Fundamenta Informaticae LXXII (2006) 1001–1016
IOS Press

Calculi of Approximation Spaces

Andrzej Skowron*
Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warszawa, Poland, skowron@mimuw.edu.pl

Jarosław Stepaniuk†
Department of Computer Science, Białystok University of Technology
Wiejska 45a, 15-351 Białystok, Poland, jstepan@ii.pb.bialystok.pl

James Peters ‡
Department of Electrical and Computer Engineering, University of Manitoba
Winnipeg, Manitoba R3T 5V6, Canada, jfpeters@ee.umanitoba.ca

Roman Swiniarski
Department of Mathematical and Computer Sciences, San Diego State University
5500 Campanile Drive San Diego, CA 92182, USA, rswiniar@sciences.sdsu.edu
Institute of Computer Science, Polish Academy of Sciences, Ordona 21, 01-237 Warsaw, Poland

Abstract. This paper considers the problem of how to establish calculi of approximation spaces. Approximation spaces considered in the context of rough sets were introduced by Zdzisław Pawlak more than two decades ago. In general, a calculus of approximation spaces is a system for combining, describing, measuring, reasoning about, and performing operations on approximation spaces. An approach to achieving a calculus of approximation spaces that provides a basis for approximating reasoning in distributed systems of cooperating agents is considered in this paper. Examples of basic concepts are given throughout this paper to illustrate how approximation spaces can be beneficially used in many settings, in particular for complex concept approximation. The contribution of this paper is the presentation of a framework for calculi of approximation spaces useful for approximate reasoning by cooperating agents.

Keywords: rough sets, approximation spaces, concept approximation, learning, approximate reasoning.

Address for correspondence: Jarosław Stepaniuk, Department of Computer Science, Białystok University of Technology, Wiejska 45a, 15-351 Białystok, Poland

*The research has been supported by the grant 3 T11C 002 26 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

†The research has been supported by the grant 4 T11C 014 25 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

‡This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986 and grant T247 from Manitoba Hydro.

1. Introduction

This paper considers the problem of how to establish calculi of approximation spaces. Approximation spaces are fundamental structures for the rough set approach [18, 40, 23] and have been the focus of a number of recent investigations (see, e.g., [4, 5, 7, 20, 24, 22, 40]). The term $calculus$ has been attributed to G. W. v. Leibniz. Leibniz thought of a calculus as an instrument of discovery inasmuch as it provides a system for combining, describing, measuring, reasoning about and performing operations on objects of interest such as terms of a logical formula in a logical calculus or infinitesimally small quantities in a differential calculus(see, e.g., [11]). The calculus of classes described by Alfred Tarski [47] shares some of the features found in a calculus of approximation spaces. The term $class$ is synonymous with set, an assemblage of distinct entities, either individually specified or which satisfy certain specified conditions (e.g., equivalence class of y consisting of all objects equivalent to y). In a calculus of classes, the kinds of classes (e.g., the empty class and the universal class), relations between classes (e.g., inclusion, overlap, identity), and operations on classes $(\cup, \cap, -)$ are specified. Similarly, the calculus of approximation spaces distinguishes between kinds of spaces (e.g., completely and incompletely known approximation spaces), relations between approximation spaces (e.g., inclusion), and operations on approximation spaces.

In this paper we present a generalization of approximation spaces. Using such approximation spaces we show how the rough set approach can be used for approximation of concepts assuming that only partial information on approximation spaces is available. Hence, searching for concept approximation, i.e., the basic task in machine learning and pattern recognition can be formulated as searching for relevant approximation spaces. In the paper, we also characterize approximation space operations called constrained sums that are used in searching for complex concept approximation. We also discuss an important role of constrained sums in hierarchical modeling and in approximate reasoning. Constrained sums are generic operations for approximate reasoning in distributed environments and in multiagent systems.

Our approach is also related to the perception based computing. Studying cognition and in particular, perception based computing [1, 2, 10, 9, 14, 15, 16, 49] is becoming now one of the very active research direction for methods of complex concept approximation [5, 4, 17, 27, 20] and in the consequence for building intelligent systems.

The paper is organized as follows. In Sect. 2, we recall the definition of approximation spaces. Next, we describe a constructive approach for computing values of uncertainty and rough inclusion functions. These functions are the basic components of approximation spaces. Parameters of the uncertainty and rough inclusion functions are tuned in searching for relevant approximation spaces. Among such parameters, we distinguish sensory environments and their extensions. These parameters are used for constructive definition of uncertainty and rough inclusion functions. In particular, we show how information systems can be defined from such approximation spaces. We recall the basic definitions of concept approximation using the approximation spaces. The presented examples of approximation spaces are showing that the discussed approach generalizes several known approaches to approximation in rough set theory. Next, in Sect. 3, we discuss the problem of concept approximation under assumption that only a partial information about approximation spaces is available. In this case, to decide if a given object belongs to the upper or lower approximation of a given concept, it is necessary to estimate the exact value of the rough inclusion function for the neighborhood of this object and the approximated concept because the exact value may be not available. We present an illustrative example for such estimation following a well known heuristic used for rule based classifier construction. In this way, we are showing that searching for relevant approximation spaces is closely related to the basic task of classifier construction in machine learning and in pattern recognition. Searching methods for relevant approximation spaces for complex concept approximation are important in hierarchical learning and in approximate reasoning, in particular in distributed environments and multiagent systems. The search space is equal to the set generated from some generic approximation spaces and some special operations on approximation spaces that we call constrained sums. We discuss applications of such spaces for complex concept approximation using hierarchical learning, approximation of vague dependencies and ontology approximation. Sect. 4 introduces an approximation space-based framework for doing research in modeling complex dynamical systems and analysis of their behavior. First, we present some general remarks and next, in Sect. 4.1, operations on approximation spaces are considered. In Sect. 4.2, we discuss approximation spaces in the hierarchical learning framework and, in Sect. 4.3, we describe an approach to reinforcement learning using approximation spaces.

2. Approximation Spaces

In this section we recall the definition of an approximation space from [34, 44].

Definition 2.1. A *parameterized approximation space* is a system $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$, where

- U is a non-empty set of objects,
- $I_{\#} : U \rightarrow P(U)$ is an uncertainty function, where $P(U)$ denotes the power set of U,
- $\nu_{\$} : P(U) \times P(U) \rightarrow [0, 1]$ is a rough inclusion function,

and $\#, \$$ denote vectors of parameters (the indexes $\#, \$$ will be omitted if it does not lead to misunderstanding).

