Journal of Experimental & Theoretical Artificial Intelligence,
Vol. 17, No. 1–2, January–June 2005, 83–102

Hierarchical modelling in searching for complex patterns: constrained sums of information systems

ANDRZEJ SKOWRON†* and JAROSŁAW STEPANIUK‡

† Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
‡Department of Computer Science, Białystok University of Technology, Wiejska 45a, 15-351 Białystok, Poland

(*Received 15 April 2004; in final form 10 August 2004*)

This paper outlines an approach to hierarchical modelling of complex patterns that is based on operations of sums with constraints on information systems. It is shown that such operations can be treated as a universal tool in hierarchical modelling of complex patterns.

Keywords: Rough sets; Information granules; Concept approximation; Hierarchichal modelling; Infomorphisms; Information nets

1. Introduction

One of the main tasks in granular computing is to develop calculi of information granules (Zadeh 1997, Skowron and Stepaniuk 2001). 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 (Barwise and Seligman 1997). In Skowron *et al*. (2003), we have shown that classifications (Barwise and Seligman 1997) and information systems (Pawlak 1991) are, in a sense, equivalent. We also extend the results included in Skowron *et al*. (2003) 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) (Kloesgen and Zytkow 2002) or in a more general sense in generating relevant granules for approximate reasoning using calculi on information granules (Polkowski and Skowron 1999, Skowron *et al*. 2003).

This paper is organized as follows. In section 2, we present basic concepts. In section 3, we introduce sums of information systems and approximation spaces. In section 4, we discuss constrained sums of information systems and hierarchical

*Corresponding author. Email: skowron@mimuw.edu.pl

Journal of Experimental & Theoretical Artificial Intelligence
ISSN 0952–813X print/ISSN 1362–3079 online © 2005 Taylor & Francis Ltd
http://www.tandf.co.uk/journals
DOI: 10.1080/09528130512331315873

information systems. Applications of constrained sums of information systems in modelling patterns for concept approximation is outlined in sections 5 and 6. In section 7, we show how rough set approximations of primitive concepts linked by infonets can be used in construction of approximations of more complex concepts.

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 or the variable precision rough set model.

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

Definition 1 (*Skowron and Stepaniuk 1996, Stepaniuk 2000*)**.** *A parameterized 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 can define the degree of inclusion between two subsets of U (see, for example, Skowron and Stepaniuk 1996, Stepaniuk 2000):

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

This measure is widely used by data mining and rough set communities. However, Łukasiewicz (1913) was first who used this idea to estimate the probability of implications.

For example, any information system $IS = (U, A)$ defines an approximation space $AS_A = (U, I_A, \nu_{SRI})$ where $I_A(x)$ is the A-indiscernibility class (Pawlak 1991) defined by x.

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

Definition 2 *For an approximation space* $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$ *and any subset* $X \subseteq U$ *the lower and the upper approximations are defined by*

$LOW(AS_{\#,\$}, X) = \{x \in U\colon \nu_{\$}(I_{\#}(x), X) = 1\}$,
$UPP(AS_{\#,\$}, X) = \{x \in U\colon \nu_{\$}(I_{\#}(x), X) > 0\}$, *respectively.*

2.2 *Infomorphisms*

In this section, we recall the definition of infomorphism for two information systems (Skowron *et al.* 2003). We also present some new properties of infomorphisms. The infomorphisms for classifications are introduced and studied in Barwise and Seligman (1997).

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. $(a \text{ in } V) \in \Sigma(IS)$, 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. $\|a \text{ in } V\|_{IS} = \{x \in U\colon a(x) \in V\}$.
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 3 (Barwise and Seligman 1997, Skowron *et al.* 2003). *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 $f^\wedge : \Sigma(IS_1) \to \Sigma(IS_2)$, $f^\vee : U_2 \to U_1$, satisfying the following equivalence*

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

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

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

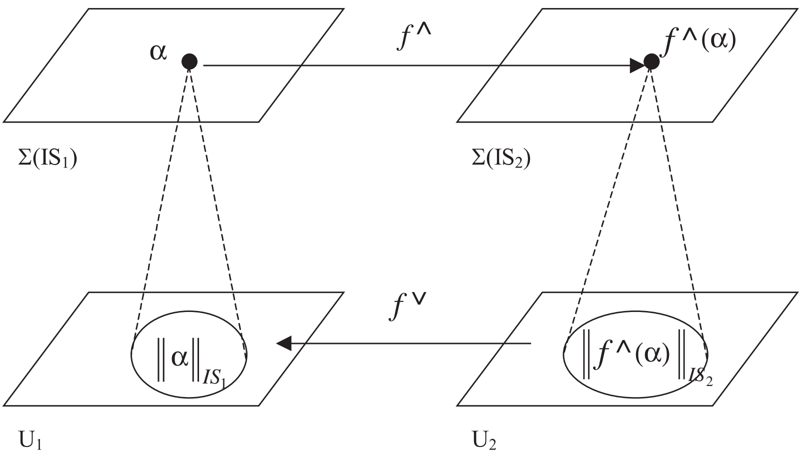

Figure 1. Infomorphism illustration.

Example 4 *General scheme of infomorphism is depicted in figure 1, where*

- $IS_1 = (U_1, A_1)$ *and* $IS_2 = (U_2, A_2)$ *are information systems*;
- U_1 *and* U_2 *are sets of objects*;
- A_1 *and* A_2 *are sets of attributes*;
- $\Sigma(IS_1)$ *and* $\Sigma(IS_2)$ *are sets of formulas over sets* A_1 *and* A_2 *of attributes, respectively*;
- $f^\wedge$ *is a function such that* $f^\wedge : \Sigma(IS_1) \rightarrow \Sigma(IS_2)$;
- $f^\vee : U_2 \rightarrow U_1$;
- α *is a formula such that* $\alpha \in \Sigma(IS_1)$ *and* $f^\wedge(\alpha) \in \Sigma(IS_2)$;
- $\|\alpha\|_{IS_1} = \{x \in U_1 : x \models_{IS_1} \alpha\}$; *and*
- $\|f^\wedge(\alpha)\|_{IS_2} = \{x \in U_2 : x \models_{IS_2} f^\wedge(\alpha)\}$.

