Fundamenta Informaticae 67 (2005) 203–217
IOS Press

Modelling Complex Patterns by Information Systems

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

Jan G. Bazan†
Institute of Mathematics, University of Rzeszów
Rejtana 16A, 35-959 Rzeszów, Poland
bazan@univ.rzeszow.pl

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

Abstract. We outline an approach to hierarchical modelling of complex patterns that is based on operations of sums with constraints on information systems. We show that such operations can be treated as a universal tool in hierarchical modelling of complex patterns.

Keywords: complex patterns, rough sets, approximation spaces, information systems, infomorphisms

*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.
Address for correspondence: 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.

1. Introduction

One of the main tasks in granular computing is to develop calculi of information granules [18], [10], [11]. Information systems used in rough set theory are particular kinds of information granules. In the paper we introduce and study operations on such information granules basic for reasoning in distributed systems of information granules. The operations are called constrained sums. They are developed by interpreting infomorphisms between classifications [2]. In [12] we have shown that classifications [2] and information systems [5] are, in a sense, equivalent. We also extend the results included in [12] on applications of approximation spaces to study properties of infomorphisms. Operations, called constrained sums seem to be very important in searching for patterns in data mining (e.g., in spatio-temporal reasoning) or in a more general sense in generating relevant granules for approximate reasoning using calculi on information granules [12].

This paper is organised as follows. In Section 2 we present basic concepts. In Section 3 we discuss constrained sums of information systems and hierarchical information systems. Applications of constrained sums of information systems in modelling patterns for concept approximation are outlined in Section 4 and Section 5.

2. Approximation Spaces and Infomorphisms

In this section we recall basic notions for our considerations.

2.1. Approximation Spaces

We recall a general definition of an approximation space. Several known approaches to concept approximations can be covered using such spaces, e.g., the tolerance based rough set model.

For every non-empty set U, let $P(U)$ denote the set of all subsets of U.

Definition 2.1. [9], [16] A *parameterised approximation space* is a system $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$, where

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

and $\#, \$$ are denoting vectors of parameters.

The uncertainty function defines for every object x a set of similarly described objects. A set $X \subseteq U$ is *definable in* $AS_{\#,\$}$ if and only if it is a union of some values of the uncertainty function.

The rough inclusion function defines the degree of inclusion between two subsets of U (see, e.g., [9], [16]):

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

This measure is widely used by data mining and rough set communities. However, Jan Łukasiewicz [4] was first who used this idea to estimate the probability of implications.

For example, any information system $IS = (U, A)$ defines for any $B \subseteq A$ an approximation space $AS_B = (U, I_B, \nu_{SRI})$ where $I_B(x)$ is the B-indiscernibility class [5] defined by x.

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

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

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

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

2.2. Infomorphisms

In this section we recall the definition of infomorphism for two information systems [12]. We also present some new properties of infomorphisms. The infomorphisms for classifications are introduced and studied in [2].

We denote by $\Sigma(IS)$ the set of Boolean combinations of descriptors over IS and by $\|\alpha\|_{IS} \subseteq U$ is denoted the semantics of α in IS. More precisely, the set $\Sigma(IS)$ is defined recursively by

1. $(a\ in\ V) \in \Sigma(IS)$, for any $a \in A$ and $V \subseteq V_a$.
2. If $\alpha \in \Sigma(IS)$ then $\neg\alpha \in \Sigma(IS)$.
3. If $\alpha, \beta \in \Sigma(IS)$ then $\alpha \wedge \beta \in \Sigma(IS)$.
4. If $\alpha, \beta \in \Sigma(IS)$ then $\alpha \vee \beta \in \Sigma(IS)$.

The semantics of formulas from $\Sigma(IS)$ with respect to an information system IS is defined recursively by

1. $\|a\ in\ V\|_{IS} = \{x \in U : a\left(x\right) \in V\}$.
2. $\|\neg\alpha\|_{IS} = U - \|\alpha\|_{IS}$.
3. $\|\alpha \wedge \beta\|_{IS} = \|\alpha\|_{IS} \cap \|\beta\|_{IS}$.
4. $\|\alpha \vee \beta\|_{IS} = \|\alpha\|_{IS} \cup \|\beta\|_{IS}$.

For all formulas $\alpha \in \Sigma(IS)$ and for all objects $x \in U$ we will denote $x \models_{IS} \alpha$ if and only if $x \in \|\alpha\|_{IS}$.

Definition 2.3. [2], [12] If $IS_1 = (U_1, A_1)$ and $IS_2 = (U_2, A_2)$ are information systems, then an infomorphism between IS_1 and IS_2 is a pair $(f^\wedge, f^\vee)$ of functions $f^\wedge : \Sigma(IS_1) \rightarrow \Sigma(IS_2)$, $f^\vee : U_2 \rightarrow U_1$, satisfying the following equivalence

$$f^\vee(x) \models_{IS_1} \alpha \text{ if and only if } x \models_{IS_2} f^\wedge(\alpha), \tag{1}$$

for all objects $x \in U_2$ and for all formulas $\alpha \in \Sigma(IS_1)$.

The infomorphism will be denoted shortly by $(f^\wedge, f^\vee) : IS_1 \rightleftarrows IS_2$.

Let us consider a simple example of infomorphism related to object granulation.

