Fundamenta Informaticae 54 (2003) 263–277 263
IOS Press

Rough Sets and Infomorphisms: Towards Approximation of Relations in Distributed Environments

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

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

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

Abstract. We discuss the relationships between information systems and classifications as well as between infomorphisms and definability of relations (between whole objects and their parts) in information systems. Infomorphisms between information systems (classifications) IS_1 and IS_2 make it possible to define some formulas over IS_2 by means of formulas over IS_1. The remaining formulas over IS_2 can be approximatively defined by means of formulas over IS_1. The approximation operations are defined using the rough set approach. We present definitions and examples of such approximations.

[*]The research has been supported by the State Committee for Scientific Research of the Republic of Poland (KBN) research grant 8 T11C 025 19 and partially by the Wallenberg Foundation grant. The paper has been prepared within the framework of COST Action 274 "Theory and Application of Relational Structures as Knowledge Instruments".
Address for correspondence: Institute of Mathematics, University of Warsaw, Banacha 2, 02-097 Warszawa, Poland
[†]The research has been supported by the State Committee for Scientific Research of the Republic of Poland (KBN) research grant 8 T11C 025 19.
[‡]The research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) research grant 185986.

Keywords: information systems, classifications, rough set theory, infomorphisms, definability, approximations, approximate reasoning, information nets

1. Introduction

In constructing intelligent systems it is necessary to develop tools for approximate reasoning about vague and incomplete concepts in distributed environments [25], [18], [21], [22], [13], [14], [23],[15]. In this paper we discuss relationships between information flow [2] and rough sets [12]. The former approach has been introduced to deal with reasoning in distributed systems while the latter makes it possible to deal with incomplete and vague concepts. Infomorphisms between information systems IS_1 and IS_2 are basic links between information systems. They make it possible to introduce logics of distributed systems modelled by nets (channels) of information systems connected by infomorphisms [2].

The aim of our project is to investigate if the logics defined by nets of information systems can be extended to the case of approximate reasoning. The paper realizes the first step in this direction. We distinguish formulas over IS_2 definable by means of formulas over IS_1 assuming there exists an infomorphism from IS_1 to IS_2. The formulas over IS_2 not-definable by formulas over IS_1 can be approximated by means of formulas over IS_1. The approximations are defined using the rough set approach. We present definitions and examples of such approximations. If there are infomorphisms from any information system from a given family to a given information system IS, then approximations of formulas (types [2]) over IS can be constructed by means of composition of patterns from IS definable in information systems from the family. Certainly, this requires the assumption that the set of formulas (types) of IS is closed with respect to the composition operation. The contribution of this article is the introduction of approximations of concepts not-definable exactly (or types, using terminology from [2]; see, also, [19]) in nets of information systems linked by infomorphisms. This makes it possible to reason in an approximate way in a given node of information net about concepts definable in the other nodes.

The paper is organized as follows. In Section 2 we show that classifications [2] and information systems [9, 10, 11, 12] are equivalent. We also recall the infomorphism definition [2]. In Section 3 we discuss approximations of concepts in information systems linked by infomorphisms. An example of such approximations is also included. Approximations of concepts in information nets are introduced in Section 4.1 together with an illustrative example. Section 4.2 is dedicated to approximate reasoning in information nets. In Section 5 we show how rough set approximations of primitive concepts linked by infonets can be used in construction of approximations of more complex concepts.

2. Basic Concepts

In this section we present basic notions for our approach, i.e., information systems and infomorphisms.

2.1. Information Systems and Classifications

In this section we show that information systems [12] and classifications [2], [1], [6] are equivalent. First, let us recall these basic concepts.

Let $IS = (U, A)$ be an *information system*, where U is a set of objects and A is a set of attributes. For every $a \in A$ let V_a be a set of values of attribute a. For $B \subseteq A$ we denote by $IND(B)$ the

B-indiscernibility relation defined by [12]:

$$xIND(B)y \text{ if and only if } a(x) = a(y) \text{ for any } a \in B. \tag{1}$$

A decision system is any information system with a distinguished attribute called the decision. Decision systems will be denoted by $DT = (U, A, d)$ where U is the set of objects, A is a set of so called condition attributes and d is the decision.