Observe, that in the above definition, we specify, in fact, a family of approximation spaces defined by the ranges in which parameters can be changed. Searching for relevant approximation space for the target task is an important and computationally hard task in concept approximation. Parameters of approximation spaces can be of different forms. They can be related to such things as feature (attribute) selection, tolerance threshold setting, and tuning of inclusion measures (see, e.g., [36]).

2.1. Uncertainty function

The uncertainty function defines for every object x a set of objects described similarly to x. The set $I(x)$ is called the neighborhood of x (see, e.g., [18, 34]).

We assume that the values of the uncertainty function are defined using a *sensory environment*, i.e., a pair $(L, \|\cdot\|_U)$, where L is a set of formulas, called the *sensory formulas*, and $\|\cdot\|_U : L \longrightarrow P(U)$ is the *sensory semantics*. We assume that for any sensory formula α and any object $x \in U$, the information is available if $x \in \|\alpha\|_U$ holds. The set $\{\alpha : x \in \|\alpha\|_U\}$ is called the *signature of* x in AS and is denoted by $Inf_{AS}(x)$. For any $x \in U$, the *set* $\mathcal{N}_{AS}(x)$ of *neighborhoods of* x in AS is defined by

$\{\|\alpha\|_U : x \in \|\alpha\|_U\}$, and from this set the neighborhood $I(x)$ is constructed. For example, $I(x)$ is defined by selecting an element from the set $\{\|\alpha\|_U : x \in \|\alpha\|_U\}$ or by $I(x) = \bigcap \mathcal{N}_{AS}(x)$. Observe that any sensory environment $(L, \|\cdot\|_U)$ can be treated as a parameter of I from the vector $\#$ (see Definition 2.1).

Let us consider two examples. Any decision table $DT = (U, A, d)$ [18] defines an approximation space $AS_{DT} = (U, I_A, \nu_{SRI})$, where, as we will see, $I_A(x) = \{y \in U : a(y) = a(x) \text{ for all } a \in A\}$. Any sensory formula is a descriptor, i.e., a formula of the form $a = v$, where $a \in A$ and $v \in V_a$ with the standard semantics $\|a = v\|_U = \{x \in U : a(x) = v\}$. Then, for any $x \in U$ its signature $Inf_{AS_{DT}}(x)$ is equal to $\{a = a(x) : a \in A\}$ and the neighborhood $I_A(x)$ is equal to $\bigcap \mathcal{N}_{AS_{DT}}(x)$. Another example can be obtained assuming that for any $a \in A$, there is given a tolerance relation $\tau_a \subseteq V_a \times V_a$ (see, e.g., [34]). Let $\tau = \{\tau_a\}_{a \in A}$. Then, one can consider a tolerance decision table $DT_\tau = (U, A, d, \tau)$ with tolerance descriptors $a =_{\tau_a} v$ and their semantics $\|a =_{\tau_a} v\|_U = \{x \in U : v\tau_a a(x)\}$. Any such tolerance decision table $DT_\tau = (U, A, d, \tau)$ defines the approximation space $AS_{DT_\tau} = (U, I_A, \nu_{SRI})$ with the signature $Inf_{AS_{DT_\tau}}(x) = \{a =_{\tau_a} a(x) : a \in A\}$, and the neighborhood $I_A(x) = \bigcap \mathcal{N}_{AS_{DT_\tau}}(x)$ for any $x \in U$.

The fusion of $\mathcal{N}_{AS_{DT_\tau}}(x)$ for computing the neighborhood of x can have many different forms; the intersection is only an example. One can also consider some more general uncertainty functions, e.g., with values in $P^2(U)$ [40]. For example, to compute the value of $I(x)$, some subfamilies of $\mathcal{N}_{AS}(x)$ can first be selected and the family consisting of intersection of each such a subfamily, is next taken as the value of $I(x)$.

Notice that any sensory environment $(L, \|\cdot\|_U)$ defines an information system with the universe U of objects. Any row of such an information system for an object x consists of information if $x \in \|\alpha\|_U$ holds, for any sensory formula α. Let us also observe that in our examples, we have used a simple sensory language defined by descriptors of the form $a = v$. One can consider a more general approach by taking, instead of the simple structure $(V_a, =)$, some other relational structures R_a with the carrier V_a for $a \in A$ and a signature τ. Then any formula (with one free variable) from a sensory language with the signature τ that is interpreted in R_a defines a subset $V \subseteq V_a$ and induces a neighborhood on the universe of objects consisting of all objects having values of the attribute a in the set V. Notice that this is the basic step in hierarchical modelling [39].

2.2. Rough inclusion function

One can consider general constraints which the rough inclusion functions should satisfy. Searching for such constraints initiated investigations resulting in the discovery and development of rough mereology (see, e.g., [28]). In this subsection, we present only some examples of rough inclusion functions.

The rough inclusion function $\nu_\$: P(U) \times P(U) \to [0, 1]$ defines the degree of inclusion of X in Y, where $X, Y \subseteq U$.

In the simplest case, rough inclusion can be defined by (1) (see, e.g., [34, 18]).

$$\nu_{SRI}(X, Y) = \begin{cases} \frac{card(X \cap Y)}{card(X)} & \text{if} \quad X \neq \emptyset \\ 1 & \text{if} \quad X = \emptyset. \end{cases} \tag{1}$$

This measure is widely used by the data mining and rough set communities. It is worth mentioning that Jan Łukasiewicz [13] was the first one who used this idea to estimate the probability of implications. However, rough inclusion can have a much more general form than inclusion of sets to a degree (see, e.g., [28, 40]).

Another example of a rough inclusion function ν_t can be defined using standard rough inclusion and a threshold $t \in (0, 0.5)$ as shown in (2).

$$\nu_t(X,Y) = \begin{cases} 1 & \text{if} \quad \nu_{SRI}(X,Y) \geq 1-t \\ \frac{\nu_{SRI}(X,Y)-t}{1-2t} & \text{if} \quad t \leq \nu_{SRI}(X,Y) < 1-t \\ 0 & \text{if} \quad \nu_{SRI}(X,Y) \leq t. \end{cases} \tag{2}$$

The rough inclusion function ν_t is used in the variable precision rough set approach [52]. Another example of rough inclusion is used for relation approximation [43, 44] and function approximation [40]. Then the inclusion function ν^* for subsets $X, Y \subseteq \mathcal{R}$ and $\mathcal{R}$ is the set of reals, is defined by Eq. 3.

$$\nu^*(X,Y) = \begin{cases} \frac{card(\pi_1(X \cap Y))}{card(\pi_1(X))} & \text{if} \quad \pi_1(X) \neq \emptyset \\ 1 & \text{if} \quad \pi_1(X) = \emptyset. \end{cases} \tag{3}$$

where π_1 is the projection operation on the first coordinate. Assume now, that X is a cube and Y is the graph $G(f)$ of the function $f : \mathcal{R} \longrightarrow \mathcal{R}$. Then, e.g., X is in the lower approximation of f if the projection on the first coordinate of the intersection $X \cap G(f)$ is equal to the projection of X on the first coordinate. This means that the part of the graph $G(f)$ is "well" included in the box X, i.e., for all arguments that belong to the box projection on the first coordinate the value of f is included in the projection of the box X on the second coordinate.