Proposition 2.1. Let $IS_1 = (U, A)$ and $IS_2 = (U/IND(A), A)$ be information systems such that $IND(A)$ is the A-indiscernibility relation [5] and $a([x]_{IND(A)}) = a(x)$ for $a \in A$ and $x \in U$ (notice that a on the left hand side of the equality is from IS_2). Then we have two infomorphisms

$$(id, sel) : IS_1 \rightleftarrows IS_2 \quad (id, i) : IS_2 \rightleftarrows IS_1 \tag{2}$$

where $id(\alpha) = \alpha$ for any α definable over IS_1 (IS_2), $sel([x]_{IND(A)}) \in [x]_{IND(A)}$, and $i(x) = [x]_{IND(A)}$ for any $x \in U$.

The above definition of infomorphisms can be generalised by changing the definition of satisfiability relation $\models_{IS}$. Instead of crisp definition (i.e., either $x \models_{IS} \alpha$ or $non(x \models_{IS} \alpha)$ for any $x \in U$) one can use a *rough satisfiability relation* specified by three binary relations $\models^i_{IS} \subseteq U \times \Sigma(IS)$ where $i \in \{0, 1, ?\}$. Such relations can be defined relative to a given indiscernibility relation $IND(B)$, for some $B \subseteq A$, by

1. $x \models^1_{IS,B} \alpha$ if and only if $[x]_{IND(B)} \subseteq \|\alpha\|_{IS}$ (i.e., $x \in LOW\,(AS_B, \|\alpha\|_{IS})$);
2. $x \models^?_{IS,B} \alpha$ if and only if $[x]_{IND(B)} \cap \|\alpha\|_{IS} \neq \emptyset$ and $[x]_{IND(B)} \cap (U - \|\alpha\|_{IS}) \neq \emptyset$ (i.e., $x \in UPP\,(AS_B, \|\alpha\|_{IS}) - LOW\,(AS_B, \|\alpha\|_{IS})$);
3. $x \models^0_{IS,B} \alpha$ if and only if $[x]_{IND(B)} \subseteq \|\alpha\|_{IS}$ if and only if $x \in LOW\,(AS_B, U - \|\alpha\|_{IS})$.

for any $x \in U$ and for any $\alpha \in \Sigma(IS)$. Observe that using the rough set approach we are not restricted to such three valued satisfiability. For example, one can use some additional information about $shades$ of boundary region for introducing more degrees (values). In particular one can consider a rough approximation of fuzzy satisfiability relation. Yet another satisfiability relation should be considered assuming that in set approximation some inductive reasoning is used. Properties of infomorphisms over such satisfiability relations will be studied elsewhere.

3. Constrained Sums of Information Systems

In this section we consider operations on information systems that can be used in searching for hierarchical patterns. The operations are parameterised by constraints. Hence, in searching for relevant patterns one can search for relevant constraints and elementary information systems used to construct hierarchical patterns represented by constructed information systems.

This operation is more general than theta join operation used in databases [3]. We start from the definition in which the constraints are given explicitly.

Definition 3.1. Let $IS_i = (U_i, A_i)$ for $i = 1, \ldots, k$ be information systems and let R be a k-ary constraint relation in $U_1 \times \ldots \times U_k$, i.e., $R \subseteq U_1 \times \ldots \times U_k$. These information systems can be combined into a single information system relatively to R, denoted by $+_R(IS_1, \ldots, IS_k)$, with the following properties:

- The objects of $+_R(IS_1, \ldots, IS_k)$ consist of k-tuples $(x_1, \ldots, x_k)$ of objects from R, i.e., all objects from $U_1 \times \ldots \times U_k$ satisfying the constraint R.

- The attributes of $+_R(IS_1, \ldots, IS_k)$ consist of the attributes of $A_1, \ldots, A_k$, except that if there are any attributes in common, then we make distinct copies, so as not to confuse them.

In the case where $R = U_1 \times \ldots \times U_k$ we will write $+(IS_1, \ldots, IS_k)$ instead of $+_R(IS_1, \ldots, IS_k)$.

Usually the constraints are defined by conditions expressed by Boolean combination of descriptors of attributes (see Section 2.2). It means that the constraints are built from expressions $a\ in\ V$, where a is an attribute and $V \subseteq V_a$, using propositional connectives $\wedge$, $\vee$, $\neg$. Observe, that in the constraint definition we use not only attributes of parts (i.e., from information systems $IS_1, \ldots, IS_k$) but also some other attributes specifying relation between parts.

Let us note that constraints can be defined in two steps. First a partial information about constraints is given by data tables. Next, the constraints are induced from such data tables using machine learning, data mining or rough set technology. This makes the constrained sum different from the theta join [3].

Example 3.1. In some applications constraint relations can be induced from decision tables with positive and negative examples for the characteristic functions of such relations. Let us consider an example of a binary relation R defined on pairs $(x, y) \in U \times U$ of council members defined in the following way: $(x, y) \in R$ if and only if x, y are from the same faculty. We would like to induce a classifier for such a relation from examples of votes of council members for different issues. Observe, that the number of examples can be much smaller then $card(U \times U)$. The examples are stored in a decision table DT. Objects from DT are pairs (x, y) of council members (where $x \neq y$). The condition attributes from DT represent issues on which council members are voting. Some positive and negative examples are stored in DT. Condition attributes have values from $\{1, -1, 0\} \times \{1, -1, 0\}$ where 1 denotes a vote *for* the issue defined by the attribute, -1 a vote *against* the issue, and 0 a *neutral* vote. In $a(x, y) = (i, j)$ i denotes the vote of x for the issue a and j - the vote of y for a. For example, if a denote the issue *new building for Computer Science Faculty,* then $a(x, y) = (1, -1)$ means that x voted *for* the issue a and y voted *against* this issue. From the decision table with such conditions one can induce a rule based classifier $Classifier$ defining the constraint $\{(x, y) \in U \times U : Classifier(x, y) = 1\}$ (see, e.g., [20, 1]).