By IS/A we denote the A-quotient system $(U/A, A)$ where U/A is the set $\{[x]_A : x \in U\}$ of all equivalence (indiscernibility) classes of $IND(A)$ and $a([x]_A) = a(x)$ for any $a \in A$ and $x \in U$.

Definition 2.1. [2] A *classification* is any tuple

$$\mathcal{A} = (\Sigma_{\mathcal{A}}, C, \models_{\mathcal{A}}) \tag{2}$$

where $\Sigma_{\mathcal{A}}, C$ are sets called the set of *types* and the set of *tokens*, respectively and $\models_{\mathcal{A}}$ is a binary relation in $\Sigma_{\mathcal{A}} \times C$, i.e., $\models_{\mathcal{A}} \subseteq \Sigma_{\mathcal{A}} \times C$.

The notation $x \models_{\mathcal{A}} \alpha$ reads "*x satisfies α relative to classification $\mathcal{A}$*" for $x \in C$ and $\alpha \in \Sigma_{\mathcal{A}}$. This is an analogy to the satisfiability relation used widely in logic.

Example 2.1. Consider the classification $\mathcal{A} = (\Sigma_{\mathcal{A}}, C, \models_{\mathcal{A}})$ where $\Sigma_{\mathcal{A}}$ is a set of design patterns as in [3], C is a set of designs, and relation $x \models_{\mathcal{A}} \alpha$ asserts that design x conforms (i.e., satisfies the requirements of) pattern α. Informally, a design pattern is a schema (representation of a plan in the form of an outline) for a reusable solution to a general problem. In effect, the schema of a design pattern defines a type (class of designs).

Any classification $\mathcal{A}$ defines an information system $IS_{\mathcal{A}} = (U, A)$ where $U = C$ and $A = \{a_\alpha\}_{\alpha \in \Sigma_{\mathcal{A}}}$ where $a_\alpha : C \to \{0, 1\}$ and $a_\alpha(x) = 1$ if and only if $x \models_{\mathcal{A}} \alpha$ for any type $\alpha \in \Sigma_{\mathcal{A}}$.

Example 2.2. The classification $\mathcal{A}$ from Example 2.1 defines an information system $IS_{\mathcal{A}} = (U, A)$ where $U = C$, $A = \{a_\alpha\}_{\alpha \in \Sigma_{\mathcal{A}}}$ where $a_\alpha(x) = 1$ if and only if $x \models_{\mathcal{A}} \alpha$ for any pattern $\alpha \in \Sigma_{\mathcal{A}}$. That is, $a_\alpha(x) = 1$ means that design x satisfies the requirements specified by pattern α. Otherwise, $a_\alpha(x) = 0$ and x fails to satisfy the requirements for pattern α.

Any binary information system $IS = (U, A)$, i.e., an information system where $V_a = \{0, 1\}$ for any $a \in A$ defines a classification $\mathcal{A}_{IS} = (\Sigma_{\mathcal{A}}, C, \models_{\mathcal{A}})$ where $\Sigma_{\mathcal{A}} = \{\alpha_a\}_{a \in A}$, $C_{IS} = U$ and $x \models_{\mathcal{A}} \alpha_a$ if and only if $a(x) = 1$.

We have the following proposition:

Proposition 2.1. For any binary information system IS and for any classification $\mathcal{A}$ we obtain

$$IS_{\mathcal{A}_{IS}} = IS \tag{3}$$

$$\mathcal{A}_{IS_{\mathcal{A}}} = \mathcal{A}. \tag{4}$$

For information system $IS = (U, A)$ with arbitrary attributes (i.e., multi-valued attributes), one can consider its binary representation, i.e., an information system $IS_{bin} = (U, A_{bin})$ where $A_{bin} = \{(a =$

$v) : (a \in A \wedge v \in V_a) \wedge \exists c(c \in U \wedge a(c) = v)\}$ and $(a = v)(c) = 1$ if and only if $a(c) = v$. The indiscernibility relations of IS and IS_{bin} are the same, i.e., we have $IND(A) = IND(A_{bin})$.

Hence, to this end we will not distinguish between information systems and classifications defined by them.

Let us consider one more classification related to information systems.

