of this paper, we associate a single constraint where the lower approximation of a relation is a subset of the upper approximation. For other alternatives, see (Doherty & Szałas 2004).

Definition 2 A (*relational, crisp*) *database* is 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.

An *approximate database* is a tuple

$$D = \left\langle U, \left\{ r^j = \left\langle r_1^j, r_2^j \right\rangle \mid j \in J \right\} \right\rangle ,$$

where r_1^j, r_2^j are of the same arity and $r_1^j \subseteq r_2^j$.[1]

The *type* of a (crisp or approximate) database is a sequence $\langle a_j \mid j \in J \rangle$, where for any $j \in J$, a_j is the number of arguments (*arity*) of r^j. ■

[1]Observe that crisp relational databases are approximate relational databases with $r_1^j = r_2^j$ for all $j \in J$.

We will also require a definition of approximate queries. In essence, given a crisp query, an approximate query provides an upper and lower approximation on the query.

Definition 3 An *approximate query* is a pair

$$Q = \langle Q'(\bar{x}), Q''(\bar{x})\rangle\,,$$

where Q' and Q'' are formulas of a given logic, in which $\bar{x}$ consists of all free variables (common to Q' and Q''), such that for any underlying database[2] D,

$$D \models Q'(\bar{x}) \rightarrow Q''(\bar{x}).$$

Formulas Q', Q'' are called the *lower* (respectively, *upper*) *approximation part* of Q.

By $Q'(\bar{x})_D$ (respectively, $\langle Q'(\bar{x}), Q''(\bar{x})\rangle_D$) we denote the result of evaluating the query $Q'(\bar{x})$ (respectively, the approximate query $\langle Q'(\bar{x}), Q''(\bar{x})\rangle$) on the database D. ■

Strongest Necessary and Weakest Sufficient Conditions

Strongest necessary and weakest sufficient conditions, as used in this paper and defined below, were first introduced for the propositional case in (Lin 2000) and generalized for the first-order case in (Doherty, Łukaszewicz, & Szałas 2001).

Definition 4 By *a necessary condition of a formula α on the set of relation symbols P under theory T* we shall understand any formula ϕ containing only symbols in P such that $T \models \alpha \rightarrow \phi$. It is the *strongest necessary condition*, denoted by $\textsc{Snc}(\alpha; T; P)$ if, additionally, for any necessary condition ψ of α on P under T, $T \models \phi \rightarrow \psi$ holds. ■

Definition 5 By *a sufficient condition of a formula α on the set of relation symbols P under theory T* we shall understand any formula ϕ containing only symbols in P such that $T \models \phi \rightarrow \alpha$. It is the *weakest sufficient condition*, denoted by $\textsc{Wsc}(\alpha; T; P)$ if, additionally, for any sufficient condition ψ of α on P under T, $T \models \psi \rightarrow \phi$ holds. ■

The set P in the definitions for wsc's and snc's is referred to as the *target language*.

The following lemma has been proven in (Doherty, Łukaszewicz, & Szałas 2001).

Lemma 6 For any formula α, any set of relation symbols P and theory T such that the set of free variables of T is disjoint with the set of free variables of α:

- the strongest necessary condition $\textsc{Snc}(\alpha; T; P)$ is defined by $\exists\bar{\Phi}.[T \wedge \alpha]$,
- the weakest sufficient condition $\textsc{Wsc}(\alpha; T; P)$ is defined by $\forall\bar{\Phi}.[T \rightarrow \alpha]$,

where $\bar{\Phi}$ consists of all relation symbols appearing in T and α but not in P. ■

[2] We deal with relational databases, where queries are formulated as first-order or fixpoint formulas (for textbooks on this approach see, e.g., (Abiteboul, Hull, & Vianu 1996; Ebbinghaus & Flum 1995; Immerman 1998)).

The above characterizations are second-order. However, for a large class of formulas, one can obtain logically equivalent first-order formulas (Doherty, Łukaszewicz, & Szałas 1997; Gabbay & Ohlbach 1992) or fixpoint formulas (Nonnengart & Szałas 1998) by applying techniques for eliminating second-order quantifiers. Many of these techniques have been implemented. For an overview of existing second-order quantifier elimination techniques see, e.g., (Nonnengart, Ohlbach, & Szałas 1999).

In Theorem 7 below we quote a result from (Nonnengart & Szałas 1998) which allows one to eliminate second-order quantifiers for formulas when they can be transformed into a certain form.

Let e, t be any expressions and s any subexpression of e. By $e(s := t)$ we mean the expression obtained from e by substituting each occurrence of s by t. Let $\alpha(\bar{x})$ be a formula with free variables $\bar{x}$. Then by $\alpha(\bar{x})[\bar{a}]$ we mean the application of $\alpha(\bar{x})$ to arguments $\bar{a}$. In what follows $\textsc{Lfp}\,\Phi.\alpha(\Phi)$ and $\textsc{Gfp}\,\Phi.\alpha(\Phi)$ denote the least and greatest fixpoint operators, respectively.