Usually, there are several parameters that are tuned in searching for a relevant rough inclusion function. Such parameters are listed in the vector $\#$. An example of such a parameter is the threshold mentioned for the rough inclusion function used in the variable precision rough set model. We would like to mention some other important parameters. Among them are pairs $(L^*, \|\cdot\|_U^*)$ where L^* is an extension of L and $\|\cdot\|_U^*$ is an extension of $\|\cdot\|_U$, where $(L, \|\cdot\|_U)$ is a sensory environment. For example, if L consists of sensory formulas $a = v$ for $a \in A$ and $v \in V_a$ then one can take as L^* the set of descriptor conjunctions. For rule based classifiers, we search for relevant patterns for decision classes in such a set of formulas. We present more details in Sect 2.3.

2.3. Lower and upper approximations

The lower and the upper approximations of subsets of U are defined as follows.

Definition 2.2. For any approximation space $AS_{\#,\$} = (U, I_\#, \nu_\$)$ and any subset $X \subseteq U$, the lower and upper approximations are defined by

$LOW\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_\$\left(I_\#(x), X\right) = 1\}$,

$UPP\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_\$\left(I_\#(x), X\right) > 0\}$, respectively.

The lower approximation of a set X with respect to the approximation space $AS_{\#,\$}$ is the set of all objects which can be classified with certainty as objects of X with respect to $AS_{\#,\$}$. The upper approximation of a set X with respect to the approximation space $AS_{\#,\$}$ is the set of all objects which can be possibly classified as objects of X with respect to $AS_{\#,\$}$.

Several known approaches to concept approximations can be covered using the approximation spaces discussed here, e.g., the approach given in [18], approximations based on the variable precision rough set model [52] or tolerance (similarity) rough set approximations (see, e.g., [34] and references in [34]).

Classification methods for concept approximation developed in machine learning and pattern recognition make it possible to decide for a given object if it belongs to the approximated concept or not [8]. The classification methods yield the decisions using only partial information about approximated concepts. This fact is reflected in the rough set approach by assumption that concept approximations should be defined using only partial information about approximation spaces. To decide if a given object belongs to the (lower or upper) approximation of a given concept, the rough inclusion function values are needed. In the next section, we show how such values necessary for classification making, are estimated on the basis of available partial information about approximation spaces.

3. Concept Approximation by Partial Information about Approximation Spaces

In machine learning and pattern recognition [8], we often search for approximation of a concept $C \subset U^*$ in an approximation space $AS^* = (U^*, I^*, \nu^*)$ having only a partial information about AS^* and C, i.e., information restricted to a sample $U \subset U^*$. Let us denote the restriction of AS^* to U by $AS = (U, I, \nu)$, i.e., $I(x) = I^*(x) \cap U$, $\nu(X,Y) = \nu^*(X,Y)$ for $x \in U$, and $X, Y \subseteq U$.

To decide if a given object x belongs to the lower approximation or to the upper approximation of $C \subset U^*$, it is necessary to know the value $\nu^*(I^*(x), C)$. However, in case there is only partial information about the approximation space AS^* available, one must make an estimation of such a value rather than its exact value. In machine learning, pattern recognition or data mining, different heuristics are used for estimation of the values of ν^*. Using different heuristic strategies, values of another function ν' are computed and they are used for estimation of values of ν^*. Then, the function ν' is used for deciding if objects belong to C or not. Hence, we define an approximation of C in the approximation space $AS' = (U^*, I^*, \nu')$ rather than in AS^*. Usually, it is required that the approximations of $C \cap U$ in AS and AS' are close (or the same). If a new portion of objects extending the sample U to U_1 is received, then the closeness of approximations of C in the new approximation space $AS_1 = (U_1, I_1, \nu_1)$ (where I_1, ν_1 are obtained by restriction of I^*, ν^* to U_1) with approximations over AS' restricted to U_1 is verified. If the approximations are not close enough, then the definition of ν' is modified using new information about the extended sample. In this way, we gradually improve the quality of approximation of C on larger parts of the universe U^*. This idea is presented in Figure 1.

Now we would like to explain in more detail a method for estimation of values $\nu^*(I^*(x), C)$. Let us consider an illustrative example. In the example we follow a method often used in rule based classifiers [40]. The method is based on the following steps. First, a set of patterns that are used as left hand sides of decision rules, is induced. Each pattern describes a set of objects in U^* with a satisfactory degree of inclusion to one of decision classes (C or $U^* - C$ for the binary decision). Next, for any object the set of all such patterns that are matched to a satisfactory degree by the given object is extracted. Finally, it is applied a conflict resolution strategy (e.g., voting) for resolving conflicts between votes for different decisions by the matched patterns.

We now present an illustrative example to describe this process more formally in the framework of approximation spaces. First, we assume that among parameters of rough inclusion functions are pairs $(PAT, \|\cdot\|_{U^*})$, where PAT is a set of descriptor conjunctions over a set of condition attributes and $\|\cdot\|_{U^*} : PAT \longrightarrow P(U^*)$ is the semantics of patterns in U^*. Using such parameters we estimate the value $\nu^*(\|pat\|_{U^*}, C)$ by $\nu(\|pat\|_U, C \cap U)$, for any $pat \in PAT$, and we obtain, for a given threshold