We denote by $\Sigma(IS)$ a set of formulas over 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. $\|a \in V\|_{IS} = \{x \in U : a(x) \in V\}$.
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 $x \vDash_{IS} \alpha$ if and only if $x \in \|\alpha\|_{IS}$.

Now one can define a classification

$$\mathcal{B}ool_{IS} = (\Sigma(IS), U, \models_{IS}) \tag{5}$$

where $c \models_{IS} \alpha$ if and only if $c \in \|\alpha\|_{IS}$.

Binary attributes in the information system $IS_{\mathcal{B}ool_{IS}}$ are characteristic functions of sets defined by formulas from $\Sigma(IS)$.

Information systems (or classifications) are simple relational structures. However, they are general enough to represent information about features of complex objects perceived by different agents [5]. For example, structural objects can be represented by means of values of features related to their parts and relations between them. Hence, such relational structures are widely used in many areas including rough sets [12], [16] and information nets [2].

2.2. Infomorphisms

In this section we introduce the definition of infomorphism for two information systems. This notion is considered in [2], [6].

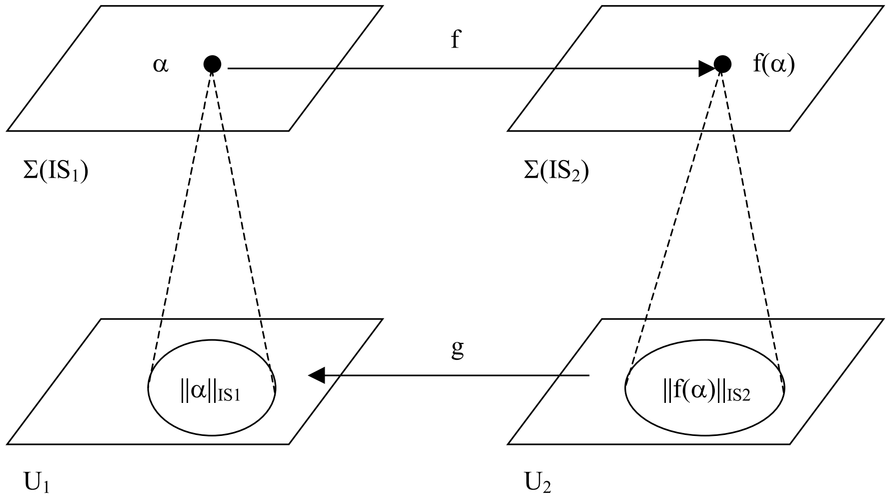

Figure 1. Infomorphism illustration

Definition 2.2. If $IS_1 = (U_1, A_1)$ and $IS_2 = (U_2, A_2)$ are information systems and $f : \Sigma(IS_1) \rightarrow \Sigma(IS_2)$, $g : U_2 \rightarrow U_1$, then an infomorphism is a pair (f, g) of functions satisfying the following equivalence

$$g(x) \models_{IS_1} \alpha \text{ if and only if } x \models_{IS_2} f(\alpha) \tag{6}$$

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

The infomorphism will be denoted shortly by $(f, g) : IS_1 \rightleftarrows IS_2$.
For any classifications $\mathcal{A}, B$ we assume $(f, g) : \mathcal{A} \rightleftarrows \mathcal{B}$ if and only if $(f, g) : IS_{\mathcal{A}} \rightleftarrows IS_{\mathcal{B}}$.

Proposition 2.2. For any infomorphism $(f, g) : IS_1 \rightleftarrows IS_2$ we obtain the following equality

$$g^{-1}(\|\alpha\|_{IS_1}) = \|f(\alpha)\|_{IS_2} \text{ for any } \alpha \in \Sigma(IS_1). \tag{7}$$

Proof:
In fact $x \in g^{-1}(\|\alpha\|_{IS_1})$ if and only if $g(x) \in \|\alpha\|_{IS_1}$. The last condition is equivalent to $x \in \|f(\alpha)\|_{IS_2}$ which follows from the infomorphism definition. □

Example 2.3. 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 is a function such that $f : \Sigma(IS_1) \to \Sigma(IS_2)$
- $g : U_2 \to U_1$
- α is a formula such that $\alpha \in \Sigma(IS_1)$ and $f(\alpha) \in \Sigma(IS_2)$
- $\|\alpha\|_{IS_1} = \{x \in U_1 : x \vDash_{IS_1} \alpha\}$
- $\|f(\alpha)\|_{IS_2} = \{x \in U_2 : x \vDash_{IS_2} f(\alpha)\}$.