An occurrence of a relation symbol Φ is *positive* (respectively *negative* in a formula α, if it is in α in the scope of an even (respectively odd) number of negations.[3] Formula α is *positive* (respectively *negative*) wrt relation symbol Φ if all occurrences of Φ in α are only positive (respectively only negative).

Theorem 7 Assume that all occurrences of the predicate variable Φ in the formula β bind only variables and that formula α is positive w.r.t. Φ.

- if β is negative w.r.t. Φ then
$$\begin{aligned}\exists\Phi\forall\bar{y}\ [\alpha(\Phi) \rightarrow \Phi(\bar{y})] \wedge [\beta(\neg\Phi)]\ \equiv&\\ \beta[\Phi(\bar{t}) := \textsc{Lfp}\,\Phi(\bar{y}).\alpha(\Phi)[\bar{t}]]&\end{aligned} \quad (1)$$
- if β is positive w.r.t. Φ then
$$\begin{aligned}\exists\Phi\forall\bar{y}[\Phi(\bar{y}) \rightarrow \alpha(\Phi)] \wedge [\beta(\Phi)]\ \equiv&\\ \beta[\Phi(\bar{t}) := \textsc{Gfp}\,\Phi(\bar{y}).\alpha(\Phi)[\bar{t}]].&\end{aligned} \quad (2)$$

■

The formula that results when Theorem 7 is applied to an input formula, is a fixpoint formula. If the input formula is non-recursive wrt the relations that are to be eliminated, then the resulting formula is a first-order formula[4]. The input formula can also be a conjunction of the form (1) or a conjunction of formulas of the form (2) since those conjunctions can be transformed equivalently into the form required in Theorem 7.

Agent Communication with Heterogeneous Ontologies

The original proposal for developing a communicative functionality for agents in the context of heterogeneous ontologies was initiated in (Doherty, Łukaszewicz, & Szałas 2001).

[3] As usual, it is assumed here that all implications of the form $p \rightarrow q$ are substituted by $\neg p \vee q$ and all equivalences of the form $p \equiv q$ are substituted by $(\neg p \vee q) \wedge (\neg q \vee p)$.

[4] In such cases, fixpoint operators appearing on the righthand sides of formulas (1) and (2) can simply be removed.

In this case, only strongest necessary conditions replaced the original query and no appeal was made to approximate queries. Let us now further develop the idea.

In this case, we assume an agent Ag_1 wants to ask a question Q to an agent Ag_2. Agent Ag_1 can use any of the terms in $\bar{R}, \bar{S}$, where the terms in $\bar{S}$ are unknown to agent Ag_2, while both have the terms in $\bar{R}$ in common. Let $T(\bar{R}, \bar{S})$ be a mapping theory in agent Ag_1's knowledge base describing some relationships between $\bar{R}$ and $\bar{S}$. It is then natural for agent Ag_1 to use its mediation function to first compute the weakest sufficient condition $\text{Wsc}(Q; T(\bar{R}, \bar{S}); \bar{R})$ and the strongest necessary condition $\text{Snc}(Q; T(\bar{R}, \bar{S}); \bar{R})$, with the target language restricted to the common agent vocabulary $\bar{R}$ and then to replace the original query by the computed conditions.

The new query is generally not as precise as the original one, but is the best that can be asked. Namely,

- the weakest sufficient condition provides one with tuples satisfying the query with certainty
- the strongest necessary condition provides one with tuples that might satisfy the query
- the complement of the strongest necessary condition provides one with tuples that with certainty do not satisfy the query.

Observe that the difference between the strongest necessary and the weakest sufficient conditions contains tuples for which it is unknown whether they do or do not satisfy the query.

In summary, instead of asking the original query Q which can be an arbitrary first-order or fixpoint formula, agent Ag_1 will ask a pair of queries

$$\left\langle \text{Wsc}(Q; T(\bar{R}, \bar{S}); \bar{R}), \text{Snc}(Q; T(\bar{R}, \bar{S}); \bar{R}) \right\rangle$$

which represent the lower and upper approximation of Q.

An interesting application of these techniques and those related to perceptive limitations of agents which will be considered later in the paper involves communication between ground operators and Unmanned Aerial Vehicles (UAVs). One can assume that vocabularies and associated ontologies used by human operators and autonomous systems such as UAVs will often be different. Consequently, communication between these agents will have to be mediated in various ways. Besides ontological mismatches, it will often be the case that communicating agents will have different perceptive capabilities. This in turn will add to the complexity of the communicative component in multi-agent systems with human users. The following example, focuses on some of these problems.