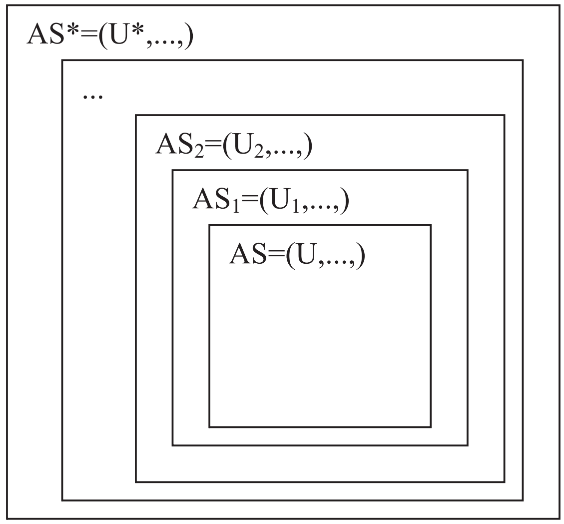

Figure 1. Partial information about approximation space AS^*

$deg \in [0,1]$, the set S_1 of all patterns pat such that $\nu(\|pat\|_U, C \cap U) \geq deg$, i.e., consisting of patterns "for" the concept C. In an analogous way we obtain the set S_2 of all patterns pat satisfying $\nu^*(\|pat\|_{U^*}, U^* - C) \geq deg$. S_2 consists of patterns "for" the complement $U^* - C$ of the concept C. Next, we estimate $\nu^*(I^*(x), \|pat\|_{U^*})$ for $pat \in S_i$, for $i = 1, 2$. To do this we use our assumption on computing $I^*(x)$ for $x \in U^*$. We assume that the sensory formulas from L are descriptors $a = v$ over the condition attributes from a given set of condition attributes A with the semantics in U^* defined by $\|a = v\|_{U^*} = \{x \in U^* : a(x) = v\}$ for $a \in A$ and $v \in V_a$, where V_a is the value set of a. We also have $I^*(x) = \{y \in U^* : Inf_A(x) = Inf_A(y)\}$, where $Inf_A(x) = \{(a, a(x)) : a \in A\}$.

Often, we estimate $\nu^*(I^*(x), \|pat\|_{U^*})$ using a matching strategy based on similarity of the syntactic description of x by $Inf_A(x)$ and the pattern pat. In this way we obtain for a given x the set S'_i of all patterns $pat \in S_i$ (for $i = 1, 2$) such that $\nu(I^*(x) \cap U, \|pat\|_U) \geq deg_1$ where $deg_1 \in [0,1]$ is a given threshold. Finally, the estimation $\nu'(I^*(x), C)$ of the value $\nu^*(I^*(x), C)$ is obtained by application to the sets S'_1, S'_2 a conflict resolution strategy for resolving conflicts between patterns "for" and "against" the membership of x to C.

Usually, the function ν' is parameterized, e.g., by a threshold to which at least the patterns should be included into the decision classes. Also the discussed sets of patterns are among parameters of ν' tuned in the process of rule based classifier construction. Moreover, matching strategies used for estimation of matching degrees are usually parameterized and such parameters are also among tuned parameters of ν'. In machine learning, pattern recognition and data mining, many different searching techniques have been developed for inducing concept approximations of the high quality. Among such components are relevant features, patterns, measures of closeness, model quality measures.

The approximation spaces defined above have been generalized in [40] to approximation spaces consisting of information granules.

4. Approximation Spaces and Constrained Sums in Distributed Environments

In this section, we outline an approach to approximate reasoning in distributed environments using approximation spaces and operations on approximation spaces, called the constrained sums. The approximation spaces are allocated in different agents [12] and they are used to approximate concepts, and to discover new concepts as well as dependencies between concepts. The agents or their teams can interact what helps them to construct new approximation spaces more relevant for understanding of their environments (e.g., to better approximate concepts or to discover new concepts) and to select more relevant actions to achieve agent goals. The interactions are realized by constrained sums. We would like to emphasize that the dynamics of such a system is based on learning by individual agents and agent teams new concepts and dependencies that are next used for performing further actions by agents or their teams.

To illustrate the importance of approximation spaces in modelling by multiagent systems of complex dynamical systems, we briefly discuss some aspects of such a modelling.

1. Assume that in the environment there are distinguished units (agents). Each agent is equipped with an ontology of vague concepts and approximation spaces used for approximation of concepts and dependencies from the ontology. Any agent, to achieve his goal, is performing some actions with pre- and post-conditions specified by concepts from ontology. Concepts are learned hierarchically starting with some sensory concepts. Observe, that an agent ontology can be dynamically changed when agents learn more about the environment. It means that the agent ontology is evolving in time.

2. Agents can identify complex situations by means of their local theories that are represented by sets of dependencies between concepts. Only approximations of concepts and dependencies between them are available for agents.

3. The changes recorded by agents in the satisfiability of concepts make it possible for them to reason about possible changes in the satisfiability of other concepts available in their theories. Here, approximate reasoning schemes (AR schemes) play an important role [36].

4. To better approximate their concepts or to discover new concepts, agents cooperate. Their interactions are described by means of constrained sums [3, 37, 38].

5. Through cooperation, agents learn new patterns, concepts and dependencies between them considered within the framework provided by approximations spaces (see, e.g. [20, 19, 21, 24, 22]). This kind of interaction of agents with their environments can be used to develop tools in reinforcement learning for selection of relevant actions making it possible to achieve goals by agents and agent teams. For example, agents can recognize through identification of changes in observed situations that performing some sequences of actions will be more relevant for them from the point of view of their goals. Observe, that this is much more general approach to reinforcement learning than the approach traditionally used (see, e.g. [30, 31, 46, 50]).

6. Agents can negotiate to form coalitions in which they are linked by special constrained sums or their composition. Each coalition is constructing a space of constrained sums over the approximation spaces of members of the coalition. In such a construction, negotiations play important role.

Any coalition will be involved in searching for relevant constrained sums in which can be defined relevant patterns for approximation of complex concepts important for coalition. To do this the coalition should specify the search space of constrained sums and the searching strategies over such a space.

7. The system has a hierarchical structure, e.g., negotiations related to further searching for relevant approximation spaces, concept approximations and dependencies between concepts can be performed on the coalition level.

8. The important problem to consider is when and how to fuse discovered patterns into a new concept. This is related to the necessity of information granulation. This can save the required memory for storing patterns and at the same time still is making it possible to achieve goals.

From different points discussed the above, it follows that our approach to modelling of a dynamical system by a multiagent system can be understood as a system of evolving local theories of agents belonging to the system. These theories, are changing in time as the result of interactions between agents. The agents learn to select the relevant behaviour for their goals on the basis of properties of changes in their theories.