Let us observe that the information system $+_R(IS_1, \ldots, IS_k)$ can be also described using an extension of the sum $+(IS_1, \ldots, IS_k)$ by adding a new binary attribute that is the characteristic function of the relation R and by taking a subsystem of the received system consisting of all objects having value one for this new attribute.

The constraints used to define the sum (with constraints) can be often specified by information systems. The objects of such systems are tuples consisting of objects of information systems that are arguments of the sum. The attributes describe relations between elements of tuples. One of the attribute is a characteristic function of the constraint relation (restricted to the universe of the information system). In this way we obtain a decision system with the decision attribute defined by the characteristic function of the constraint and conditional attributes are the remaining attributes of this system. From such a decision table one can induce a classifier for the constraint relation. Next, such a classifier can be used to select tuples in the construction of a constrained sum.

The constructed constrained sum of information systems can consist of some incorrect objects. This is due to improper filtering of objects by the classifier for constraints induced from data (with accuracy usually less than 100%). One should take this issue into account in constructing nets of information systems.

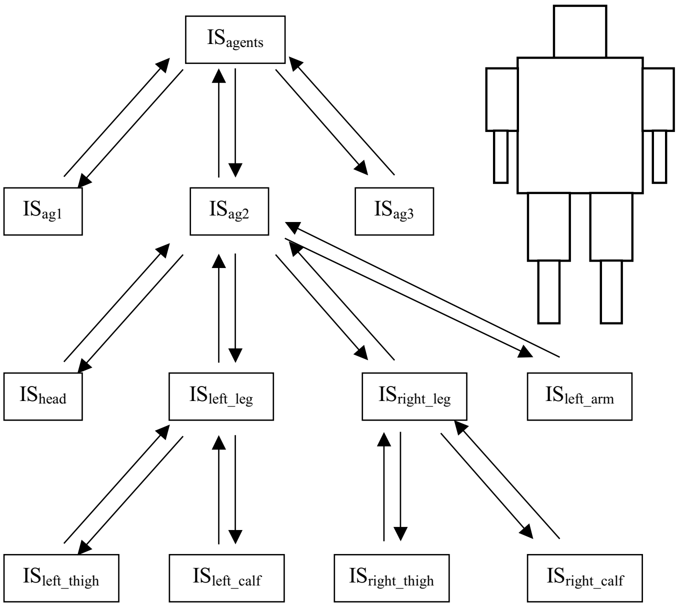

Figure 1. An example of information system hierarchy

4. Hierarchical Modelling

Problems of approximation of complex concepts create nowadays a challenge for science (see, e.g., [8], [19], [11]). For example, in identification of dangerous situations on the road by unmanned vehicle aircraft (UAV), the target concept (i.e., concept that we would like to approximate) is too complex to be directly approximated from feature value vectors (e.g., vector values of sensor measurements). One of the emerging approaches to deal with such cases is the hierarchical (or layered) learning [17], [1] approach to concept approximation.

Hierarchical modelling builds up complex information systems by initially starting with some basic simple information systems which are combined using appropriate constrained sums to define more complex information systems. In turn these more complex information systems can be combined (using appropriate constrained sums) to make still more complex information systems. An example of a tree like hierarchy of information systems (with infomorphisms) is presented in Figure 1.

We attempt to built a software system supporting the modelling process of hierarchical learning. This software will help to discover relevant patterns for complex concepts. The modelling process is based on constructing successively relevant constrained sums toward the complex concept approximation. It is important to note that in such hierarchical modelling we gradually construct (induce) new sets. In construction of a given target concept approximations on a higher level we use the already constructed approximations for simpler concepts and domain knowledge about the new target concept to find its approximations. Domain knowledge can be represented in different way, e.g., by means of decision tables describing on a sample of objects a relation of the concept with already defined concepts or by dependencies between concepts.

For structural objects we usually have more complex relational structures than those represented so far by information systems [5]. Starting from the basic level of hierarchical modelling we often have to deal with relations (on objects) of arity higher than one, together with unary predicates corresponding to descriptors widely used in information systems. For example, often the *to be a part to a degree* relation [11] or some time related relation is used. Approximations of concepts on this level are derived by means of neighbourhoods of objects defined by the uncertainty function and the rough inclusion [9], [11].

Hence, we propose the following definition of decision systems for structural objects.

Definition 4.1. Let $\mathcal{R}$ be a relational structure over a (finite) universe U and let $\mathcal{N}_{\mathcal{R}}$ be a family of neighbourhoods, i.e., relational structures that are substructures of restrictions of $\mathcal{R}$ to subsets of U. A decision system over the relational structure R and the family of neighbourhoods $\mathcal{N}_{\mathcal{R}}$ is any decision system $DT = (\mathcal{N}_{\mathcal{R}}, A, d)$ [5].