Example 8 Consider a situation where a ground operator (agent Ag_G) is communicating with a UAV (agent Ag_V), while it is flying over a road segment. Assume Ag_V can provide information about the following approximate relations, $\bar{R}$, and that Ag_V has these relations in common with Ag_G:

- $V(x, y)$ – there is a visible connection between objects x and y
- $S(x, y)$ – the distance between objects x and y is small
- $E(x, y)$ – objects x and y have equal speed
- $C(x, z)$ – object x has color z, (we consider colors b, dr, r standing for "brown", "dark red" and "red", respectively).

Table 1: Perceived situation on the road segment considered in Example 8.

Object	V	S	E	C
1	2	2, 5	2, 5	b
2	1	1, 3, 4	1, 3, 4	b
3	-	2	2	b
4	-	2	2	r
5	-	1	1	dr

We can assume that the concepts "visible connection", "small distance" and "color" were previously acquired via machine learning techniques with sample data generated from video logs provided by a UAV on previous flights while flying over similar road systems.

Assume also that agent Ag_G has a vocabulary consisting of not only $\bar{R}$, but other relations $\bar{S}$, not known by Ag_V. In particular, $\bar{S}$ includes a relation $Con(x, y)$, which denotes that objects x and y are connected. Suppose that Ag_G knows the following facts about Con which are included in Ag_G's knowledge base:

$$\forall x, y.[V(x, y) \to Con(x, y)] \tag{3}$$

$$\forall x, y.[Con(x, y) \to (S(x, y) \wedge E(x, y))] \tag{4}$$

and that (3) and (4) are consistent (checking the consistency of such formulas with the contents of Ag_G's database can be checked efficiently (Doherty, Łukaszewicz, & Szałas 1999)).

Suppose Ag_G wants to ask Ag_V for information about all connected brown objects currently perceived by Ag_V. This can be represented as the following query,

$$Con(x, y) \wedge C(x, b) \wedge C(y, b). \tag{5}$$

Since Ag_V can not understand queries with the term Con, Ag_G has to reformulate query (5) using only terms in $\bar{R}$ which are also understood by Ag_V. The most informative query it can then ask, assuming the assumptions stated in the introduction, is:

$$\begin{aligned} &\langle \text{Wsc}((5); (3) \wedge (4); \{V, S, E, C\}), \\ &\qquad \text{Snc}((5); (3) \wedge (4); \{V, S, E, C\}) \rangle . \end{aligned} \tag{6}$$

By applying Lemma 6 and Theorem 7 one obtains the following logically equivalent formulation of (6):[5]

$$\langle V(x, y) \wedge C(x, b) \wedge C(y, b), \tag{7}$$

$$S(x, y) \wedge E(x, y) \wedge C(x, b) \wedge C(y, b) \rangle . \tag{8}$$

Observe that objects perceived by Ag_V satisfying (7) belong to the lower approximation of the set of objects satisfying the original query (5) and objects perceived by Ag_V satisfying (8) belong to the upper approximation of the set of objects satisfying the original query (5). Thus:

- all objects satisfying formula (7) also satisfy the original query (5)

[5] These steps can be computed automatically.

- all objects not satisfying formula (8) do not satisfy the original query (5)
- on the basis of the available information and the capabilities of Ag_V, it remains unknown to Ag_G whether objects satisfying formula $((8) \wedge \neg(7))$ do or do not satisfy the original query (5).

Suppose Table 1 represents the perceived situation on the road segment as sensed by Ag_V.

Table 1 represents these relations by indicating, for each perceived object, with which entities a given relation holds. For example,

- the first row means that there is a visible connection between objects 1 and 2, the distance between object 1 and objects $2, 5$ is small, object 1 has equal speed with objects $2, 5$ and that the color of object 1 is b
- the third row means that that there no visible connection between object 3 and any other objects, the distance between object 3 and object 2 is small, object 3 has equal speed with object 2 and that the color of object 3 is b.

Query (6), approximating the original query (5), computed over the database shown in Table 2, results in the following

$$\langle \{\langle 1,2\rangle, \langle 2,1\rangle\}, \{\langle 1,2\rangle, \langle 2,1\rangle, \langle 2,3\rangle, \langle 3,2\rangle\}\rangle,$$

which will be returned as an answer to Ag_G's original query. As a result, Ag_G will know that tuples $\langle 1,2\rangle, \langle 2,1\rangle$ satisfy the query (5), tuples $\langle 2,3\rangle, \langle 3,2\rangle$ might satisfy the query and, for example, the tuple $\langle 1,5\rangle$ does not satisfy the query (in fact, object 5 is not brown). ∎

Tolerance Spaces

Tolerance spaces, as defined below, have been introduced in (Doherty, Łukaszewicz, & Szałas 2003). 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 of the indiscernibility partitions used in rough set theory where instead of partitions, the neighborhoods provide a covering of the universe. Conceptually, these spaces are quite versatile. Later in the paper, they will be used to represent limitations on an agent's perceptive capabilities. They can also be used to model the uncertainty of data associated with sensors.