Proposition 5 (*Skowron et al. 2003*). *For any infomorphism* $(f^\wedge, f^\vee) : IS_1 \rightleftarrows IS_2$ *we obtain the following equality*

$$(f^\vee)^{-1}(\|\alpha\|_{IS_1}) = \|f^\wedge(\alpha)\|_{IS_2} \quad \text{for any } \alpha \in \Sigma(IS_1) \tag{2}$$

Definition 6 *Let* $(f^\wedge, f^\vee)$ *be an infomorphism between* IS_1 *and* IS_2. *We define two binary relations* $\sim_{f^\wedge} \subseteq \Sigma(IS_1) \times \Sigma(IS_1)$ *and* $\approx_{f^\vee} \subseteq U_2 \times U_2$ *as follows*

1. $\alpha \sim_{f^\wedge} \beta$ *if and only if* $f^\wedge(\alpha) = f^\wedge(\beta)$ *for any* $\alpha, \beta \in \Sigma(IS_1)$,
2. $x \approx_{f^\vee} y$ *if and only if* $f^\vee(x) = f^\vee(y)$ *for any* $x, y \in U_2$.

We obtain the following proposition:

Proposition 7 *For any infomorphism* $(f^\wedge, f^\vee) : IS_1 \rightleftarrows IS_2$ *between* IS_1 *and* IS_2 *the following properties hold*:

1. *The relations* $\sim_{f^\wedge}$ *and* $\approx_{f^\vee}$ *are equivalence relations*,
2. $\alpha \sim_{f^\wedge} \beta$ *if and only if*
$$\|\alpha\|_{IS_1} \cap f^\vee(U_2) = \|\beta\|_{IS_1} \cap f^\vee(U_2),$$
for any $\alpha, \beta \in \Sigma(IS_1)$,
3. $x \approx_{f^\vee} y$ *if and only if*
$$(x \in \|f^\wedge(\alpha)\|_{IS_2} \quad \text{if and only if } y \in \|f^\wedge(\alpha)\|_{IS_2}) \quad \text{for any } \alpha \in \Sigma(IS_1),$$
where $x, y \in U_2$,
4. *either* $[x]_{\approx_{f^\vee}} \subseteq \|f^\wedge(\alpha)\|_{IS_2}$ *or* $[x]_{\approx_{f^\vee}} \cap \|f^\wedge(\alpha)\|_{IS_2} = \emptyset$ *for any* $\alpha \in \Sigma(IS_1)$ *and* $x \in U_2$,
5. *any formula* $\alpha \in f^\wedge(\Sigma(IS_1))$ *is definable in* $U_2/\approx_{f^\vee}$, i.e., $\|\alpha\|_{IS_2}$ *is a union of some equivalence classes from* $U_2/\approx_{f^\vee}$.

Let us recall that formulas from $\Sigma(IS_2) - f^\wedge(\Sigma(IS_1))$ can be defined approximately in $U_2/\approx_{f^\vee}$ (see Skowron *et al.* 2003).

Proposition 7 gives a characterization of infomorphisms.

Definition 8 *Let* $(f^\wedge, f^\vee) : IS_1 \rightleftarrows IS_2$ *be an infomorphism between* IS_1 *and* IS_2. *We define two information systems*:

$$IS'_1 = (f^\vee(U_2), \Sigma(IS'_1)) \quad \text{and} \quad IS'_2 = (U'_2, \Sigma(IS'_2)),$$

where

- $\Sigma(IS'_1)$ *is a subset of* $\Sigma(IS_1)$ *consisting of exactly one element from each equivalence class from* $\Sigma(IS_1)/\sim f^\wedge$ *and*

- *U'_2 is a subset of U_2 consisting of exactly one element from each equivalence class from $U_2/\approx_{f^\vee}$ and $\Sigma(IS'_2) = f^\wedge(\Sigma(IS_1))$.*

Proposition 9 *Let $(f^\wedge, f^\vee) : IS_1 \rightleftarrows IS_2$ be an infomorphism between IS_1 and IS_2. Then $(g^\wedge, g^\vee) : IS'_1 \rightleftarrows IS'_2$ is an infomorphism where $(g^\wedge, g^\vee)$ is a pair of bijections defined by $g^\wedge(\alpha) = f^\wedge(\alpha)$ and $g^\vee(x) = f^\vee(x)$ for any $\alpha \in \Sigma(IS'_1)$ and any $x \in U'_2$.*

In Proposition 9, we assume that $\|\alpha\|_{IS'_2} = \|\alpha\|_{IS_2} \cap U'_2$ for $\alpha \in \Sigma(IS'_2)$. This proposition expresses that on domains accessible in communication (between two agents represented by information systems) established by a given infomorphism, the infomorphism is defined by selection functions on equivalence classes of formulas and objects, respectively. Such functions are bijections. From this fact it follows that, roughly speaking, infomorphisms of information systems can be realized by operations that we call constrained sums. The details are presented in the following sections.

However, observe that the communication established by infomorpshims do not assure the complete knowledge between communicating agents (information systems). In particular, formulas (concepts) from $\Sigma(IS_2) - f^\wedge(\Sigma(IS_1))$ are not in general definable in IS_1, only their approximations are known for IS_1 (Skowron *et al.* 2003).

3. Sum of information systems and approximation spaces

3.1 *Sum of information systems*

In this section, we introduce a sum of two information systems.