Proposition 2.3. If $(f, g) : IS_1 \rightleftarrows IS_2$ then (f, g) can be extended to an infomorphism between $Bool_{IS_1}$ and $Bool_{IS_2}$.

Let us consider two examples of infomorphisms related to information systems and decision systems.

Example 2.4. Let $IS = (U, A)$ be a binary information system. Assume $f(\alpha) = \alpha$ for any $\alpha \in \Sigma(IS)$. Let g be a selection from U/A into U, i.e., a function satisfying the condition $g([x]_A) \in [x]_A$ for any $x \in U$. Then we have $(f, g) : IS \rightleftarrows IS/A$. From Proposition 2.3 it follows that there exists an infomorphism between classifications $Bool_{IS}$ and $Bool_{IS/A}$.

Example 2.5. Let $DT = (U, A, d)$ be a decision table and let $B \subseteq A$ be a decision reduct [7], i.e., a minimal set (with respect to the inclusion) such that $\delta_B(x) = \delta_A(x)$ for any $x \in U$ where $\delta_B(x) = \{v \in V_d : \exists x'(xIND(B)x' \text{ and } d(x') = v)\}$ and V_d is the value set of the decision d. We define two classification systems $\mathcal{A}_i(DT) = (\Sigma_i, U, \models_i)$ for $i = 1, 2$ related to the decision table DT and the decision reduct B defined in the following way:

- Σ_1 is the set of implications $\bigwedge_{a \in A}(a = v_a) \Rightarrow (\delta_A = V)$ such that $V \subseteq V_d$ and there exists an object $x \in U$ satisfying $\bigwedge_{a \in A}(a = v_a) \wedge (\delta_A = V)$.
- Σ_2 is the set of implications $\bigwedge_{a \in B}(a = v_a) \Rightarrow (\delta_B = V)$ such that $V \subseteq V_d$ and there exists an object $x \in U$ satisfying $\bigwedge_{a \in B}(a = v_a) \wedge (\delta_B = V)$.
- For $i = 1, 2$ $x \models_i \alpha$ if and only if $x \in \|\alpha\|_{DT_i}$ for any $x \in U$ and $\alpha \in \Sigma_i$.

Assuming $f(\bigwedge_{a \in A}(a = v_a) \Rightarrow (\delta_A = V)) = \bigwedge_{a \in B}(a = v_a) \Rightarrow (\delta_B = V)$ and g to be the identity on U we have $(f, g) : DT_1 \rightleftarrows DT_2$.

3. Approximations and Infomorphisms

In this section, we discuss approximations of concepts definable by types in classifications. Assume $(f, g) : IS_1 \rightleftarrows IS_2$. We say that a type β from $\Sigma(IS_2)$ is $\Sigma(IS_1)$-*definable* if and only if there is $\alpha \in \Sigma(IS_1)$ such that $\beta = f(\alpha)$. Then by Proposition 2.2 we have $g^{-1}(\|\alpha\|_{IS_1}) = \|\beta\|_{IS_2}$ which explains why we say that β is $\Sigma(IS_1)$-definable by α. Hence, types of IS_2 from the f-image of $\Sigma(IS_1)$ are definable in IS_1.

Proposition 3.1. If $(f, g) : IS_1 \rightleftarrows IS_2$ then any type from $f(\Sigma(IS_1))$ is $\Sigma(IS_1)$-definable.