Observe that in the described modeling, the very basic notions are approximation spaces and constrained sums that can generate new approximation spaces. Constructing relevant constrained sums requires negotiations and conflict resolution between agents constructing approximation spaces for different concepts relevant to local and global goals. Hence, strategies for coalition formation in cooperative searching for relevant approximation spaces are needed.

The methodology based on approximation spaces seems to be promising for modelling of complex dynamical systems, e.g., in which human beings cooperate with robots. In the following sections, we discuss the basic operations on approximation spaces defined by constrained sums and two examples of applications of calculi on approximation spaces in hierarchical learning and reinforcement learning.

4.1. Operations on approximation spaces

In this section, we introduce operations on approximation spaces called constrained sums of approximation spaces. On the basis of such operations we have developed a methodology for discovery of relevant patterns for complex concept approximations (see [3, 37, 38]), e.g., in hierarchical learning, ontology approximation, and spatio-temporal reasoning (see, e.g., [5, 6, 17, 39]). This methodology is also relevant for approximate reasoning in distributed environments.

We assume that for approximation spaces

$$AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$$

considered in this section, the following conditions are satisfied (see Sect. 2.1 and Sect. 2.2):

1. The values of the uncertainty function $I_{\#}$ are defined using the *sensory environment* $(L, \|\cdot\|_U)$ of $AS_{\#,\$}$, where L is a set of formulas, called the *sensory formulas*, and $\|\cdot\|_U : L \longrightarrow P(U)$ is the *sensory semantics*. The sensory environment is one of the components of the vector $\#$.

2. The values $I_{\#}(x)$ are defined by $\bigcap \mathcal{N}_{AS}(x)$ for any $x \in U$.

3. $\nu = \nu_{SRI}$ (i.e., ν is the standard rough inclusion, see Sect. 2)

4. Only a partial information about the approximation space $AS_{\#,\$}$ is given, i.e., the restriction of $AS_{\#,\$}$ to a subset $U_o \subseteq U$.

5. The values of the rough inclusion function $\nu_{\$}$ are estimated using a pair $(L^*, \|\cdot\|^*_U)$ where L^* is an extension of L and $\|\cdot\|^*_U$ is an extension of $\|\cdot\|_U$, where $(L, \|\cdot\|_U)$ is the sensory environment of $AS_{\#,\$}$. The pair $(L^*, \|\cdot\|^*_U)$ is one of the components of the vector $\$$. The formulas from L^* are patterns that are used for estimation of values of $\nu_{\$}$ (see Sect. 3).

Now, operations on approximation spaces, by analogy to the constrained sums of information systems [3, 37, 38], can be defined as follows. To simplify the notation we consider only the case of binary operations.

For approximation spaces AS^i where $i = 1, 2$ we consider the class $CONSTRAINT(AS^1, AS^2)$ of all approximation spaces $AS = (U, I, \nu)$ satisfying the following conditions:

1. For estimation of values of ν an extension $(L^*, \|\cdot\|^*_U)$ of $(L^{*,i}, \|\cdot\|^{*,i}_U)$, where $i = 1, 2$ is used. An extension is satisfying the conditions: $L^{*,1} \cup L^{*,2} \subseteq L$ and $\|\alpha\|_U = \|\alpha\|^i_U$ for $\alpha \in L^i$, where $i = 1, 2$. The formulas from $L^* - (L^{*,1} \cup L^{*,2})$ are called constraints. We also assume that any constraint α is a boolean combination of formulas from $L^{*,1} \cup L^{*,2}$, e.g., conjunction of formulas $\alpha_1 \wedge \alpha_2$ for $\alpha_1 \in L^{*,1}$ and $\alpha_2 \in L^{*,2}$.

2. The sensory environment of AS is defined by $(L, \|\cdot\|_U)$.

Any operation o on approximation spaces such that $o(AS^1, AS^2) \in CONSTRAINT(AS^1, AS^2)$ for any approximation spaces AS^1, AS^2 from a generic set $\mathcal{AS}$ of approximation spaces, is called the constrained sum.

4.2. Hierarchical Learning

In hierarchical learning we consider the space $SPACE(\mathcal{AS}, \mathcal{F})$ generated from $\mathcal{AS}$ by the set $\mathcal{F}$ of constrained sums. Searching for relevant approximation spaces from $SPACE(\mathcal{AS}, \mathcal{F})$ is making it possible to discover relevant patterns for concept approximation [5, 6, 17]. Constrained sums of approximation spaces are tools for modelling patterns that are more relevant for approximation of concepts than patterns defined by arguments of the constrained sum.

Assume that approximation spaces AS^1, AS^2 are used for approximation of concepts C_1, C_2 and that the dependency *if* C_1 and C_2 *then* C holds. Notice that usually many such dependencies should be considered for the concept approximation. However, for simplicity of presentation we consider only one.

If the concept C is, in a sense, not *too far* from C_1 and C_2 then one can search in $SPACE(\mathcal{AS}, \mathcal{F})$ for a constrained sum $o(AS^1, AS^2)$ relevant for the approximation of concept C. If $AS = o(AS^1, AS^2)$, then patterns defined in AS are more general than the conjunction of sensory formulas defined by AS^1, AS^2. This happens because new patterns are defined by joining patterns of AS^1, AS^2 relative to constraints. Let us recall that patterns of AS^1, AS^2 belong to the extension of the set of sensory formulas of AS^1, AS^2 (see the definition of $CONSTRAINT(AS^1, AS^2)$). Hence, sensory formulas of AS are conjunction of patterns of AS^1, AS^2 and constraints. By allowing to use patterns instead of sensory formulas in joining AS^1, AS^2, we define a search space for relevant patterns and from such a space relevant joins (relative to constraints) of patterns for approximation of concepts are extracted.

If the concept C is *far* from C_1 and C_2 then we use a hierarchy of dependencies between concepts from domain knowledge in searching for a relevant approximation space for C. The approximation of C is obtained using hierarchical learning [5, 6, 17]. We assume that (1) the generic concepts C_1 and C_2 on the lowest level of the hierarchy can be approximated by some generic approximation spaces, (2) the concepts on a given level that are not generic follow from concepts on the lower level, and (3) the relevant approximation spaces for concepts on a given level of the hierarchy can be discovered using relevant constrained sums of approximation spaces from the previous hierarchy level. Then we proceed as follows. Starting with some generic concepts and approximation spaces relevant for them we construct approximation spaces for concepts on the first level in hierarchy that follow from these generic concepts and can be approximated by relevant constrained sums over these approximation spaces. Next, we perform the same procedure for the recently approximated concepts and the concepts on the next level of hierarchy. We continue the procedure until the target concept is approximated.