Definition 10 *Let $IS_1 = (U_1, A_1)$ and $IS_2 = (U_2, A_2)$ be information systems. These information systems can be combined into a single information system, denoted by $+(IS_1, IS_2)$, with the following properties:*

- *The objects of $+(IS_1, IS_2)$ consist of pairs (x_1, x_2) of objects from IS_1 and IS_2 i.e. $U = U_1 \times U_2$.*
- *The attributes of $+(IS_1, IS_2)$ consist of the attributes of IS_1-copy and IS_2-copy, with disjoint attribute sets.*

Proposition 11 *There are infomorphisms $(f_k^\wedge, f_k^\vee) : IS_k \rightleftarrows +(IS_1, IS_2)$ for $k = 1, 2$ defined as follows:*

- *$f_k^\wedge(\alpha) = \alpha_{IS_k}$ (the IS_k-copy of α) for each $\alpha \in \Sigma(IS_k)$.*
- *for each pair $(x_1, x_2) \in U$, $f_k^\vee(x_1, x_2) = x_k$.*

Given any information system IS_3 and infomorphisms $(f_{k,3}^\wedge, f_{k,3}^\vee) : IS_k \rightleftarrows IS_3$, for $k = 1, 2$ there is a unique infomorphism $(f_{1+2,3}^\wedge, f_{1+2,3}^\vee) : +(IS_1, IS_2) \rightleftarrows IS_3$ such that the diagram in figure 2 commutes.

Example 12 *Let us consider a diagnostic agent testing failures of the space robotic arm. Such an agent should observe the arm and detect a failure if, for example, some of its parts are in abnormal relative position. Assume, in our simple example that projections of some parts on a plane are observed and a failure is detected if projection of some parts that are triangles or rectangles are in some relation, for example, the triangle is not included sufficiently inside of the rectangle. Hence, any considered object*

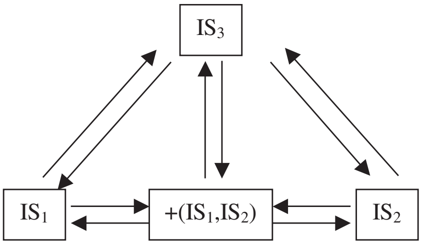

Figure 2. Sum of information systems IS_1 and IS_2.

Table 1. Information system $IS_{rectangle}$ with uncertainty functions.

$U_{rectangle}$	a	b	$I_a(\cdot)$	$I_b(\cdot)$	$I_{A_1}(\cdot)$
x_1	165	Yes	$\{x_1, x_3, x_5, x_6\}$	$\{x_1, x_3\}$	$\{x_1, x_3\}$
x_2	175	No	$\{x_2, x_4, x_6\}$	$\{x_2, x_4, x_5, x_6\}$	$\{x_2, x_4, x_6\}$
x_3	160	Yes	$\{x_1, x_3, x_5\}$	$\{x_1, x_3\}$	$\{x_1, x_3\}$
x_4	180	No	$\{x_2, x_4\}$	$\{x_2, x_4, x_5, x_6\}$	$\{x_2, x_4\}$
x_5	160	No	$\{x_1, x_3, x_5\}$	$\{x_2, x_4, x_5, x_6\}$	$\{x_5\}$
x_6	170	No	$\{x_1, x_2, x_6\}$	$\{x_2, x_4, x_5, x_6\}$	$\{x_2, x_6\}$

Table 2. Information system $IS_{triangle}$ with uncertainty function I_{A_2}.

$U_{triangle}$	c	$I_{A_2}(\cdot)$
y_1	t_1	$\{y_1, y_3\}$
y_2	t_2	$\{y_2\}$
y_3	t_1	$\{y_1, y_3\}$

consists of parts: a triangle and a rectangle. Objects are perceived by some attributes expressing properties of parts and a relation (*constraint*) *between them. In this section, we omit considering constraints in sums of information systems.*

First, *we construct an information system*, *called the sum of given information systems. Such system represents objects composed from parts without any constraint. It means that we consider as the universe of objects the Cartesian product of the universes of parts* (*table 1, columns 1–3; table 2, columns 1–2; table 3, columns 1–4*). *Let us consider three information systems* $IS_{rectangle} = (U_{rectangle}, A_{rectangle})$, $IS_{triangle} = (U_{triangle}, A_{triangle})$, *and* $+(IS_{rectangle}, IS_{triangle}) = (U_{rectangle} \times U_{triangle}, \{(a, 1), (b, 1), (c, 2)\})$ *presented in tables 1–3, respectively. Let* $U_{rectangle}$ *be a set of rectangles and* $A_{rectangle} = \{a, b\}$, $V_a = [0, 300]$ *and* $V_b = \{\text{yes}, \text{no}\}$, *where the value of a means a length in millimetres of horizontal side of rectangle and for any object* $x \in U_{rectangle}$ $b(x) = \text{yes}$ *if and only if x is a square.*

Let $U_{triangle}$ *be a set of triangles and* $A_{triangle} = \{c\}$ *and* $V_c = \{t_1, t_2\}$, *where* $c(x) = t_1$ *if and only if x is an acute-angled triangle and* $c(x) = t_2$ *if and only if x is a right-angled triangle.*

We assume all values of attributes are obtained from a given projection plane. The results of measurements are represented in information systems. Tables 1–2 include only illustrative examples of the results of such measurements.

Table 3. An information system $+(IS_{rectangle}, IS_{triangle})$ with uncertainty function I_{A_1,A_2}.