Let us consider an example of a neighbourhood family $\mathcal{N}_{\mathcal{R}}$. Assume $N(x) \subseteq U$ is selected for any object $x \in U_0 \subseteq U$ where U_0 is a finite sample of U. Then $\mathcal{N}_{\mathcal{R}}$ is equal to the set of all restrictions of $\mathcal{R}$ to $N(x)$ for $x \in U_0$. For real-life applications it is necessary to discover (from given data and domain knowledge) relevant relational structure $\mathcal{R}$, family of neighbourhoods $\mathcal{N}_{\mathcal{R}}$ as well as the set of conditional attributes A over such neighbourhoods.

Higher arity relations on objects can be often approximated from data. However, in some cases such relations are explicitly defined on basic objects (e.g., using a distance between objects) that can be indiscernible [5]. Then a granulation of these relations should be performed what leads to relations defined on neighbourhoods of objects rather than on objects [6].

For decision problems with complex structural objects one should consider hierarchical structures of information systems over different neighbourhood families representing parts of different relational structures. Any higher level of such a hierarchy is defined over the relational structures of the lower levels. The above definition of decision systems can be also used on higher levels of hierarchical modelling.

The relational structures constructed on the lower level of hierarchical construction are used to define new information systems on the next level of construction. Such information systems for more complex objects are defined by a composition of information systems from lower level of hierarchy representing parts of these more complex objects [12]. Each object on a higher level of hierarchical construction represents partial information about neighbourhoods (relational structures) of composed information systems, i.e., it consists of a subset of the universe together with object relationships defined by relations from the underlying relational structures. In specification of objects on higher levels some constraints between composed neighbourhoods from lower levels are also used. In this way neighbourhoods are generalisations of windows, widely used in temporal reasoning (e.g., time windows in time series analysis).

The neighbourhoods of objects from the universe on a higher level of hierarchy are constructed using the following information:

1. parts of the object structure represented by neighbourhoods on lower level of hierarchy (by applying some operations to them);
2. attributes (formulas) defined over the neighbourhoods constructed on the lower level of hierarchy;
3. formulas describing constraints between composed neighbourhoods from the lower level of hierarchy (that are also based on new conditional attributes for the higher level);
4. degrees to which (at least) the considered above formulas are satisfied.

In spatio-temporal reasoning we often have to deal with information systems called decision tables, i.e., information systems with a distinguished decision attribute [5]. The approximation of decision classes is expressed by conditional attributes of the decision system. The conditional attributes over neighbourhoods representing objects in such decision tables should be relevant for approximation of decision classes defined by the decision. For structural objects these conditional attributes are dependent on the neighbourhood structure. Important problems for spatio-temporal reasoning include the discovery of neighbourhoods and their properties relevant for decision classes approximation. For other applications, such as multi-criteria decision making, the relevant neighbourhoods are given and only their relevant properties should be discovered. The conditional attributes of decision systems on a higher level are defined over neighbourhoods available on that level. Such conditional attributes can also be defined by classifiers, in particular, rough-fuzzy classifiers [11].

The above described modelling process can be expressed by means of constrained sums of information systems. Constraints are expressed in some language that is interpreted in a set of tuples of composed information systems. Formulas expressing constraint for a given sum are built using relational symbols related to the components as well as some other relational symbols used to "filter" the relevant tuples for pattern modelling in concept approximations. Hence, constraints should define a type of tuples as well as their diversity.

In the following section we present examples of languages for constraint modelling.

5. Constraints in Hierarchical Modelling

In this section we discuss an important problem of searching for relevant constraints in a given language of constraints (called also the pattern language). Such relevant constraints make it possible to construct relevant patterns in constrained sums of information systems. We present examples of such languages. Observe that constraints are parameters of constrained sum. Hence, any pattern language describes a set of possible constrained sum operations. Searching for relevant constraints can be treated as a searching for relevant constrained sum operations.

Our general idea is based on the assumption that in searching for relevant patterns for target concepts the induction can be performed successfully only if the target concept is "not too far" from the already approximated concepts. The phrase "not far" means that one can expect to find the relevant patterns in the language used for approximation of the target concept from simpler concepts. In this way dependencies between "close" concepts can be modelled. Such dependencies are taken from domain knowledge. Hierarchical schemes of reasoning from domain knowledge can be used in searching for complex patterns relevant for complex target concepts that are "far from" the available basic concepts. The phrase "far from" means that one can hardly expect that such patterns can be induced directly from the patterns relevant for the basic concepts. In this way, using the domain knowledge, one can gradually construct relevant patterns for concepts that can lead finally to the patterns for the target complex concept.

5.1. Conjunctions of Descriptors and Conjunctions of Generalised Descriptors

The language of Boolean combination of descriptors and its semantics are defined in Section 2.2. Now we consider some sublanguages of this language.

The first example is a language $\mathcal{L}_{des}$ consisting expressions that are conjunction of descriptors over attributes from information systems from arguments of constrained sum. Any constraint from $\mathcal{L}_{des}$

defines a pattern for the constructed constrained sum of information system. Such pattern is relevant with respect to the target concept related to the constructed constrained sum if it is included to a satisfactory degree in such concept. The resulting constraint can be described as a disjunction of relevant patterns that make it possible to define the target concept approximation.