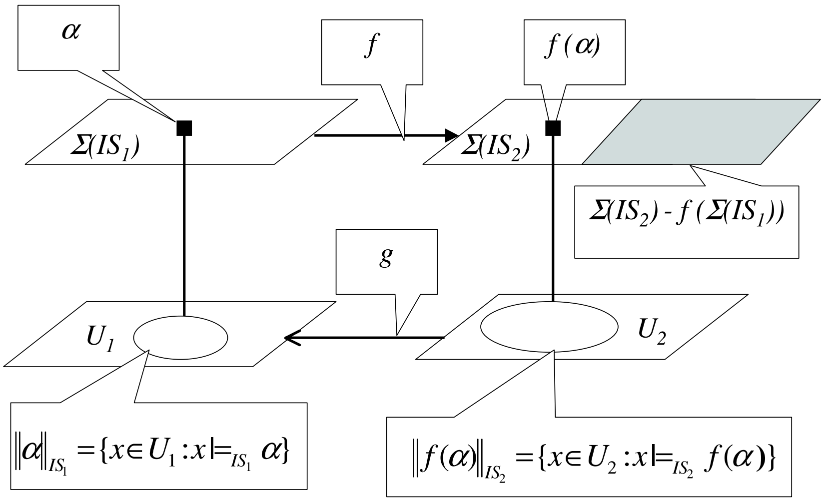

Figure 2. Approximations of types

Now, we show that any $\alpha \in \Sigma(IS_2) - f(\Sigma(IS_1))$ can be approximated by types from $\Sigma(IS_2)$ (and $\Sigma(IS_1)$).

We define approximations analogously to the rough set approach [12].

Definition 3.1. For any type $\alpha \in \Sigma(IS_2) - f(\Sigma(IS_1))$ we define its $\Sigma(IS_1)$–lower and $\Sigma(IS_1)$–upper approximations by

$$\underline{\Sigma(IS_1)}\alpha = \bigcup_{\beta \in \Sigma(IS_1)} \{\|f(\beta)\|_{IS_2} : \|f(\beta)\|_{IS_2} \subseteq \|\alpha\|_{IS_2}\} \tag{8}$$

$$\overline{\Sigma(IS_1)}\alpha = \bigcup_{\beta \in \Sigma(IS_1)} \{\|f(\beta)\|_{IS_2} : \|f(\beta)\|_{IS_2} \cap \|\alpha\|_{IS_2} \neq \emptyset\}. \tag{9}$$

Proposition 3.2. We have the following equalities:

$$\underline{\Sigma(IS_1)}\alpha = \bigcup_{\beta \in \Sigma(IS_1)} \{g^{-1}(\|\beta\|_{IS_1}) : \|f(\beta)\|_{IS_2} \subseteq \|\alpha\|_{IS_2}\} \tag{10}$$

$$\overline{\Sigma(IS_1)}\alpha = \bigcup_{\beta \in \Sigma(IS_1)} \{g^{-1}(\|\beta\|_{IS_1}) : \|f(\beta)\|_{IS_2} \cap \|\alpha\|_{IS_2} \neq \emptyset\}. \tag{11}$$

Example 3.1. Let $IS_1 = (U_1, A_1)$ and $IS = (U, A)$ be information systems, where $U_1 = \{x_1, \ldots, x_6\}$ and $U = U_1 \times \{y_1, y_2\}$ (see Table 1 and Table 2, respectively).

Let $A_1 = \{a, b\}$, $V_a = \{v_1, v_2\}$ and $V_b = \{w_1, w_2\}$. We assume that $A = \{a', b', d\}$, $V_d = \{+, -\}$, $a'((x_i, y_j)) = a(x_i)$ and $b'((x_i, y_j)) = b(x_i)$, where $i = 1, \ldots, 6$ and $j = 1, 2$. Let $f : \Sigma(IS_1) \rightarrow \Sigma(IS)$ be such that for every formula $\alpha \in \Sigma(IS_1)$ $f(\alpha) = \alpha'$, where α' is obtained from α by translation of $a = v_i$ to $a' = v_i$ and $b = w_j$ to $b' = w_j$, where $i, j = 1, 2$.