Let us consider one more example of applications of constrained sums of approximation spaces for approximation of dependency between vague concepts. This problem is important in ontology approximation [42, 41]. Any concept from the left hand side of a given vague dependency is called its premise and the dependency conclusion is the concept from the right hand side of the dependency. The approximation of a given vague dependency is defined by a method which makes it possible for any object to compute the arguments "for" and "against" the membership of an object to the conclusion of the dependency on the basis of analogous arguments relative to the dependency premisses [41]. Any argument "for" or "against" is a compound information granule (pattern) consisting of a pattern together with a degree to which (at least) this pattern is included to the concept and a degree to which (at least) the analyzed object is included to the pattern. Such arguments "for" and "against" that are relevant for the dependency conclusion are constructed from the arguments "for" and "against" for the dependency premisses by using constrained sums. Such constructions are called the local schemes. Any local scheme (production rule) (see, e.g., [36]) or rough mereological connective (see, e.g., [29]) yields the fusion result of arguments for premisses that is next taken as the argument for the dependency conclusion. By composition of local schemes more advanced fusion schemes are obtained, called approximate reasoning schemes (AR schemes) (see, e.g., [5, 36, 29, 39]). They show how the arguments from premisses of dependencies are fused to arguments for more compound concepts derived in a given ontology from premisses. AR schemes can correspond to different parts of complex spatio-temporal objects. Hence, there is a need for composing AR schemes for parts into AR schemes for objects composed from these parts [39].

4.3. Reinforcement Learning

By way of another illustration of the utility of approximation spaces, a rough set approach to reinforcement learning is briefly considered in this section. The study of reinforcement learning carried out in the context of approximation spaces is the outgrowth of recent work on approximate reasoning and intelligent systems (see, e.g., [28, 27, 32, 33, 21, 23, 25, 26]). An overview of a Monte Carlo approach to reinforcement learning with approximation spaces is given in [22, 25]. The basic problem that provides a setting for reinforcement learning is formulated by [31]: a system is required to interact with its environment to achieve a particular task or goal, and based on the feedback about the current state of the environment, what action should the system perform next? Reinforcement learning itself is the act of learning the correct action to take in a specific situation based on feedback obtained from the environment [46].

Feedback is in the form of a numerical reward that results from an action performed by an agent. Specifically, reinforcement learning can be divided into *off-line* and *on-line* learning. Off-line learning is similar to the idea of a student learning by instruction from a teacher. In effect, the agent is taught what it needs to know before venturing into the environment in which it is to operate. In contrast, on-line learning resembles an infant learning to walk (see, e.g., [22]). Learning occurs in real-time in which the agent is exploring its environment and constantly adding to its experience in order to make better decisions in the future. Learning techniques are typically applied to stationary or non-stationary models of the environment. In stationary models all the state transition probabilities are fixed, whereas in non-stationary models they change over time.

Let $DT = (U, A, d)$ be a decision system corresponding to a swarmbot (system of cooperating bots (see, e.g., [20, 26])), where U is a non-empty set of objects, A is a set of swarmbot features, and d is a distinguished attribute representing a swarmbot decision $d : U \longrightarrow \{0, 1\}$. The decision table $DT = (U, A, d)$ (see Figure 2) provides a record of objects during different episodes in the life of system. Any object x is a tuple (s, ac, r, t) where s is a swarmbot state, r is the reward for the action performed by a swarmbot in the previous step, ac is an action proposed in state s, and t is the time in which the state s has been observed. We assume that in A are attributes $State$, $Action$, $Reward$, $Time$ such that $State(x) = s$, $Reward(x) = r$, $Action(x) = ac$, $Time(x) = t$. Let $AS_{DT,B} = (U, I_B, \nu_{SRI})$ denote an approximation space defined in the context of a decision system $DT_B = (U, B, d)$, where $B \subseteq A$ and $I_B(x)$ is the B-indiscernibility class defined by x. We use the notation s_t to denote a pair $(s, Time(s))$. Further, let $B_{ac}(x)$ be a block in the partition of U containing x relative to action ac (i.e., $B_{ac}(x)$ contains objects with a particular action ac that are B-indiscernible with x). The block $B_{ac}(x)$ where $ac \in V_{Action}$ is defined in (4).

$$B_{ac}(x) = \begin{cases} I_{B\cup\{Action\}}(x), & \text{if } Action(x) = ac, \\ \emptyset, & \text{otherwise.} \end{cases} \tag{4}$$

By B_*D, we denote the set $LOW\,(AS_{DT,B}, D)$, where $D = \{x \in U : d(x) = 1\}$. We assume that $d(x) = 1$ specifies that the action ac in x has been accepted by a swarmbot. B_*D represents certain knowledge expressed by means of attributes from B about the objects in D. For this reason, B_*D provides a useful standard or norm in gaining knowledge about the proximity of objects to what is considered normal. The term $normal$ applied to a set of objects denotes objects with actions which have been accepted by the system. The introduction of some form of standard objects makes it possible to measure the inclusion of blocks of B-indiscernible action-specific objects in the set of those objects that create a B-definable part of a standard. The framework provided by an approximation space makes it possible to derive pattern-based rewards, which are used by swarms that learn to choose actions in response to perceived states of their environment (see, e.g., Figure 2). The notation $\bar{r}_{ac,t}$ denotes an average rough inclusion value computed within the context of an approximation space as shown in (5).

$$\bar{r}_{ac,t} = \frac{\sum_{\{i\in\{0,\ldots,n_x\}:B_{ac}(x_{t-i})\neq\emptyset\}} \nu_{SRI}(B_*D, B_{ac}(x_{t-i}))}{n_x} = \frac{\sum_{\{i\in\{0,\ldots,n_x\}:B_{ac}(x_{t-i})\neq\emptyset\}} card(B_{ac}(x_{t-i}))}{n_x \cdot card(B_*D)}. \tag{5}$$

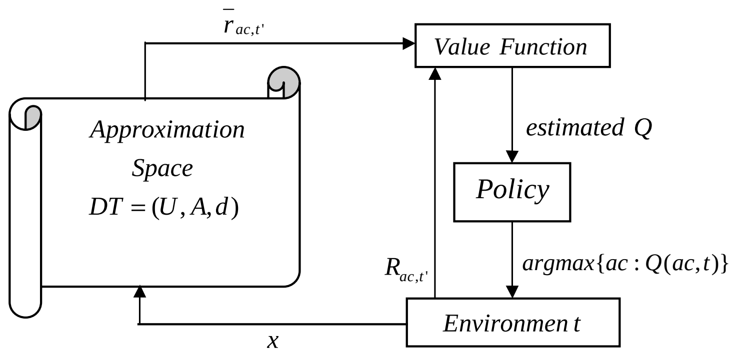

Figure 2. Approximation spaces and reinforcement learning framework

where $Time(x) = t$, $Action(x) = ac$, n_x is the length of window dependent on x bounded on the availability of information[1] on objects in DT, $(t - n_x, \ldots, t)$ denotes the time window of the length n_x, $x_{t-n_x}, \ldots, x_t$ are objects in this time window (i.e., in an episode of the system) recorded in the decision table DT. The last equality holds because we have $B_{ac}(x_{t-i}) \subseteq B_* D$.