$U_{rectangle} \times U_{triangle}$	$(a,1)$	$(b,1)$	$(c,2)$	$I_{A_1,A_2}((\cdot,\cdot))$
(x_1,y_1)	165	Yes	t_1	$\{x_1,x_3\}\times\{y_1,y_3\}$
(x_1,y_2)	165	Yes	t_2	$\{x_1,x_3\}\times\{y_2\}$
(x_1,y_3)	165	Yes	t_1	$\{x_1,x_3\}\times\{y_1,y_3\}$
(x_2,y_1)	175	No	t_1	$\{x_2,x_4,x_6\}\times\{y_1,y_3\}$
(x_2,y_2)	175	No	t_2	$\{x_2,x_4,x_6\}\times\{y_2\}$
(x_2,y_3)	175	No	t_1	$\{x_2,x_4,x_6\}\times\{y_1,y_3\}$
(x_3,y_1)	160	Yes	t_1	$\{x_1,x_3\}\times\{y_1,y_3\}$
(x_3,y_2)	160	Yes	t_2	$\{x_1,x_3\}\times\{y_2\}$
(x_3,y_3)	160	Yes	t_1	$\{x_1,x_3\}\times\{y_1,y_3\}$
(x_4,y_1)	180	No	t_1	$\{x_2,x_4\}\times\{y_1,y_3\}$
(x_4,y_2)	180	No	t_2	$\{x_2,x_4\}\times\{y_2\}$
(x_4,y_3)	180	No	t_1	$\{x_2,x_4\}\times\{y_1,y_3\}$
(x_5,y_1)	160	No	t_1	$\{x_5\}\times\{y_1,y_3\}$
(x_5,y_2)	160	No	t_2	$\{x_5\}\times\{y_2\}$
(x_5,y_3)	160	No	t_1	$\{x_5\}\times\{y_1,y_3\}$
(x_6,y_1)	170	No	t_1	$\{x_2,x_6\}\times\{y_1,y_3\}$
(x_6,y_2)	170	No	t_2	$\{x_2,x_6\}\times\{y_2\}$
(x_6,y_3)	170	No	t_1	$\{x_2,x_6\}\times\{y_1,y_3\}$

We assume that $(a,1)((x_i,y_j)) = a(x_i)$, $(b,1)((x_i,y_j)) = b(x_i)$ *and* $(c,2)((x_i,y_j)) = c(y_j)$, *where* $i = 1,\ldots,6$ *and* $j = 1,2$.

3.2 *Sum of approximation spaces*

In this section, we present a simple construction of approximation space for the sum of given approximation spaces.

Let $AS_{\#_k} = (U_k, I_{\#_k}, \nu_{SRI})$ be an approximation space for information system IS_k, where $k = 1,2$. We define an approximation space $+(AS_{\#_1}, AS_{\#_2})$ for information system $+(IS_1, IS_2)$ as follows:

1. the universe is equal to $U_1 \times U_2$,
2. $I_{\#_1,\#_2}((x_1,x_2)) = I_{\#_1}(x_1) \times I_{\#_2}(x_2)$,
3. the inclusion relation ν_{SRI} in $+(AS_{\#_1}, AS_{\#_2})$ is the standard inclusion function.

Proposition 13 *Let* $X \subseteq U_1$ *and* $Y \subseteq U_2$. *We have the following properties of approximations*:

$$LOW(+(AS_{\#_1}, AS_{\#_2}), X \times Y) = LOW(AS_{\#_1}, X) \times LOW(AS_{\#_2}, Y) \qquad (3)$$

$$UPP(+(AS_{\#_1}, AS_{\#_2}), X \times Y) = UPP(AS_{\#_1}, X) \times UPP(AS_{\#_2}, Y) \qquad (4)$$

Proof We have $I_{\#_1,\#_2}((x_1,x_2)) \subseteq X \times Y$ iff $I_{\#_1}(x_1) \subseteq X$ and $I_{\#_2}(x_2) \subseteq Y$. Moreover, $I_{\#_1,\#_2}((x_1,x_2)) \cap (X \times Y) \neq \emptyset$ iff $I_{\#_1}(x_1) \cap X \neq \emptyset$ and $I_{\#_2}(x_2) \cap Y \neq \emptyset$.

Example 14 *Let* $A_1 = \{a,b\}$, $A_2 = \{c\}$ *(see Example 12). For information system* $IS_{rectangle}$, *we define an approximation space* $AS_{A_1} = (U_{rectangle}, I_{A_1}, \nu_{SRI})$ *such that* $y \in I_a^5(x)$ *if and only if* $|a(x) - a(y)| \leq 5$. *This means that rectangles x and y are similar with respect to the length of horizontal sides if and only if the difference of lengths is not greater than 5 millimetres. Let* $y \in I_b(x)$ *if and only if* $b(x) = \mathrm{b}(y)$ *and* $y \in I_{A_1}(x)$ *if and* *only if* $\forall_{c \in A_1} y \in I_c(x)$. *Thus, we obtain uncertainty functions represented in the last three columns of table 1. For information system* $IS_{triangle}$ *we define an approximation space as follows:* $y \in I_{A_2}(x)$ *if and only if* $c(x) = c(y)$ (*see the last column of table 2*). *For* $+(IS_{rectangle}, IS_{triangle})$ *we obtain* $I_{A_1,A_2}((x, y)) = I_{A_1}(x) \times I_{A_2}(y)$ (*see the last column of table 3*).

4. Constrained sums

In this section, we consider operations on information systems that can be used in searching for hierarchical patterns. The operations are parameterized 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.

4.1 *Constrained sums of information systems*

In this section, we consider a new operation on information systems often used in searching, e.g. for relevant patterns. This operation is more general than theta join operation used in databases (Garcia-Molina *et al.* 2002). We start from the definition in which the constraints are given explicitly.

Definition 15 *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* $U_1 \times \cdots \times U_k$, i.e. $R \subseteq U_1 \times \cdots \times U_k$. *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* $U_1 \times \cdots \times U_k$ *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, to not to confuse them.*

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. In our example (see table 4), the constraint R_1 is defined as follows: *the triangle is sufficiently included in the rectangle*. Any row of this table represents an object (x_i, y_j) composed of the triangle y_j included sufficiently into the rectangle x_i.