U_1	a	b
x_1	v_2	w_1
x_2	v_1	w_2
x_3	v_2	w_1
x_4	v_1	w_2
x_5	v_2	w_2
x_6	v_2	w_2

Table 1. Information System IS_1

Let $g : U \to U_1$ be a projection, that is, for every $(x, y) \in U$ $g((x, y)) = x$.
Let α be a formula $d = +$. We have $\|\alpha\|_{IS_1} = X_+ = \{(x, y) \in U_2 : d((x, y)) = +\}$.
The IS_1-lower approximation of X_+ is equal to

$$\{(x_1, y_1), (x_1, y_2), (x_3, y_1), (x_3, y_2)\} = g^{-1}(\{x_1, x_3\}). \tag{12}$$

The IS_1-upper approximation of X_+ is equal to

$$\{(x_1, y_1), (x_1, y_2), (x_3, y_1), (x_3, y_2), (x_2, y_1), (x_2, y_2), (x_4, y_1), (x_4, y_2)\} = g^{-1}(\{x_1, x_2, x_3, x_4\}). \tag{13}$$

One can observe that computed approximations coincide with the $\{a', b'\}$-lower and $\{a', b'\}$-upper approximations of the decision class X_+ using the rough set approach.

Assume $(f, g) : IS_1 \rightleftarrows IS_2$. We say that a type α from $\Sigma(IS_1)$ is $\Sigma(IS_2)$-*definable* if and only if $\|\alpha\|_{IS_1} \subseteq g(U_2)$. The latter assumed condition for α implies

$$g(\|f(\alpha)\|_{IS_2}) = \|\alpha\|_{IS_1} \tag{14}$$

what explains why we say that α is $\Sigma(IS_2)$-definable (by $f(\alpha)$).

Proposition 3.3. If $(f, g) : IS_1 \rightleftarrows IS_2$ then any type $\alpha \in \Sigma(IS_1)$ satisfying $\|\alpha\|_{IS_1} \subseteq g(U_2)$ is $\Sigma(IS_2)$-definable (by $f(\alpha)$).

For any $\alpha \in \Sigma(IS_2) - f(\Sigma(IS_1))$ the difference

$$\overline{\Sigma(IS_1)}\alpha - \underline{\Sigma(IS_1)}\alpha \tag{15}$$

is called the $\Sigma(IS_1)$-*boundary region* of α and is denoted by $BN_{\Sigma(IS_1)}(\alpha)$.

One can observe that the quality of approximation of any type $\alpha \in \Sigma(IS_2)$ (with nonempty $\Sigma(IS_1)$-upper approximation) can be measured by means of the following coefficient:

$$\gamma_{IS_1}(\alpha) = 1 - \frac{card(BN_{\Sigma(IS_1)}(\alpha))}{card(\overline{\Sigma(IS_1)}\alpha)} = \frac{card(\underline{\Sigma(IS_1)}\alpha)}{card(\overline{\Sigma(IS_1)}\alpha)}. \tag{16}$$

$\gamma_{IS_1}(\alpha)$ is called the $\Sigma(IS_1)$-*degree* of approximation of α in IS_1.

Certainly, one can choose other measures based, e.g., on entropy.

Now it is possible to introduce a concept of approximations of one information system by another one under the constraint that they are linked by infomorphism.

U	a'	b'	d
(x_1, y_1)	v_2	w_1	$+$
(x_1, y_2)	v_2	w_1	$+$
(x_2, y_1)	v_1	w_2	$+$
(x_2, y_2)	v_1	w_2	$+$
(x_3, y_1)	v_2	w_1	$+$
(x_3, y_2)	v_2	w_1	$+$
(x_4, y_1)	v_1	w_2	$+$
(x_4, y_2)	v_1	w_2	$-$
(x_5, y_1)	v_2	w_2	$-$
(x_5, y_2)	v_2	w_2	$-$
(x_6, y_1)	v_2	w_2	$-$
(x_6, y_2)	v_2	w_2	$-$

Table 2. Information System IS

Definition 3.2. Let $\varepsilon \in [0, 1]$ be a given real number. An information system IS_2 is ε–approximated by IS_1 if and only if there exists an infomorphism $(f, g) : IS_1 \rightleftarrows IS_2$ satisfying the following condition:

$$\gamma_{IS_1}(\alpha) \geq \varepsilon \tag{17}$$

for any $\alpha \in \Sigma(IS_2)$.

4. Reasoning in Information Nets

4.1. Information Nets

Given two information systems IS_1 and IS_2, these information systems can be combined into a single information system IS. Thus we obtain information net with three information systems (see Figure 3).