Searching for relevant patterns in $\mathcal{L}_{des}$ can be realised by generating decision rules from a decision table constructed in the following way (see also [11]):

1. Extract a sample consisting of a subtable of the sum of considered information systems together with a decision, given by expert, expressing if the object is matching the target concept.

2. Generate decision rules from such decision table using rough set methods (see, e.g., [20]). The left hand sides of decision rules for the decision corresponding to the target concept define relevant patterns, certainly one can consider approximate decision rules that are included into the target concept to a satisfactory degree. The degree can be expressed by confidence and support coefficients [11].

Patterns from the language $\mathcal{L}_{des}$ are the simplest ones. The generalisation is obtained by conjunction of only some of the descriptors from arguments of the constrained sum of information systems instead of the total their descriptions from argument information systems.

Now, we explain how the process of searching for relevant patterns over a more expressible language for constraints is possible. Instead of descriptors we consider so called generalised descriptors of the form $(a\ in\ V)$ where $V \subseteq V_a$. The problem of searching for relevant patterns is now related to searching for the relevant conjunctions of generalised descriptors over attributes from information systems composed by the constrained sum. Heuristic searching for such patterns have been proposed by several authors (see, e.g., [1]). One can apply them to samples of the sum of information systems completed to decision tables by adding the expert decision for the considered target concept.

5.2. Patterns Extracted from Classifiers

An interesting language that can be used in searching for relevant patterns is the language of patterns extracted from already induced classifiers on some training data sets for simpler concepts [1]. We present an example illustrating such an approach.

For a given a decision table $DT = (U, A, d)$ with $V_d = \{1, \dots, r\}$ [5] by $\mathbf{RULES}(DT)$ we denote a set of decision rules induced by some rule extraction method [20]. For any new object $x \in \mathcal{U}$, where $U \subseteq \mathcal{U}$ let $MatchRules(DT, x)$ be the set of rules from $\mathbf{RULES}(DT)$ supported by x. One can define the rough membership function $\mu_{CLASS_k} : \mathcal{U} \rightarrow [0, 1]$ for the concept determined by $CLASS_k = \{x \in U : d(x) = k\}$ (where $k = 1, \dots, r$) by

1. Let $\mathbf{R}_{yes}(x)$ be the set of all decision rules from $MatchRules(DT, x)$ for k^{th} class and let $\mathbf{R}_{no}(x) \subset MatchRules(DT, x)$ be the set of decision rules for other classes.

2. We define two real valued functions $w_{yes}(x), w_{no}(x)$, called “for” and “against” weight functions for the object x by

$$w_{yes}(x) = \sum_{\mathbf{r} \in \mathbf{R}_{yes}(x)} strength(\mathbf{r}), \qquad w_{no} = \sum_{\mathbf{r} \in \mathbf{R}_{no}(x)} strength(\mathbf{r}), \tag{3}$$

where $strength(\mathbf{r})$ is a normalised function depending on $length$, $support$, $confidence$ of **r** and some global information about the decision table DT like table size, class distribution.

3. One can define the value of $\mu^{\theta,\omega}_{CLASS_k}(x)$ by

$$\mu^{\theta,\omega}_{CLASS_k}(x) = \begin{cases} \text{undefined} & \text{if } \max(w_{yes}(x), w_{no}(x)) < \omega \\ 0 & \text{if } w_{no}(x) - w_{yes}(x) \geq \theta \text{ and } w_{no}(x) > \omega \\ 1 & \text{if } w_{yes}(x) - w_{no}(x) \geq \theta \text{ and } w_{yes}(x) > \omega \\ \frac{\theta + (w_{yes}(x) - w_{no}(x))}{2\theta} & \text{in other cases,} \end{cases}$$

where ω, θ are parameters set by user. These parameters make it possible in a flexible way to control the size of boundary region for the concept approximations.

Now, one can include into the language of patterns used to define constraints expressions interpreted as the above parameterised functions $\mu^{\theta,\omega}_{CLASS_k}$. Such expressions are induced for concepts corresponding to arguments of the constrained sum and their composition can be used in searching for relevant patterns for the target concept approximation. Certainly, one can construct such patterns using other weight functions or other strategies for synthesising classifiers such as k-nn (see, e.g., [1]). In this way it is possible to enrich the expressibility of the language. The relevant constraints for the target concept related to the constrained sum of information systems are next extracted from such a language.

5.3. Rough Patterns for Vague Concepts

Another language of patterns that can be used in searching for relevant constraints is the language of rough-fuzzy patterns for vague concepts.

Let us first discuss shortly an example of rough-fuzzy patterns.

Let $DT = (U, A, d)$ be a decision table with binary decision $d : U \longrightarrow \{0, 1\}$, i.e., d is the characteristic function of some $X \subseteq U$. If the decision table is inconsistent [5] then one can define a new decision deg such that $deg(x) \in [0, 1]$ for any $x \in U$ can be interpreted as a degree to which x belong to X [5], [11]. Let us consider such new decision table $DT' = (U, A, deg)$.

For given reals $0 < c_1 < \ldots < c_k$ where $c_i \in (0, 1]$ for $i = 1, \ldots, k$ we define c_i-cut by $X_i = \{x \in U : \nu(x) \geq c_i\}$ for $i = 1, \ldots, k$. Assume that $X_0 = U$ and $X_{k+1} = X_{k+2} = \emptyset$.