The notation $Q(ac, t)$ in Figure 2 denotes the value of an action ac at time t. There are different strategies for computing $Q(ac, t)$. They make it possible to estimate how successful the performance of an action ac at time t can be. For example, let $W_{ac,t}$ denote an average of the weights over a time window (see (6)).

$$W_{ac,t} = \frac{\sum_{i=1}^{n_x} \bar{r}_{ac,t-i}}{n_x}. \tag{6}$$

Let us now assume that $R_{ac,t'}$ denotes the (discounted) sum of the rewards (returns) during an episode for the action ac over the time window ending at t where $t < t' < t + 1$. Then the value $Q(ac, t)$ can be defined by (7).

$$Q(ac, t) = \begin{cases} \sum_{i=1}^{n_x} \frac{\bar{r}_{ac,t-i}}{W_{ac,t-i}} \left[R_{ac,t-i} - Q(ac, t-i)\right], & \text{if} \quad n_x > 1, \\ r_{ac,t-1}, & \text{otherwise.} \end{cases} \tag{7}$$

where x is such an object from DT that $t = Time(x)$ and $r_{ac,t-1}$ is the reward for the action ac performed at time $t - 1$.

In our example, after computing $Q(ac, t)$ for all actions, we extract the best action using a greedy policy defined by *argmax* (see Figure 2). The notation $argmax\{ac : Q(ac, t)\}$ in Figure 2 corresponds to a "stimulus", which is one of the four $whys$ introduced by Niko Tinbergen [48] to explain observed behaviors (see, e.g., [20, 22]). Another example of an approximation space-based learning strategy has been successfully used in a new form of off-policy Monte Carlo reinforcement learning (see, e.g., [22, 25]).

[1] Assuming a fixed length $T > 0$ of the time window and letting m_x denote the number of all predecessors (with respect to time order) of x in DT, one can assume that $n_x = min\{T, m_x\}$. In (5) and (6), we let $n_x \geq 1$.

Conclusions

Steps towards the discovery of a calculus of approximation spaces in distributed systems have been presented in this paper. This article has included consideration of an approximation of concepts using the rough set approach. In particular, the role of approximation spaces in modelling of complex dynamical multiagent systems has been emphasized. As part of the research described in this paper, we are developing evolutionary strategies searching for local agent behavioral rules in systems based on approximation spaces. These rules should make it possible not only to achieve local goals of agents but also global goals of complex dynamical systems, e.g., preserving some global invariants. Observe, that these evolutionary strategies can evolve in time as the result of adaptive changes of their parameters. It is also the case that the calculus of approximation spaces considered in this paper, has significant implications for an approach to unsupervised learning by cooperating agents in non-stationary environments.

References

[1] Albus, J.S., Meystel, A. M.: *Engineering of mind: An introduction to the science of intelligent systems*. John Wiley, New York, 2001.

[2] Anderson, J.R.: *Rules of the Mind*. Lawrence Erlbaum, Hillsdale, NJ, 1993.

[3] Barwise, J., Seligman, J.: *Information Flow: The Logic of Distributed Systems*. Cambridge University Press Tracts in Theoretical Computer Science **44**, 1997.

[4] Bazan, J., Peters, J., F., Skowron, A.: Behavioral pattern identification through rough set modelling. In: LNAI **3642**, Springer, Heidelberg, 2005, 688-697.

[5] Bazan, J., Skowron, A.: Classifiers based on approximate reasoning schemes. In: Dunin-Kepliçz, B., Jankowski A., Skowron A., and Szczuka M., (Eds.), *Monitoring, Security, and Rescue Techniques in Multiagent Systems MSRAS*, Advances in Soft Computing. Springer, Heidelberg, 2005, 191–202.

[6] Bazan, J., Nguyen, S. Hoa, Nguyen, H. Son, Skowron, A.: Rough set methods in approximation of hierarchical concepts. *Proc. of RSCTC'2004*, LNAI **3066**, Springer, Heidelberg, 2004, 346–355.

[7] Gomolinska, A.: Rough validity, confidence, and coverage of rules in approximation spaces. LNCS **3400**, *Transactions on Rough Sets* **III**, Springer, Heidelberg, 2005, 57–81.

[8] Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. Springer-Verlag, Heidelberg, 2003.

[9] Kieras, D., Meyer, D.E.: An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction* **12**: 391-438, 1997.

[10] Langley, P., Laird, J.E.: Cognitive architectures: Research issues and challenges. *Technical Report*, Institute for the Study of Learning and Expertise, Palo Alto, CA, 2002.

[11] Leibniz, G.W., *Opuscles et fragments inedits de Leibniz*, edited by L. Couturat. Paris, 1903, 256.

[12] Luck, M., McBurney, P., Preist, Ch.: *Agent Technology: Enabling Next Generation. A Roadmap for Agent Based Computing*, Agent Link, 2003.

[13] Łukasiewicz, J.: Die logischen Grundlagen der Wahrscheinilchkeitsrechnung, Kraków 1913. In: Borkowski, L. (ed.), *Jan Łukasiewicz - Selected Works*. North Holland, Amstardam, Polish Scientific Publishers, Warsaw, 1970.

[14] Meyer, D. E., Kieras, D. E.: A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review* **104**: 3–65, 1997.

[15] Meyer, D. E., Kieras, D. E.: A computational theory of executive control processes and human multiple-task performance: Part 2. Accounts of Psychological Refractory-Period Phenomena. *Psychological Review* **104**: 749–791, 1997.

[16] Newell, A.: (1990). *Unified Theories of Cognition*. Cambridge, Harvard University Press, MA, 1980.