Let us also note that constraints are defined using primitive (measurable) attributes different than those from information systems describing parts. This makes the constrained sum different from the theta join operation (see, for example, Garcia-Molina *et al.* 2002). On the other hand, one can consider that the constraints are defined in two steps. In the first step, we extend the attributes for parts and in the second step we define the constraints using some relations on these new attributes.

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

Table 4. Information system $+_{R_1}(IS_{rectangle}, IS_{triangle})$.

$(U_{rectangle} \times U_{triangle}) \cap R_1$	$(a,1)$	$(b,1)$	$(c,2)$
(x_1, y_1)	165	Yes	t_1
(x_1, y_2)	165	Yes	t_2
(x_2, y_1)	175	No	t_1
(x_2, y_2)	175	No	t_2
(x_3, y_1)	160	Yes	t_1
(x_3, y_2)	160	Yes	t_2
(x_4, y_1)	180	No	t_1
(x_4, y_2)	180	No	t_2
(x_5, y_1)	160	No	t_1
(x_5, y_2)	160	No	t_2
(x_6, y_1)	170	No	t_1
(x_6, y_2)	170	No	t_2

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 decision table one can induce classifier for the constraint relation. Next, such classifier can be used to select tuples in the construction of constrained sum.

Example 16 *Let us consider three information systems* $IS_{rectangle} = (U_{rectangle}, A_{rectangle})$*,* $IS_{triangle} = (U_{triangle}, A_{triangle})$*,* $+_{R_1}(IS_{rectangle}, IS_{triangle})$*, presented in tables 1, 2 and 4, respectively. We assume that* $R_1 = \{(x_i, y_j) \in U_{rectangle} \times U_{triangle}: i = 1, \ldots, 6, j = 1, 2\}$*. We also assume that* $(a,1)((x_i, y_j)) = a(x_i), (b,1)((x_i, y_j)) = b(x_i)$ *and* $(c,2)((x_i, y_j)) = c(y_j)$*, where* $i = 1, \ldots, 6$ *and* $j = 1, 2$.

The above examples are illustrating an idea of specifying constraints by examples. Table 4 can be used to construct a decision table partially specifying characteristic functions of the constraint. Such a decision table should be extended by adding relevant attributes related to the object parts making it possible to induce the high quality classifiers for the constraint relation. The classifier can be next used to filter composed pairs of objects that satisfy the constraint. This is important construction because the constraint specification usually cannot be defined directly in terms of measurable attributes. It can be specified, for example, in natural language. This is the reason that the process of inducing of the relevant classifiers for constraints can require hierarchical classifier construction (Pal *et al*. 2004).

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.

4.2 *Constrained sum of approximation spaces*

Let $IS_i = (U_i, A_i)$ for $i = 1, \ldots, k$ be information systems. Let $AS_{\#_i} = (U_i, I_{\#_i}, \nu_{SRI})$ be an approximation space for information system IS_i, where $i = 1, \ldots, k$ and let $R \subseteq U_1 \times \cdots \times U_k$ be a constraint relation. We define an approximation space $+_R(AS_{\#_1}, \ldots, AS_{\#_k})$ for $+_R(IS_1, \ldots, IS_k)$ as follows:

1. the universe is equal to R,
2. $I_{\#_1,\ldots,\#_k}((x_1, \ldots, x_k)) = (I_{\#_1}(x_1) \times \cdots \times I_{\#_k}(x_k)) \cap R$,
3. the inclusion relation ν_{SRI} in $+_R(AS_{\#_1}, \ldots, AS_{\#_k})$ is the standard inclusion function.

Proposition 17 *Let $X_i \subseteq U_i$ for $i = 1, \ldots, k$. We obtain the following properties of approximations*:

$$\begin{aligned} &LOW(+_R(AS_{\#_1}, \ldots, AS_{\#_k}), X_1 \times \cdots \times X_k) \\ &\quad = R \cap (LOW(AS_{\#_1}, X_1) \times \cdots \times LOW(AS_{\#_k}, X_k)), \qquad (5) \\ &UPP(+_R(AS_{\#_1}, \ldots, AS_{\#_k}), X_1 \times \cdots \times X_k) \\ &\quad = R \cap (UPP(AS_{\#_1}, X_1) \times \cdots \times UPP(AS_{\#_k}, X_k)). \qquad (6) \end{aligned}$$

Example 18 *Let us consider the approximation spaces AS_{A_1} and AS_{A_2} defined in Example 14. We recall, that $AS_{A_1} = (U_{rectangle}, I_{A_1}, \nu_{SRI})$ where $y \in I_a^5(x)$ if and only if $|a(x) - a(y)| \leq 5$. This means that rectangles x and y are similar with respect to the length of horizontal sides if and only if the difference of lengths is not greater than 5 millimetres. Let $y \in I_b(x)$ if and only if $b(x) = b(y)$ and $y \in I_{A_1}(x)$ if and only if $\forall_{c \in A_1} y \in I_c(x)$.*

We obtain that the lower approximation of $\{x_1, x_2, x_3\} \times \{y_2\}$ is equal to

$$\begin{aligned} &LOW(+_{R_1}(AS_{A_1}, AS_{A_2}), (\{x_1, x_2, x_3\} \times \{y_2\})) \\ &\quad = R_1 \cap (LOW(AS_{A_1}, \{x_1, x_2, x_3\}) \times LOW(AS_{A_2}, \{y_2\})) \\ &\quad = R_1 \cap (\{x_1, x_3\} \times \{y_2\}) = \{(x_1, y_2), (x_3, y_2)\}. \end{aligned}$$