Definition 9 A *tolerance function* on a set U is a function $\tau : U \times U \longrightarrow [0,1]$ such that for all $x, y \in U$,

$$\tau(x,x) = 1 \quad \text{and} \quad \tau(x,y) = \tau(y,x).$$ ∎

Definition 10 For $p \in [0,1]$, a *tolerance relation to a degree at least p, based on τ*, is a relation τ^p given by

$$\tau^p \stackrel{\text{def}}{=} \{\langle x,y\rangle \mid \tau(x,y) \geq p\}.$$

The relation τ^p is also called the *parameterized tolerance relation*. ∎

In what follows, $\tau^p(x,y)$ is used to denote the characteristic function for the relation τ^p.

A parameterized tolerance relation is used to construct tolerance neighborhoods for individuals.

Definition 11 A *neighborhood function wrt τ^p* is a function given by

$$n^{\tau^p}(u) \stackrel{\text{def}}{=} \{u' \in U \mid \tau^p(u,u') \text{ holds}\}.$$

A *neighborhood* of u wrt τ^p is the value $n^{\tau^p}(u)$. ∎

The concept of tolerance spaces plays a fundamental role in our approach.

Definition 12 A *tolerance space* is a tuple $TS = \langle U, \tau, p\rangle$, consisting of

- a nonempty set U, called the *domain* of TS
- a *tolerance function* τ
- a *tolerance parameter* $p \in [0,1]$. ∎

Approximations by tolerance spaces are as defined below.

Definition 13 Let $TS = \langle U, \tau, p\rangle$ be a tolerance space and let $S \subseteq U$. The *lower and upper approximation of S wrt TS*, denoted respectively by S_{TS^+} and $S_{TS^\oplus}$, are defined by

$$S_{TS^+} = \{u \in U : n^{\tau_p}(u) \subseteq S\}$$
$$S_{TS^\oplus} = \{u \in U : n^{\tau_p}(u) \cap S \neq \emptyset\}.$$ ∎

In the definition, U might represent a primitive data set such as that used for a particular sensor, or a complex data set such as a set of tuples. For example, consider a relational database with one relation S of k-arity and with universe U consisting of all k-tuples. In this case, the relation may represent raw data about S. Suppose there is also a tolerance space $TS = \langle U, \tau, p\rangle$. TS creates neighborhoods around tuples. An agent, when asking whether a tuple $\bar{x}$ is a member of the relation is really asking whether the neighborhood around the the tuple is a member of the relation. If so, the answer is yes, if there is an intersection, the answer is maybe, if the intersection is empty, the answer is no.

In fact, this particular use of tolerance spaces can be generalized to relational databases with an arbitrary number of relations where the data in the database is assumed to be raw data about the relations. Using tolerance spaces, the relational database can be turned into an approximate database where each relation is viewed as having an upper and lower approximation. Rather than machine learning the approximate relations directly, one can assume the tolerance spaces as given and apply them to raw data to generate an approximate database. The following definition and example show how this is done.

Definition 14 Let $D = \left\langle U, \{r^j \mid j \in J\}\right\rangle$ be a relational database. Then we say that a sequence of tolerance spaces $\overline{TS} = \langle TS_j \mid j \in J\rangle$ is *compatible with D* provided that for any $j \in J$, $TS_j = \langle U_j, \tau_j, p_j\rangle$, where U_j is the set of all tuples of arity the same as the arity of r^j. ∎

Definition 15 Let $D = \left\langle U, \{r^j \mid j \in J\}\right\rangle$ be a relational database and $\overline{TS}$ be a sequence of tolerance spaces compatible with D. If D is crisp, then by an *approximation of D wrt $\overline{TS}$*, we mean the structure

$$D^{\overline{TS}} = \left\langle U, \left\{\left\langle {r^j}_{TS_j^+}, {r^j}_{TS_j^\oplus}\right\rangle \mid j \in J\right\}\right\rangle.$$

If D is approximate, where for $j \in J$, $r^j = \left\langle r_1^j, r_2^j \right\rangle$, then the *approximation of* D *wrt* $\overline{TS}$ is defined as

$$D^{\overline{TS}} = \left\langle U, \left\{ \left\langle (r_1^j)_{TS_j^+}, (r_2^j)_{TS_j^\oplus} \right\rangle \mid j \in J \right\} \right\rangle. \blacksquare$$

Note that in the latter case, one can still apply additional tolerance spaces to upper and lower approximations of a relation since these are also represented as relations in the database.

Example 16 Consider again, a situation where a ground operator (agent Ag_G) is communicating with a UAV(agent Ag_V), while it is flying over a road segment. Assume Ag_V can provide information about the relations $\bar{R}$, considered in Example 8, and that Ag_V has these in common with Ag_G.