Any $B \subseteq A$ satisfying the following condition:

$$UPP(AS_B, (X_i - X_{i+1})) \subseteq (X_{i-1} - X_{i+2}), \text{ for } i = 1, \ldots, k, \tag{4}$$

is called relevant for approximation of cuts $0 < c_1 < \ldots < c_k$ in DT'.

The condition (4) expresses that fact that the boundary region of the set between any two successive cuts is included into the union of this set and two adjacent to it such sets.

The language $\mathcal{L}_{rf}$ of rough-fuzzy patterns for DT' consists of tuples $(B, c_1, \ldots, c_k)$ defining approximations of regions between cuts, i.e.,

$$(LOW(AS_B, (X_i - X_{i+1})), UPP(AS_B, (X_i - X_{i+1}))), \text{ for } i = 0, \ldots, k, \tag{5}$$

where we assume that B is relevant for approximation of cuts $0 < c_1 < \ldots < c_k$ in DT'.

Observe that searching for relevant patterns describing regions between cuts is related to tuning parameters $(B, c_1, \ldots, c_k)$ to obtain relevant patterns for the target concept approximation.

From concept description in DT' (on a sample U) of one can induce the concept approximation on an extension $U^* \supseteq U$. We consider, in a sense, richer classifiers, i.e., the classifiers that make it possible to predict different degrees to which the concept is satisfied. Such degrees can correspond to linguistic terms (e.g., low, medium, high) linearly ordered and to the boundary regions between successive degrees. Next we construct language of patterns from such classifiers analogously like in Section 5.2. Searching for patterns defining constraints aims at extracting patterns form such language corresponding to the arguments of the constrained sum that after composing create patterns include to the target concept to a satisfactory degree.

5.4. Modelling Clusters

In this subsection we discuss a modelling of clusters.

Let $IS = (U, A)$ be an information system and let $Sim \subseteq U \times U$ be a similarity relation. We assume that Sim is a reflexive and symmetric relation, i.e.,

- $(x, x) \in Sim$ for any $x \in U$,
- if $(x, y) \in Sim$ then $(y, x) \in Sim$ for any $x, y \in U$.

For every object $x \in U$ we define a cluster $\{y \in U : (x, y) \in Sim\}$ of objects.

Let us define a new information system $IS_* = (U \cup \{*\}, A_*)$ where for every attribute $a_* \in A_*$ and for every object $x \in U \cup \{*\}$ we use the following rules

If $x \neq *$, then $a_*(x) = a(x)$,

If $x = *$, then $a_*(x) = new$, where $new \notin V_a$.

Let us assume that $U = \{x_1, \ldots, x_n\}$ where $n > 0$ is a given natural number. We define a constrained relation $R \subseteq (U \cup \{*\})^n$ by

$(y_1, \ldots, y_n) \in R$ if and only if $\exists y_i \forall y_j (y_j \neq * \leftrightarrow (y_i, y_j) \in Sim)$.

The constrained sum $+_R(IS_*, \ldots, IS_*)$ of n copies of IS_* represents a set of clusters.

Hence, the objects in the constrained sum can be interpreted as similarity classes of objects ($*$ on a ith position in the sequence represents that x_i is not in the similarity class).

Example 5.1. Let $U = \{x_1, \ldots, x_7\}$ be a set of objects. A new object $(x_1, x_2, *, *, x_5, *, x_7)$ in the constructed sum $+_R(IS_*, IS_*, IS_*, IS_*, IS_*, IS_*, IS_*)$ represents a similarity class $\{x : (x_1, x) \in Sim\} = \{x_1, x_2, x_5, x_7\}$.

Observe that also new attributes can be represented using constrained sums of information systems. Let us consider a simple illustrative example.

Let t be a given real number from $[0, 1]$. We can construct an attribute a^t_{avg} such that the domain $V_{a^t_{avg}} = \{0, 1\}$. We assume that for any sequence $(x_1, \ldots, x_n)$ of objects from the universe of constrained sum

$$a^t_{avg}((x_1, \ldots, x_n)) = 1 \text{ holds}$$

if and only if

$$\sum_{\{i : a_*(x_i) \neq \, new\}} a_*(x_i) > card(\{i \in \{1, \ldots, n\} : x_i \neq *\}) \cdot t.$$

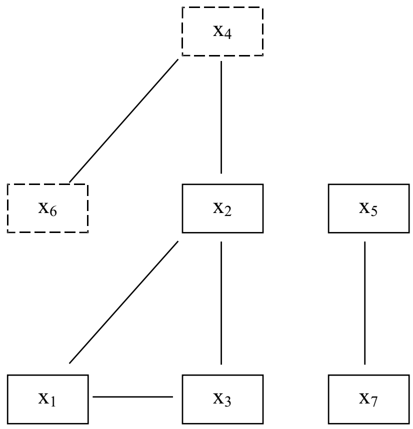

Figure 2. Overlapping clusters and a concept

The above considerations can be summarised as follows. Constraints used to define constrained sums of information systems specify a type of new composed objects as well as their admissible properties.

5.5. Clusters Included in Concept

Let $U = \{x_1, \ldots, x_n\}$ be a set of objects where $n > 0$ is a given natural number and let C be a concept which we understand as a subset of U. Let Sim be a reflexive and symmetric relation in U. We denote $Sim(y) = \{x \in U : (x, y) \in Sim\}$. An example for $n = 7$, $C = \{x_1, x_2, x_3, x_5, x_7\}$ and Sim depicted by edges of the graph is presented in Figure 2.