We also obtain that the upper approximation of $\{x_1, x_2, x_3\} \times \{y_2\}$ is equal to

$$\begin{aligned} &UPP(+_{R_1}(AS_{A_1}, AS_{A_2}), (\{x_1, x_2, x_3\} \times \{y_2\})) \\ &\quad = R_1 \cap (UPP(AS_{A_1}, \{x_1, x_2, x_3\}) \times UPP(AS_{A_2}, \{y_2\})) \\ &\quad = R_1 \cap (\{x_1, x_2, x_3, x_4, x_6\} \times \{y_2\}) = \{(x_1, y_2), (x_2, y_2), (x_3, y_2), (x_4, y_2), (x_6, y_2)\}. \end{aligned}$$

5. Hierarchical modelling

Problems of approximation of complex concepts create nowadays a challenge for science (see, for example, Vapnik 1998, Breiman 2001, Poggio and Smale 2003, Skowron and Stepaniuk 2004). For example, in identification of dangerous situations on the road by unmanned vehicle aircraft (UAV), the target concept is too complex to be directly approximated from feature value vectors. One of the emerging approaches to deal with such cases is the hierarchical (or layered) learning (Stone 2000, Bazan *et al*. 2004) approach to concept approximation.

In our project, 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, for example, 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 (Pawlak 1991). 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 (Polkowski and Skowron 1996, 1999, Skowron and Stepaniuk 2004) 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 (Skowron and Stepaniuk 1996, 2004).

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

Definition 19 *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 $DT = (\mathcal{N}_{\mathcal{R}}, A, d)$ (Pawlak 1991).*

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 $N(x)$ to U_0 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 (Pawlak 1991). Then a granulation of these relations should be performed what leads to relations defined on neighbourhoods of objects rather than on objects (Peters *et al*. 2003).

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 (Skowron *et al*. 2003). 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 generalizations of windows, widely used in temporal reasoning (e.g. time windows in time series analysis (Friedman 2001)).

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 (Pawlak 1991). 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 (Skowron and Stepaniuk 2004).

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 object tuples of composed information systems. Formulas expressing constraints 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.

6. 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 search for 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.

6.1 *Conjunctions of descriptors and conjunctions of generalized 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 realized by generating decision rules from a decision table constructed in the following way (see also Skowron and Stepaniuk 2004):

(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, for example, Bazan *et al.* 2000). 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 (Skowron and Stepaniuk 2004).

Patterns from the language $\mathcal{L}_{des}$ are the simplest ones. The generalization is obtained by conjunction 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 generalized 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 generalized descriptors over attributes from information systems composed by the constrained sum. Heuristic searching for such patterns have been proposed by several authors (see, for example, Nguyen and Skowron 1997, Nguyen *et al*. 1998, Nguyen and Nguyen 1998a). 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.

6.2 *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 (Bazan *et al*. 2004). We present an example illustrating such an approach.

For a given a decision table $\mathcal{A} = (U, A, d)$ with $V_d = \{1, \ldots, r\}$ (Pawlak 1991) by **RULES**($\mathcal{A}$) we denote a set of decision rules induced by some rule extraction method (Bazan *et al*. 2000, RSES). For any new object $x \in \mathcal{U}$, where $U \subseteq \mathcal{U}$ let $MatchRules(\mathcal{A}, x)$ be the set of rules from **RULES**($\mathcal{A}$) supported by x. One can define the rough membership function $\mu_{CLASS_k} : \mathcal{U} \rightarrow [0, 1]$ for the concept determined by $CLASS_k = \{x \in U\colon d(x) = k\}$ (where $k = 1, \ldots, r$) by

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

$$w_{yes}(x) = \sum_{\mathbf{r} \in \mathbf{R}_{yes}(x)} strength(\mathbf{r}) \qquad w_{no} = \sum_{\mathbf{r} \in \mathbf{R}_{no}(x)} strength(\mathbf{r}) \tag{7}$$

where strength(**r**) is a normalized function depending on *length*, *support*, *confidence* of **r** and some global information about the decision table $\mathcal{A}$ like table size, class distribution (see Bazan 1998).

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

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

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 parameterized 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 synthesizing classifiers such as k-nn (see, for example, Bazan *et al*. 2004). 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.

6.3 *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 $\mathcal{A} = (U, A, d)$ 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 (Pawlak 1991) 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 (Zadeh 1965, Pawlak 1991, Skowron and Stepaniuk 2004). Let us consider such new decision table $\mathcal{A}' = (U, A, deg)$.

For given reals $0 < c_1 < \cdots < c_k$ 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:

$$UPP(AS_B, (X_i - X_{i+1})) \subseteq (X_{i-1} - X_{i+2}), \quad \text{for } i = 1, \ldots, k \tag{8}$$

is called relevant for approximation of cuts $0 < c_1 < \cdots < c_k$ in $\mathcal{A}'$.

The condition (8) 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 $\mathcal{A}'$ consists of tuples $(B, c_1, \ldots, c_k)$ defining approximations of regions between cuts, i.e.

$$(LOW(AS_B, (X_i - X_{i+1})), UPP(AS_B, (X_i - X_{i+1}))), \quad \text{for } i = 0, \ldots, k \tag{9}$$

where we assume that B is relevant for approximation of cuts $0 < c_1 < \cdots < c_k$ in $\mathcal{A}'$.

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 $\mathcal{A}'$ (on a sample U) 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 6.2. Searching for patterns defining constraints aims at extracting patterns from such language corresponding to the arguments of the constrained sum that after composing create patterns included in the target concept to a satisfactory degree.