In general, an *information net* [2] is a labelled graph with nodes labelled by information systems and edges labelled by infomorphisms between information systems. Let us recall that types (formulas) from $\Sigma(IS)$ are closed with respect to the conjunction ($\wedge$). Moreover, we assume $\|\alpha \wedge \beta\|_{IS} = \|\alpha\|_{IS} \cap \|\beta\|_{IS}$. In such case we can use the conjunction of patterns (types) from $f_1(\Sigma(IS_1))$ and $f_2(\Sigma(IS_2))$ for approximations of types (formulas) from $\Sigma(IS) - (f_1(\Sigma(IS_1)) \cup f_2(\Sigma(IS_2)))$.

We define approximations using an approach analogous to the rough set approach [12].

Definition 4.1. For type $\alpha \in \Sigma(IS)$ we define its $\Sigma(IS_1, IS_2)$–lower approximation by

$$\underline{\Sigma(IS_1, IS_2)}\alpha = \bigcup_{\beta \in \Sigma(IS_1)} \bigcup_{\gamma \in \Sigma(IS_2)} \{\|f_1(\beta) \wedge f_2(\gamma)\|_{IS} : \|f_1(\beta) \wedge f_2(\gamma)\|_{IS} \subseteq \|\alpha\|_{IS}\} \tag{18}$$

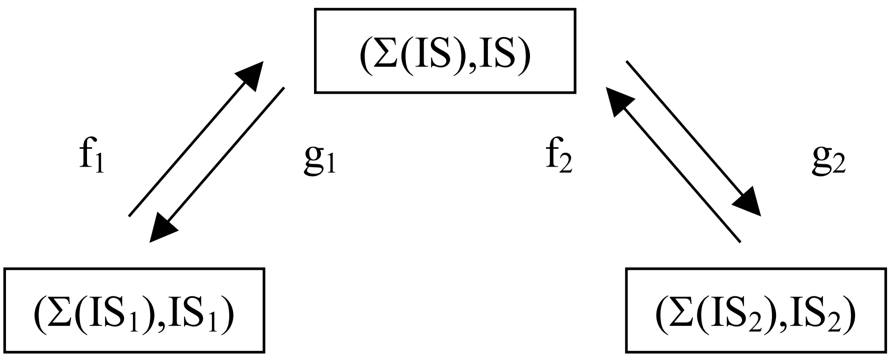

Figure 3. Information channel (the simplest net) of information systems IS_1, IS_2, IS

and $\Sigma(IS_1, IS_2)$–upper approximation by

$$\overline{\Sigma(IS_1, IS_2)}\alpha = \bigcup_{\beta \in \Sigma(IS_1)} \bigcup_{\gamma \in \Sigma(IS_2)} \{\|f_1(\beta) \wedge f_2(\gamma)\|_{IS} : \|f_1(\beta) \wedge f_2(\gamma)\|_{IS} \cap \|\alpha\|_{IS_2} \neq \emptyset\}. \quad (19)$$

Example 4.1. Let us consider three information systems $IS_1 = (U_1, A_1)$, $IS_2 = (U_2, A_2)$ and $IS = (U, A)$ presented in Table 1, Table 3 and Table 4, respectively. We obtain the following indiscernibility classes in U:

U_2	c
y_1	1
y_2	0

Table 3. Information System IS_2

$$\{(x_1, y_1), (x_3, y_1)\}, \{(x_1, y_2), (x_3, y_2)\}, \{(x_2, y_1), (x_4, y_1)\},$$

$$\{(x_2, y_2), (x_4, y_2)\}, \{(x_5, y_1), (x_6, y_1)\}, \{(x_5, y_2), (x_6, y_2)\}.$$

We obtain the following $\Sigma(IS_1, IS_2)$–approximations of the decision class

$$X_+ = \{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2), (x_3, y_1), (x_3, y_2), (x_4, y_1)\}.$$

The $\Sigma(IS_1, IS_2)$–lower approximation:

$$\{(x_1, y_1), (x_3, y_1), (x_1, y_2), (x_3, y_2), (x_2, y_1), (x_4, y_1)\}.$$

U	a'	b'	c'	d
(x_1, y_1)	v_2	w_1	1	$+$
(x_1, y_2)	v_2	w_1	0	$+$
(x_2, y_1)	v_1	w_2	1	$+$
(x_2, y_2)	v_1	w_2	0	$+$
(x_3, y_1)	v_2	w_1	1	$+$
(x_3, y_2)	v_2	w_1	0	$+$
(x_4, y_1)	v_1	w_2	1	$+$
(x_4, y_2)	v_1	w_2	0	$-$
(x_5, y_1)	v_2	w_2	1	$-$
(x_5, y_2)	v_2	w_2	0	$-$
(x_6, y_1)	v_2	w_2	1	$-$
(x_6, y_2)	v_2	w_2	0	$-$