We define new clusters $(z_1, \ldots, z_n)$ such that the following three conditions are satisfied

- $z_i \in \{Sim(x_i), \emptyset\}$
- $card(X \cap C)/card(X) > tr$ where $X = \bigcup_{z_i \neq \emptyset} z_i$ and $tr \in (0, 1]$ is a threshold
- X is maximal, this means that for $X \cup \{Sim(x_j) : z_j = \emptyset\}$ the above condition is not true.

Example 5.2. Let $U = \{x_1, \ldots, x_7\}$ be a set of objects and let $C = \{x_1, x_2, x_3, x_5, x_7\}$. Let a threshold $tr = 0.8$ and $Sim(x_1) = Sim(x_3) = \{x_1, x_2, x_3\}$, $Sim(x_2) = \{x_1, x_2, x_3, x_4\}$, $Sim(x_4) = \{x_2, x_4, x_6\}$, $Sim(x_6) = \{x_4, x_6\}$ and $Sim(x_5) = Sim(x_7) = \{x_5, x_7\}$ (see Figure 2). A new object

$$(Sim(x_1), Sim(x_2), Sim(x_3), \emptyset, Sim(x_5), \emptyset, Sim(x_7))$$

represents the following property

”$X = Sim(x_1) \cup Sim(x_2) \cup Sim(x_3) \cup Sim(x_5) \cup Sim(x_7)$ is sufficiently included in C and $X \cup Sim(x_i)$ is not sufficiently included in C for $i = 4, 6$”

5.6. Relational Information Systems

In this subsection we give examples of information systems with relational structures on the sets of attribute values. One can define a constraint relation for constrained sum based on relations defined in relational structures.

Definition 5.1. A relational information system RIS is a triple $RIS = (U, A, \{STR_a\}_{a \in A})$ where U is a set of objects, A is a set of attributes and for any $a \in A$ $STR_a = (V_a, \ldots)$ is a relational structure on a set V_a of values of a.

Example 5.3. Let us consider first two simple examples.

For any attribute $a \in A$ we consider a relational structure $STR_a = (V_a, =_a)$ where $=_a$ is an equality relation on the set of values V_a. We obtain an indiscernibility relation $IND_A = \{(x, y) \in U \times U :$ for any $a \in A$ $a(x) =_a a(y)\}$.

Let for any attribute $a \in A$ we consider a relational structure $STR_a = (V_a, \leq_a)$ where $\leq_a$ is a linear order on the set of values V_a. This is a typical case in rough set analysis of preference-ordered data [15] and information systems with time [13].

From relational systems with time one can construct information systems with objects representing history of attribute value vectors in a given time window [13].

Example 5.4. Let us consider for $i = 1, 2$ an information system IS_i with objects represented by functions assigning an attribute value vector from IS_i to each point in a time window of size T. Now, we define a constrained sum $+_R(IS_1, IS_2)$ by $(\overline{v}_1, \ldots, \overline{v}_T)R(\overline{u}_1, \ldots, \overline{u}_T)$ if and only if $\overline{v}_T = \overline{u}_T$ where $(\overline{v}_1, \ldots, \overline{v}_T)$ and $(\overline{u}_1, \ldots, \overline{u}_T)$ are objects of IS_1, IS_2, respectively. Hence, $+_R(IS_1, IS_2)$ consists of pairs of objects such that the second is the continuation in time of the first.

One can consider another constrained sum $+_{G_{\alpha,\beta}}(IS_1, IS_2)$ of information systems IS_1, IS_2 with the constraint relation $G_{\alpha,\beta}$ defined by

$$(x, y) \in G_{\alpha,\beta} \text{ if and only if } x \textit{ satisfies } \alpha \textit{ and } y \textit{ satisfies } \beta$$

where α is a conjunction of the following conditions:

1. α_1: *car* A *was accelerating in time* $[1, T/4)$ *on the right lane*;
2. α_2: *car* A *was changing the right lane to the left lane in time* $[T/5, T/3)$;
3. α_3: *car* A *was driving on the left lane in time* $[T/3, 2T/3)$;
4. α_4: *car* A *was changing the lane to the right lane in time* $[2T/3, T]$;
5. α_5: *car* A *was not changing the speed in time* $[T/4, T]$;

and β denotes the condition *car* B *was not changing the speed in time* $[1, T]$ (see Figure 3).

In (6) we assume that a satisfiability relation for considered formulas is fixed. Moreover, the intuitive meaning of formulas mentioned above should be expressed in a relevant language.

Hence, objects in $+_{G_{\alpha,\beta}}(IS_1, IS_2)$ are pairs representing behaviour patterns in a given time window of cars A and B related to the standard overtaking pattern. Next, one can search in the set of objects of $+_{G_{\alpha,\beta}}(IS_1, IS_2)$ for relevant patterns (in approximating the target concept) using some additional spatio-temporal constraints making it possible to check if, e.g., a dangerous situation was created by cars A and B in the considered time window. In this way the patterns can be tuned in the searching process for relevant patterns.