Let the actual situation on a road segment be given by the (crisp) relational database shown in Table 2. Note that it is not necessarily the case that the UAV agent Ag_V, has direct access to this information since it represents the actual situation and the agent may have limited perceptive capabilities relative to certain features such as color. We will see how tolerance spaces can be used to model these limitations.

Table 2: Database considered in Example 16 reflecting a situation on a road segment.

Object	V	S	E	C
1	2	2, 5	2, 5	r
2	1	1, 3, 4	1, 3, 4	b
3	-	2	2	dr
4	-	2	2	r
5	-	1	1	dr

Table 3: Approximation (lower approximations) of the relational database given in Table 2 wrt the perception capabilities of agent Ag_V.

Object	V_+	S_+	E_+	C_+
1	2	2, 5	2, 5	r
2	1	1, 3, 4	1, 3, 4	-
3	-	2	2	-
4	-	2	2	r
5	-	1	1	-

Table 4: Approximation (upper approximations) of the relational database given in Table 2 wrt the perception capabilities of agent Ag_V.

Object	$V_\oplus$	$S_\oplus$	$E_\oplus$	$C_\oplus$
1	2	2, 5	2, 5	r
2	1	1, 3, 4	1, 3, 4	b, dr
3	-	2	2	b, dr
4	-	2	2	r
5	-	1	1	b, dr

Consider the approximation of the relational database given in Table 2 wrt the tolerance space $T_V = \langle U, \tau_V, p_V \rangle$, where $\tau_V^{p_V}$ identifies equal elements and additionally dr with b. This tolerance space represents a simple perceptive limitation associated with the UAV agent, Ag_V. It can not distinguish between the colors black and dark red. Using Definition 15, the resulting approximation of the actual situation relative to the agent's tolerance space is presented in Tables 3 and 4. Now, e.g.,

- the first row in Table 3 indicates that there definitely is a visible connection between objects 1 and 2, the distance between object 1 and objects $2, 5$ is definitely small, object 1 definitely has equal speed with objects $2, 5$ and that the color of object 1 is definitely r
- the third row in Table 4 indicates that there cannot be any visible connection between object 3 and any other object, the distance between object 3 and object 2 might be small, object 3 might have equal speed with object 2 and that the color of object 3 might be b or dr.

Note that several tolerance spaces could be associated with each type of data in a table if desired and more complex perceptive limitations could also be modeled. $\blacksquare$

Previously, we considered how an approximate query could be generated automatically using snc's and wsc's. We now consider another means of generating an approximate query from a crisp query initially represented as a logical formula.

Definition 17 For any formula α referring to relations in D,

- by α_+ we understand it to be the formula α in which any positive occurrence of a relation symbol, say r^j, is replaced by r_+^j and any negative occurrence of r^j is replaced by $r_\oplus^j$
- by $\alpha_\oplus$ we understand it to be the formula α in which any positive occurrence of a relation symbol, say r^j, is replaced by $r_\oplus^j$ and any negative occurrence of r^j is replaced by r_+^j. $\blacksquare$

Example 18 Consider the formula $r^1(\bar{x}) \wedge \neg r^2(\bar{y})$. Then:

- $[r^1(\bar{x}) \wedge \neg r^2(\bar{y})]_+ \equiv [r_+^1(\bar{x}) \wedge \neg r_\oplus^2(\bar{y})]$
- $[r^1(\bar{x}) \wedge \neg r^2(\bar{y})]_\oplus \equiv [r_\oplus^1(\bar{x}) \wedge \neg r_+^2(\bar{y})]$.

The two formulas on the right hand side would represent the lower and upper approximation of the original formula, respectively. $\blacksquare$

It is often the case that one would like to associate perceptive limitations with specific queries in a contextual manner. We allow for this possibility by associating a tolerance space with an approximate query. We call such queries, *tolerance queries*.

Definition 19 Let D be a (crisp or approximate) database. By a *tolerance query* we mean a tuple $\langle Q, T_Q \rangle$, where

- $Q = \langle Q'(\bar{x}), Q''(\bar{x}) \rangle$ is an approximate query
- T_Q is a tolerance space for tuples of arity the same as the arity of $\bar{x}$.

If $\overline{TS}$ is a sequence of tolerance spaces compatible with D, then the *meaning of tolerance query* Q *in database* D *wrt context* $\overline{TS}$ is given by

$$\left\langle [Q'(\bar{x})_{D^{\overline{TS}}}]_{T_Q^+}, [Q''(\bar{x})_{D^{\overline{TS}}}]_{T_Q^{\oplus}} \right\rangle . \qquad \blacksquare$$

Basically, what this definition states is that when an agent queries a database with a tolerance query, the result will be filtered through a specific limitation in perceptive capability associated with the query. We will see that additional perceptive limitations of a less contextual nature may also be associated with querying agents.