6.4 *Modelling relations between objects*

Some relations between objects, e.g. binary relations, can be specified by information systems (decision tables) in which objects are tuples (e.g. pairs) of objects. Such data tables can be treated as a partial specification of relational structure. Observe, that a single information system is a representation of a relational structure with unary predicates defined by descriptors of this system. From such tables the classifiers for relations can be induced. These classifiers define approximations of relations given on data samples from the whole universe of objects. For other relations constrained sums of information systems can be used to construct relevant patterns for their approximation. Next, over such patterns are constructed classifiers defining approximations of relations. The main scheme of construction is the following: any constraint for the constrained sum is a formula (pattern) definable in terms of patterns definable on the level of arguments for the constrained sum. Certainly, the phrase 'definable' should be specified for each language of constraints.

6.5 *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 $a_*(x) = new$ (see, e.g., table 5).

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 20 *We consider data represented in table 1 and clusters defined by the function I_a. We define the relation Sim_a by*

$(x_i, x_j) \in Sim_a$ *if and only if* $x_i \in I_a(x_j)$
thus we obtain the following clusters:
$\{x_1, x_3, x_5, x_6\}$, $\{x_2, x_4, x_6\}$, $\{x_1, x_3, x_5\}$,
$\{x_2, x_4\}$, $\{x_1, x_3, x_5\}$, $\{x_1, x_2, x_6\}$.

The information system $IS_* = (\{x_1, \ldots, x_6, *\}), \{a_*, b_*\})$ is presented in table 5.

Table 5. Information system IS_*.

$U_{rectangle} \cup \{*\}$	a_*	b_*
x_1	165	Yes
x_2	175	No
x_3	160	Yes
x_4	180	No
x_5	160	No
x_6	170	No
$*$	New	New

Table 6. Exemplary decision table DT_3 after discretization of DT_1.

Classes	Number of objects	h	D
X_1	50	< 150	S
X_2	50	[150,152)	$S > M$
X_3	50	[152,157)	$S \approx M$
X_4	50	[157,160)	$S < M$
X_5	50	[160,175)	M

In the constrained sum $+_R(IS_*, IS_*, IS_*, IS_*, IS_*, IS_*)$ *the universe is equal to* $\{(x_1, *, x_3, *, x_5, x_6), (*, x_2, *, x_4, *, x_6), (x_1, *, x_3, *, x_5, *), (*, x_2, *, x_4, *, *), (x_1, *, x_3, *, x_5, *), (x_1, x_2, *, *, *, x_6)\}$.
Observe also that 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. For example 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

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

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

7. Modelling vague concepts

In this section, we discuss an example inspired by Barwise and Seligman (1997: 211–215). In general, an *information net* (Barwise and Seligman 1997) (infonet, for short) is a labelled graph with nodes labelled by information systems and edges labelled by infomorphisms between information systems (see, for example, figure 2). We would like to show how rough set approximations of primitive concepts linked by infonets can be used in construction of approximations of more complex concepts.

Assume for $i = 1, 2$ decision tables (Pawlak 1991) $DT_i = (U_i, A_i, d_i)$ are given where

- $U_i = \{x_1, \ldots, x_{250}\}$,
- $A_i = \{h\}$ where $V_h = \{1, \ldots, 220\}$,
- $V_{d_i} = \{S, M, T\}$.

Hence, in each decision table we consider 250 objects. For any object x the value $h(x)$ represents the object x height (in cm). The decision values S, M, T correspond to the concept names: *short*, *medium*, *tall*, respectively.

We consider a typical case when the decision tables are inconsistent. It means that some objects from tables with the same height (e.g. 157 cm) have different decisions (in our case, for example, S and M). Consequently the concepts S, M, T cannot be

Table 7. Exemplary decision table for *TALLER* approximations.

$X_i \times Y_j$	Number of objects	D_X	D_Y	*TALLER*
$X_1 \times Y_1$	2500	S	M	Yes
$X_2 \times Y_4$	2500	$S > M$	$S < M$	Yes
$X_3 \times Y_2$	2500	$S \approx M$	$S > M$	Yes, No

defined exactly by means of the attribute h. Using h one can define lower and upper approximations of such concepts (Pawlak 1991). One can discretize the attribute h preserving the approximations as well as the components of boundary regions between the lower approximations. These components can characterize, for example, the distribution of objects between decisions S, M, T. In the example, we assume that the discretization preserves the following constraints:

- If the decision for a given value of h was consistent before discretization then it remains the same after discretization. It means that the discretization preserves the positive region (Pawlak 1991).
- If the decision for a given value of h was inconsistent before discretization and the set of all possible decisions was equal to V then this decision set does not changes after discretization. The last two constraints allow us to preserve the generalized decision.
- Discretization makes it possible to join only such indisernibility classes of the original decision tables which have similar distributions of decisions.

For simplicity, in our example we consider only the following types of distributions:

- Singletons S, M, T (for consistent decisions S, M, T),
- $S > M$ (for indiscernibility classes with much higher number of objects with the decision S than the decision M),
- $S < M$ (for indiscernibility classes with much higher number of objects with the decision M than the decision S),
- $S \approx M$ (for indiscernibility classes with almost the same number of objects with the decision M and the decision S).

We will not present the details how the vagues terms *much higher*, *almost the same* are defined because this is not important for our illustrative example. Assume that after discretization the decision table presented in table 6 is obtained.

After discretization of DT_2 we obtain similar table DT_4. Now, we would like to present the decision table DT with the decision corresponding to concept $TALLER(x, y)$. In the construction of such a decision table, we use condition attributes related to parts x, y. These attributes are derived from attributes in DT_3 and DT_4. Finally, one can observe that we obtain infomorphisms between DT_3, DT_4 and DT. We will concentrate now on an example showing how such a construction makes it possible to derive approximations of the concept *TALLER*.

Let us consider a fragment of DT presented in table 7.

From this table one can derive patterns for approximations of the concept *TALLER*. For example, the following patterns are in the lower approximation of *TALLER*: $D_X = S \wedge D_Y = M$, $D_X = S > M \wedge D_Y = S < M$, while the pattern $D_X = S \approx M \wedge D_Y = S > M$, is in the upper approximation of *TALLER*.