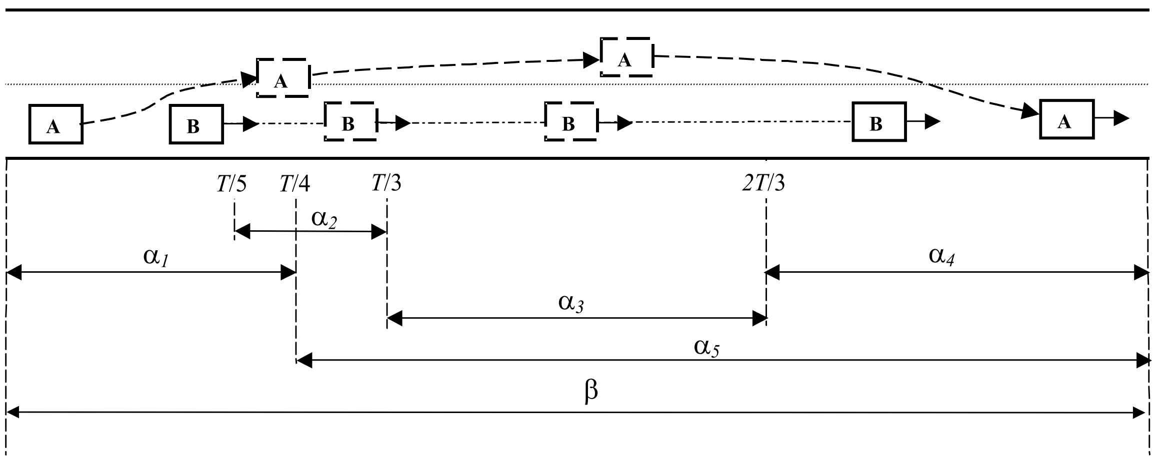

Figure 3. Illustration for overtaking pattern

Conclusions

We have outlined an approach based on constrained sum operations for hierarchical modelling of patterns. The approach is used in a software system that we are developing for construction of patterns relevant for approximation of complex spatio-temporal concepts. Using domain knowledge one can construct networks of parameterised constrained sums (i.e., in such networks each internal node is labelled by a set of constraints rather then by one constraint). Relevant patterns over such networks are discovered by strategies over such networks searching for optimal parameters.

References

[1] Bazan, J., Nguyen, S.H., Nguyen, H.S., Skowron, A.: Rough set methods in approximation of hierarchical concepts. In: Fourth International Conference on Rough Sets and Current Trends in Computing (RSCTC'04), Uppsala, Sweden, June 1-5, 2004. Lecture Notes in Computer Science, Springer-Verlag 3066, Heidelberg, 2004, 346-355.

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

[3] Garcia-Molina, H., Ullman, J.D., Widom, J.D.: Database Systems: The Complete Book, Prentice Hall, Upper Saddle River, New Jersey, 2002.

[4] Łukasiewicz, J.: Die Logischen grundlagen der Wahrscheinlichkeitsrechnung, Kraków 1913. In Borkowski, L., ed.: Jan Łukasiewicz - Selected Works. North Holland Publishing Company, Amstardam, London, Polish Scientific Publishers, Warsaw, 1970.

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

[6] Peters, J. F., Skowron, A., Synak, P., Ramanna, S.: Rough Sets and Information Granulation, *Tenth International Fuzzy Systems Association World Congress IFSA* (T. Bilgic, D. Baets, O. Kaynak, Eds.), *Lecture Notes in Artificial Intelligence*, **2715**, Springer-Verlag, Heidelberg, 2003, 370–377.

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

[8] Poggio, T., Smale, S.: The Mathematics of Learning: Dealing with Data, *Notices of the AMS*, **50**(5), 2003, 537–544.

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

[10] Skowron, A., Stepaniuk, J.: Information Granules: Towards Foundations of Granular Computing, International Journal of Intelligent Systems **16**(1), 2001, 57–86.

[11] Skowron, A., Stepaniuk, J.: Information Granules and Rough-Neuro Computing. in [7], 43–84.

[12] Skowron, A., Stepaniuk, J., Peters, J.F.: Rough Sets and Infomorphisms: Towards Approximation of Relations in Distributed Environments, Fundamenta Informaticae, **54**(1-2), 2003, 263–277.

[13] Skowron, A., Synak, P., Peters, J.F.: Spatio-Temporal Approximate Reasoning over Hierarchical Information Maps, Fundamenta Informaticae, this valume, 2005.

[14] Skowron, A., Swiniarski, R., Synak, P.: Approximation Spaces and Information Granulation. In: Fourth International Conference on Rough Sets and Current Trends in Computing (RSCTC'04), Uppsala, Sweden, June 1-5, 2004. Lecture Notes in Computer Science 3066, Springer-Verlag, Heidelberg, 2004, 116-126.

[15] Słowiński, R., Greco, S., Matarazzo, B.: Rough Set Analysis of Preference-Ordered Data, Lecture Notes in Artificial Intelligence 2475, Springer-Verlag, Heidelberg, Germany, 2002, 44–59.

[16] Stepaniuk, J.: Knowledge Discovery by Application of Rough Set Models, (Eds.) L. Polkowski, S. Tsumoto, T.Y. Lin, Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems, Physica–Verlag, Heidelberg, 2000, 137–233.

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

[18] Zadeh, L.A.: Toward a theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic. *Fuzzy Sets and Systems* **90** (1997) 111–127.

[19] Zadeh, L.A.: A new direction in AI: Toward a computational theory of perceptions. *AI Magazine* **22**(1), 2001, 73–84.

[20] The RSES Homepage – `logic.mimuw.edu.pl/∼rses`