Agent Communication with Heterogeneous Perceptive Capabilities

Consider a multi-agent application in a complex environment such as the Web where software agents reside, or a natural disaster in an urban area where physical robots reside. Each agent will generally have its own view of its environment due to a number of factors such as the use of different sensor suites, knowledge structures, reasoning processes, etc. Agents may also have different understandings of the underlying concepts which are used in their respective representational structures and will measure objects and phenomena with different accuracy. How then can agents with different knowledge structures and perceptive accuracies understand each other and effect meaningful communication and how can this be modeled? In this section, we propose a framework to do this using tolerance spaces as the main representational tool to model many of these types of limitations and mismatches.

The approach may be summarized as follows. It is assumed that each agent has its own database. The database may be crisp or approximate and generated in any number of ways, some of which have been demonstrated already. The idea is that some perceptual and other limitations have already been encoded in the respective databases of the agents. For any tolerance agent, we will also assume an additional context consisting of a sequence of tolerance spaces. These may cover all, some or none of the relations in the database and are intended to represent additional limitations which are contextual in nature. The agent need not be aware of these limitations, but will always view its knowledge through this filter when asking questions internally and this context may be used when generating a tolerance query to be asked of another agent.

When an agent asks a question of another agent using a tolerance query, the question is interpreted by the other agent through its context and its database. Two sets of tuples are returned, representing the lower and upper approximation of the original query. The agent who asked the question, will then apply the tolerance space associated with its tolerance query to the result returned by the questioned agent. The net result will be an answer which takes into account both the perceptual limitations of the questioned agent and the current limitation associated with the tolerance query. Initial work with these ideas may be found in (Doherty, Łukaszewicz, & Szałas 2003).

We begin with a general definition of a *tolerance agent* specializing that provided in (Doherty, Łukaszewicz, & Szałas 2003).

Definition 20 By a *tolerance agent* we shall understand any pair $\langle Ag, D, \overline{TS}\rangle$, where Ag is an agent, D is its (crisp or approximate) database and $\overline{TS}$, called the *context of agent* Ag, is a sequence of tolerance spaces compatible with D. $\blacksquare$

Here we do not define what an agent is specifically, as the framework we propose is independent of the particular details. The assumption is that the Ag part of a tolerance agent consists of common functionalities normally associated with agents such as planners, reactive and other methods, knowledge bases or structures, etc. The knowledge bases or structures are also assumed to have a relational component consisting of a relational database (D). When the agent introspects and queries its own knowledge base its limited perceptive capabilities are reflected in any answer to a query due to its context.

Suppose that two tolerance agents have different perceptive capabilities and consequently different tolerance spaces. It will then be necessary to define the meaning of queries and answers relative to the two tolerance agents. As previously advocated, a tolerance agent, when asked about a relation, answers by using the approximations of the relation wrt its tolerance space. On the other hand, the agent that asked the query has to understand the answer provided by the other agent wrt to its own tolerance space.

The dialog between two agents:

- *query agent* $TA_1 = \langle Ag_1, D_1, \overline{TS}_1\rangle$
- *answer agent* $TA_2 = \langle Ag_2, D_2, \overline{TS}_2\rangle$,

will then conform to the following schema:

1. TA_1 asks TA_2 a question using a tolerance query $Q = \left\langle \langle Q', Q''\rangle, T_Q\right\rangle$; in fact, it sends to TA_2 the approximate query $\langle Q', Q''\rangle$ without T_Q,
2. TA_2 computes the answer approximating its database according to its current context $\overline{TS}_2$ and returns as an answer the approximate relation $\left\langle Q'_+, Q''_\oplus\right\rangle_{D_2^{\overline{TS}_2}}$. In order to simplify notation, we denote this relation by $R = \langle R', R''\rangle$
3. TA_1 receives R as input and interprets it according to the context T_Q indicated in the query. The resulting interpretation, $\left\langle R'_{T_Q^+}, R''_{T_Q^{\oplus}}\right\rangle$, provides the answer to the query, as understood by TA_1 and takes into account the perceptive limitations of both agents.

This schema will only work properly under the assumption that there is a common sub-vocabulary between the agents. This of course can be achieved by using techniques such as those associated with agents with heterogeneous ontologies.

Remark 21 Observe that the context associated with agent Ag_1 is not present in the schema directly. Generally, this context is used when the agent *introspects* and asks questions about its own database.

For any tolerance query, although one could choose an associated tolerance space, T_Q, arbitrarily, the intention is that T_Q will usually be a function of the agent's static context $\overline{TS}_1$ and any other contingent aspects one might want to take into account. For example, some of the sensors associated with a robotics platform might be broken. As another example, if Q is of the form $r^j(\bar{x})$ then, in most cases, T_Q will be the j-th tolerance space in $\overline{TS}_1$. Another intriguing use of T_Q in a tolerance query would be as a hypothetical or *what-if* filter. An agent could ask itself, "What if I only had these sensors to work with in this mission?" Or, "If I had agent TA_2's capabilities and my database, what sort of answer would agent TA_2 get from me and how would the agent use those answers?" ■

The definitions describing this interaction now follow.