Our example shows a rough set approach for vague concept approximation. It is based on subjective knowledge which can be hierarchical. Levels of the hierarchy can be linked by infomorphisms.

8. Conclusions

We have outlined an approach based on constrained sum operations for hierarchical modelling of patterns. We plan to apply the approach for construction of patterns relevant for approximation of complex concepts. Using domain knowledge one can construct networks of parameterized constrained sums, i.e. in such networks each internal node is labelled by a set of constraints rather than by one constraint. A software system making it possible to search for relevant patterns by optimization of such parameters is the main goal of our project. We also plan to investigate the relationship of the presented approach with relational learning.

Acknowledgements

The research has been supported by the grants 3 T11C 002 26 and 4 T11C 014 25 from Ministry of Scientific Research and Information Technology of the Republic of Poland. This paper is in memory of Jan Żytkow.

References

J.G. Bazan, "A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision tables", in *Rough Sets in Knowledge Discovery 1: Methodology and Applications*, L. Polkowski and A. Skowron, Eds (Physica-Verlag, Heidelberg), 1998, 321–365.

J. Bazan, H.S. Nguyen, S.H. Nguyen, P. Synak and J. Wróblewski, "Rough set algorithms in classification problems", in *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. Studies in Fuzziness and Soft Computing*, L. Polkowski, Y.T. Lin and S. Tsumoto Eds (Physica-Verlag, Heidelberg) 2000, 49–88.

J. Bazan, S.H. Nguyen, H.S. Nguyen and A. Skowron, "Rough set methods in approximation of hierarchical concepts", in *Fourth International Conference on Rough Sets and Current Trends in Computing* (*RSCTC'04*), 2004, 346–355.

J. Barwise and J. Seligman, "Information flow: the logic of distributed systems", Cambridge University Press Tracts in Theoretical Computer Science, 1997, 44.

L. Breiman, "Statistical modeling: the two cultures", *Statistical Science*, 2001, **16**(3), 199–231.

J.H. Friedman, T. Hastie and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer-Verlag, Heidelberg), 2001.

H. Garcia-Molina, D.J. Ullman and D.J. Widom, *Database Systems: The Complete Book* (Prentice Hall, Upper Saddle River, NJ), 2002.

W. Kloesgen, J. Żytkow, (Eds), *Handbook of Knowledge Discovery and Data Mining* (Oxford University Press, Oxford), 2002.

J. Łukasiewicz, "Die logischen grundlagen der Wahrscheinlichkeitsrechnuung, Kraków", in *Jan Łukasiewicz—Selected Works*, L. Borkowski, Ed. (North Holland Publishing Company, Amstardam, London), 1913.

S.H. Nguyen and A. Skowron, "Searching for relational patterns in data", in *Proceedings: the First European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'97)*, 1997, 265–276.

H.S. Nguyen and S.H. Nguyen, "Discretization methods in data mining", in *Rough Sets in Knowledge Discovery 1. Methodology and Applications*, L. Polkowski and A. Skowron Eds (Physica-Verlag, Heidelberg), 1998a, 451–482.

S.H. Nguyen, A. Skowron and P. Synak, "Discovery of data patterns with applications to decomposition and classification problems", in *Rough Sets in Knowledge Discovery 2.*

Applications, Case Studies and Software Systems, L. Polkowski and A. Skowron Eds (Physica-Verlag, Heidelberg), 1998, 55–97.

H.S. Nguyen and S.H. Nguyen, "Pattern extraction from data", *Fundamenta Informaticae*, 34, 1998b, 129–144.

S.H. Nguyen, "Regularity analysis and its applications in data mining", in *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems. Studies in Fuzziness and Soft Computing*, L. Polkowski, T.Y. Lin and S. Tsumoto Eds (Physica-Verlag, Heidelberg), 2000, 56, 289–378.

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

J.F. Peters, A. Skowron, P. Synak and S. Ramanna, "Rough sets and information granulation", *Tenth International Fuzzy Systems Association World Congress IFSA*, T. Bilgic, D. Baets and O. Kaynak Eds, 2003, 370–377.

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

T. Poggio and S. Smale, "The mathematics of learning: dealing with data", *Notices of the AMS*, 50, 2003, 537–544.

L. Polkowski and A. Skowron, "Rough mereology: a new paradigm for approximate reasoning", *International Journal of Approximate Reasoning*, 15, 1996, 333–365.

L. Polkowski and A. Skowron, "Towards adaptive calculus of granules", in *Computing with Words in Information/Intelligent Systems 1–2*, L.A. Zadeh and J. Kacprzyk, Eds (Physica-Verlag, Heidelberg), 1999, 201–227.

A. Skowron and J. Stepaniuk, "Tolerance approximation spaces", *Fundamenta Informaticae*, 27, 1996, 245–253.

A. Skowron and J. Stepaniuk, "Information granules: towards foundations of granular computing", *International Journal of Intelligent Systems*, 16, 2001, 57–86.

A. Skowron and J. Stepaniuk, "Information granules and rough-neuro computing", in *Rough Neural Computing: Techniques for Computing with Words*, S.K. Pal, L. Polkowski and A. Skowron, Eds (Springer Verlag, Berlin), 2004, 43–84.

A. Skowron, J. Stepaniuk and F.J. Peters, "Rough sets and infomorphisms: towards approximation of relations in distributed environments", *Fundamenta Informaticae*, 2003, **54**, 263–277.

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

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

V. Vapnik, *Statistical Learning Theory* (Wiley, New York), 1998.

L.A. Zadeh," Fuzzy sets", *Information and Control*, 8, 1965, 333–353.

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

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

L.A. Zadeh, "A new direction in AI: toward a computational theory of perceptions", *AI Magazine*, 22, 2001, 73–84.