[17] Nguyen, S. Hoa, Bazan, J., Skowron, A., Nguyen, H. Son: Layered learning for concept synthesis. *Transactions on Rough Sets I: LNCS Journal Subline*, LNCS **3100**, Springer, Heidelberg, 2004, 187–208

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

[19] Peters, J. F.: Approximation spaces for hierarchical intelligent behavioral system models. In: B.D. Keplіçz, A. Jankowski, A. Skowron, M. Szczuka (Eds.), *Monitoring, Security and Rescue Techniques in Multiagent Systems, Advances in Soft Computing*, Physica-Verlag, Heidelberg, 2004, 13–30.

[20] Peters, J.F.: Rough ethology: Towards a biologically-inspired study of collective behavior in intelligent systems with approximation spaces. LNAI **3400**, *Transactions on Rough Sets* **III**: 153-174, Springer, Heidelberg, 2005.

[21] Peters, J. F. : Approximation space for intelligent system design patterns. *Engineering Applications of Artificial Intelligence* **17**(4): 1–8, 2004.

[22] Peters, J.F., Henry, C.: Reinforcement learning with approximation spaces. *Fundamenta Informaticae*, 2006 (to appear).

[23] Peters, J. F., Skowron, A., Stepaniuk, J., Ramanna, S.: Towards an ontology of approximate reason. *Fundamenta Informaticae* **51**(1-2), 157–173, 2002.

[24] Peters, J. F., Henry, C., Ramanna, S.: Reinforcement learning with pattern-based rewards. In: Proc. Fourth Int. IASTED Conf. Computational Intelligence (CI 2005), Calgary, Alberta, Canada (4-6 July 2005) 267–272.

[25] Peters, J.F.: Approximation spaces in off-policy Monte Carlo learning. Plenary paper in T. Burczynski, W. Cholewa, W. Moczulski (Eds.), *Recent Methods in Artificial Intelligence Methods*, AI-METH Series, Gliwice, 2005, 139-144.

[26] Peters, J.F., Henry, C., Lockery, D., Borkowski, M., Ramanna, S.: Approximation spaces for swarms that learn. In T. Burczynski, W. Cholewa, W. Moczulski (Eds.), *Recent Methods in Artificial Intelligence Methods*, AI-METH Series, Gliwice, 2005, 135-138.

[27] Pal, S.K., Polkowski, L., Skowron, A. (Eds.): *Rough-Neural Computing: Techniques for Computing with Words*. Springer-Verlag, Berlin, 2004.

[28] Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. *International Journal Approximate Reasoning* **15**(4): 333–365, 1996.

[29] Polkowski, L., Skowron, A.: Towards adaptive calculus of granules. In: [51], 201–227.

[30] Precup,D.: *Temporal Abstraction in Reinforcement Learning*. Ph.D. thesis, Supervisor: R.S. Sutton, Department of Computer Science, University of Massachusetts, 2000.

[31] Rummery, G.A.: *Problem Solving with Reinforcement Learning*. Ph.D. Thesis, Cambridge University, 1995.

[32] Skowron,A.: Toward intelligent systems: Calculi of information granules. *Bulletin of the Rough Set Society*, **5**(1-2): 9–30, 2001.

[33] Skowron, A., Swiniarski, R.W.: Information granulation and pattern recognition. In: [27], 2004, 599–636.

[34] Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. *Fundamenta Informaticae* **27**: 245–253, 1996.

[35] Skowron, A., Stepaniuk, J.: Information granules: Towards foundations of granular computing. *International Journal of Intelligent Systems* **16**(1): 57–86, 2001.

[36] Skowron, A., Stepaniuk, J.: Information granules and rough-neural computing. In: [27], 43–84.

[37] Skowron, A., Stepaniuk, J., Peters, J.F.: Rough sets and infomorphisms: Towards approximation of relations in distributed environments. *Fundamenta Informaticae* **54**(1-2): 263–277, 2003.

[38] Skowron, A., Stepaniuk, J.: Constrained sums of information systems. In: *Proc. RSCTC 2004*, LNCS **3066**, Springer, Heidelberg, 2004, 300–309.

[39] Skowron, A., Synak, P., Complex patterns. *Fundamenta Informaticae* **60**(1-4): 351–366, 2004.

[40] Skowron, A., Swiniarski, R., Synak, P: Approximation spaces and information granulation. *Transactions on Rough Sets III: LNCS Journal Subline*, LNCS **3400**, Springer, Heidelberg, 2005, 175–189.

[41] Skowron, A., Stepaniuk, J. (2005). Ontological framework for approximation. In: D. Ślęzak, G. Wang, M. Szczuka, I. Düntsch, Y.Y. Yao (Eds.), *Proceedings of the 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing* (RSFDGrC'2005), Regina, Canada, August 31-September 3, 2005, Part I, LNAI **3641**, Springer, Heidelberg, 718–727.

[42] Staab, S., Studer, R., (Eds.): *Handbook on Ontologies*. International Handbooks on Information Systems, Springer, Heidelberg, 2004.

[43] Stepaniuk, J.: Rough relations and logics. In: L. Polkowski, A. Skowron (Eds.), Rough Sets in Knowledge Discovery 1. Methodology and Applications, Physica Verlag, Heidelberg, 1998, 248–260.

[44] Stepaniuk, J.: Knowledge discovery by application of rough set models. In: L. Polkowski, S. Tsumoto, T.Y. Lin (Eds.), *Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems*, Physica–Verlag, Heidelberg, 2000, 137–233.

[45] Stone, P.: *Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer*. MIT Press, Cambridge, MA, 2000.

[46] Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

[47] Tarski, A., *Introduction to logic and to the methodology of deductive sciences*, Oxford University Press, NY, 1965, IV, 68–78.

[48] Tinbergen, N.: On aims and methods of ethology, *Zeitschrift f* ür *Tierpsychologie* **20**: 410–433, 1963.

[49] Veloso, M.M., Carbonell, J.G.: Derivational analogy in PRODIGY: Automating case acquisition, storage, and utilization. *Machine Learning* **10**: 249-278, 1993.

[50] C.J.C.H. Watkins, *Learning from Delayed Rewards*. Ph.D. Thesis, supervisor: Richard Young, King's College, University of Cambridge, UK, May, 1989.

[51] Zadeh, L.A., Kacprzyk, J. (Eds.): *Computing with Words in Information/Intelligent Systems* **1-2**, Physica-Verlag, Heidelberg, 1999.

[52] Ziarko, W., Variable precision rough set model, *Journal of Computer and System Sciences* **46**: 39–59, 1993.