Definition 22 Let TA_1 and TA_2 be as above. Let

$$Q = \langle \langle Q', Q'' \rangle, T_Q \rangle$$

be a tolerance query, which is asked by TA_1 and answered by TA_2. Then the *result of query* Q is defined as the meaning[6] of Q in database D_2 wrt context $\overline{TS}_2$. ■

Example 23 Consider the tolerance agents $\langle Ag_V, D, \langle T_V \rangle \rangle$ and $\langle Ag_G, D_G, \langle T_G \rangle \rangle$ where:

- Ag_V, D and T_V are as described in Examples 16 and 8 (i.e., Ag_V does not recognize the difference between colors dr and b, and the language of D consists of $\{V, S, E, C\}$ only)
- $T_G = \langle U, \tau_G, p_G \rangle$ such that $\tau_G^{p_G}$ identifies equal elements and additionally dr with r.[7]

Suppose Ag_G wants to ask Ag_V for information about colors of objects to which other objects are connected. This can be expressed by a crisp query

$$\exists x, y.[Con(x, y) \wedge C(x, z)]. \tag{9}$$

Recall that Con is not in the vocabulary of Ag_V, thus the query has to be approximated first and we can use the wsc/snc methodology introduced earlier. Accordingly, the approximation of (9) would be

$$\langle \text{WSC}((9); (3)\wedge(4); \{V, S, E, C\}), \\ \text{SNC}((9); (3)\wedge(4); \{V, S, E, C\}) \rangle . \tag{10}$$

After applying Lemma 6 and Theorem 7 to (10), one obtains the following pair of first-order queries which is logically equivalent to (10):

$$\Big\langle \exists x, y.[V(x, y) \wedge C(x, z)], \\ \exists x, y.[S(x, y) \wedge E(x, y) \wedge C(x, z)] \Big\rangle .$$

Choosing the tolerance space T_G associated with agent Ag_G, the resulting tolerance query to be asked would be:

$$\Big\langle \Big\langle \exists x, y.[V(x, y) \wedge C(x, z)], \\ \exists x, y.[S(x, y) \wedge E(x, y) \wedge C(x, z)] \Big\rangle, T_G \Big\rangle .$$

[6]As provided by Definition 19.

[7]For simplicity we provide one tolerance space and assume that two tuples are identified if the arguments representing color have values within the same neighborhood and arguments not representing colors have equal values.

Table 5: Approximation (lower approximations) of the relational database given in Table 2 wrt perception capabilities of agent Ag_G as defined in Example 23.

Object	V_+	S_+	E_+	C_+
1	2	2, 5	2, 5	-
2	1	1, 3, 4	1, 3, 4	b
3	-	2	2	-
4	-	2	2	-
5	-	1	1	-

Table 6: Approximation (upper approximations) of the relational database given in Table 2 wrt perception capabilities of agent Ag_G as defined in Example 23.

Object	$V_\oplus$	$S_\oplus$	$E_\oplus$	$C_\oplus$
1	2	2, 5	2, 5	r, dr
2	1	1, 3, 4	1, 3, 4	b
3	-	2	2	r, dr
4	-	2	2	r, dr
5	-	1	1	r, dr

Using Definition 22, agent Ag_V would then evaluate this tolerance query in the context of its perceptive capabilities, i.e., relative to the database approximation given in Tables 3 and 4. Ag_V would return the following answer to the query:

$$\Big\langle \exists x, y.[V_+(x, y) \wedge C_+(x, z)], \\ \exists x, y.[S_\oplus(x, y) \wedge E_\oplus(x, y) \wedge C_\oplus(x, z)] \Big\rangle .$$

Thus Ag_V will return the following answer to Ag_G:

$$\Big\langle \{r\}, \{r, b, dr\} \Big\rangle$$

Ag_G would then compute the final answer by interpreting the result returned by Ag_V relative to the tolerance space T_G associated with the original tolerance query using Definition 22 and the database approximation shown in Tables 5 and 6. The final answer is $\Big\langle \emptyset, \{r, b, dr\} \Big\rangle$. This takes into account both the perceptive limitations of the querying and answering agents.

The final answer states that Ag_G will not with certainty know about the specific colors of any object that has a connection to other objects, but it will know that such objects might have the colors red, black or dark-red. ■

Complexity of the Techniques

In order to build communication interfaces between agents, we have applied two approximation techniques:

1. approximations via weakest sufficient and strongest necessary conditions
2. approximations via neighborhoods induced by tolerance spaces.

First, we have used wsc's and snc's as approximate queries. Using wsc's as a database query is, in general co-NP-complete and using snc's as a query is NP-complete (as suggested by characterizations given in Lemma 6). This might make the approach questionable from the complexity point of view. However, as also discussed just after Lemma 6, for a large class of formulas, the second-order quantifiers can be eliminated for formulas of a certain syntactic form, e.g., that provided in Theorem 7. Observe that the form required in Theorem 7 is quite general, and strictly includes, e.g., Horn clauses used in logic programming.

The complexity of computing snc's and wsc's using an algorithm based on Theorem 7, e.g., a generalization of that described in (Doherty, Łukaszewicz, & Szałas 1997),[8] allows one to eliminate quantifiers in time polynomial in the size of the input formula. The resulting query, as well as so-called coherence conditions that might have to be checked together with the query (see (Doherty, Łukaszewicz, & Szałas 1999)), can be computed in time polynomial wrt the size of the database, since these are standard first-order or fixpoint queries.

The approximations and queries based on tolerance spaces are also computable in time polynomial in the size of the underlying database, assuming that tolerance functions of all necessary tolerance spaces can be computed tractably, which is the case in practical applications.

In summary, the proposed techniques provide us with tractable methods for building communication interfaces among agents with heterogenous ontologies and perceptive capabilities. The restrictions as to the class of formulas used in wsc's and snc's, as well as to the tractability of tolerance functions, are often satisfied in practical applications.

Conclusions

We have presented a formal framework for modeling a particular type of communication between software or robotic agents which takes into account heterogeneous ontologies and heterogeneous perceptive capabilities during the communicative act. The data- or knowledge bases associated with agents are formalized as approximate databases and the questions that may be asked are represented as 1st-order or fixpoint queries. Weakest sufficient and strongest necessary conditions are used to model approximate queries with specific sub-languages common to both agents. Tolerance spaces are used to model different types of perceptive capabilities associated with agents and a schema is provided which takes into account the perceptive capabilities of both the asking and answering agents. Many of the intuitions behind the framework are based on those from rough set theory. Assumptions under which the techniques will work are made very explicit and all the techniques required to experiment with this approach have been implemented. In many practical cases, the techniques used in the framework are all tractable.

Acknowledgments

Andrzej Szałas and Patrick Doherty are supported by a project grant from the Knut and Alice Wallenberg Foundation, Sweden. Witold Łukaszewicz is supported by KBN grant 8 T11C 009 19.

[8]An implementation of the DLS algorithm is available via `http://www.ida.liu.se/~patdo/aiicssite1/software/index.html`.

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1996. *Foundations of Databases*. Addison-Wesley Pub. Co.

Bailin, S. C., and Truszkowski, W. 2002. Ontology negotiations between intelligent information agents. *The Knowledge Engineering Review* 17(1):7–19.

Berners-Lee, T. 2000. *Weaving the Web*. HarperBusiness.

Doherty, P., and Szałas, A. 2004. On the correspondence between approximations and similarity. In *Proceedings of the 4th International Conference on Rough Sets and Current Trends in Computing*.

Doherty, P.; Łukaszewicz, W.; and Szałas, A. 1997. Computing circumscription revisited. *Journal of Automated Reasoning* 18(3):297–336.

Doherty, P.; Łukaszewicz, W.; and Szałas, A. 1999. Declarative PTIME queries for relational databases using quantifier elimination. *Journal of Logic and Computation* 9(5):739–761.

Doherty, P.; Łukaszewicz, W.; and Szałas, A. 2001. Computing strongest necessary and weakest sufficient conditions of first-order formulas. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 145 – 151.

Doherty, P.; Łukaszewicz, W.; and Szałas, A. 2003. Tolerance spaces and approximative representational structures. In Günter, A.; Kruse, R.; and Neumann, B., eds., *Proceedings of the 26th German Conference on Artificial Intelligence*, volume 2821 of *LNAI*, 475–489. Springer-Verlag.

Ebbinghaus, H.-D., and Flum, J. 1995. *Finite Model Theory*. Heidelberg: Springer-Verlag.

Gabbay, D. M., and Ohlbach, H. J. 1992. Quantifier elimination in second-order predicate logic. In Nebel, B.; Rich, C.; and Swartout, W., eds., *Proc. of the 3rd Int'l Conference on the Principles of Knowledge Representation and Reasoning*. Morgan Kauffman. 425–435.

Immerman, N. 1998. *Descriptive Complexity*. New York, Berlin: Springer-Verlag.

Lin, F. 2000. On strongest necessary and weakest sufficient conditions. In Cohn, A.; Giunchiglia, F.; and Selman, B., eds., *Proc. of the 7th Int'l Conference on the Principles of Knowledge Representation and Reasoning*, 167–175.

Nonnengart, A., and Szałas, A. 1998. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In Orłowska, E., ed., *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, volume 24, 307–328. Springer Physica-Verlag.

Nonnengart, A.; Ohlbach, H.; and Szałas, A. 1999. Elimination of predicate quantifiers. In Ohlbach, H., and Reyle, U., eds., *Logic, Language and Reasoning. Essays in Honor of Dov Gabbay, Part I*, 159–181. Kluwer.

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