Table 4. New Information System IS

The $\Sigma(IS_1, IS_2)$–upper approximation:

$$\{(x_1, y_1), (x_3, y_1), (x_1, y_2), (x_3, y_2), (x_2, y_1), (x_4, y_1), (x_2, y_2), (x_4, y_2)\}.$$

4.2. Approximate Reasoning in Information Nets

In this section, we present some brief remarks on approximate reasoning in information nets. We consider a problem related to reasoning about concepts defined (known) in some nodes (information systems) of an information net using concepts known in other nodes of this net. For example, one can ask if it is possible to solve the membership problem for concepts defined in a given node using some other nodes.

An approach to reasoning in information nets is presented in [2] where it is shown that some inference rules preserve validity (or nonvalidity) of dependencies transformed from one node (information system) to another linked by an infomorphism [2].

In our paper, assuming $(f, g) : IS_1 \rightleftarrows IS_2$, we have shown using the rough set approach in case of a concept $X \subseteq U_2$ known at IS_2 but not definable at IS_1, it is still possible to reason in a given node IS_1 of information net about the concept X. In such a case IS_1 can use approximations of the concept X to solve the membership problem. Let us explain this in more detail.

If $(f, g) : IS_1 \rightleftarrows IS_2$ then from the infomorphism definition we have

$$x \in \|f(\alpha)\|_{IS_2} \text{ if and only if } g(x) \in \|\alpha\|_{IS_1} \tag{20}$$

for any $x \in U_2$ and $\alpha \in \Sigma(IS_1)$. Hence, for any $\beta \in f(\Sigma(IS_1))$ and $x \in U_2$ the membership query (at node IS_2): "*if* $x \in \|\beta\|_{IS_2}$?" can be checked at node IS_1 by answering the membership query "*if* $g(x) \in \|\alpha\|_{IS_1}$?" where $\beta = f(\alpha)$.

One can extend the above approach to concept approximations because they are constructed from definable sets. Hence, if at IS_1 it is possible to check that $g(x)$ for a given $x \in U_2$ belongs to the

$\Sigma(IS_1)$-lower approximation of the concept X this means that such an object x with certainty belongs to the concept X. Moreover, if $g(x)$ for some $x \in U_2$ does not belong to the $\Sigma(IS_1)$-upper approximation of the concept X this means that such an object x with certainty belongs to the complement of the concept X.

5. Vagueness

In this section we discuss an example inspired by [2] (see pages 211–215). 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 [12] $DT_i = (U_i, A_i, d_i)$ are given where

- $U_i = \{x_1, \ldots, x_{500}\}$
- $A_i = \{h\}$ where $V_h = \{1, \ldots, 220\}$
- $V_{d_i} = \{S, M, T\}$.

Hence, in each decision table we consider 500 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, e.g., S and M). In the consequence the concepts S, M, T can not be defined exactly by means of the attribute h. Using h one can define lower and upper approximations of such concepts [12]. 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, e.g., 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 [12].
- 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 [7].
- 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)

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
...	...	...	...

Table 5. Exemplary decision table DT_3 after discretization of DT_1

$X_i \times Y_j$	number of objects	D_X	D_Y	$TALLER$
$X_1 \times Y_1$	2500	S	S	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
...	...	...	...	...

Table 6. Exemplary decision table for $TALLER$ approximations

- $S \approx M$ (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 5 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 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 5

From Table 5 one can derive patterns for approximations of the concept $TALLER$. For example, the following patterns are in the lower approximation of $TALLER$: $D_X = S \wedge D_Y = S$, $D_X = S > M \wedge D_Y = S < M$, 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.

Conclusions

This paper presents a first step towards approximate reasoning in nets of information systems. The approach is based on relationships between information flow and rough set approaches. In the presented approach, information about whole objects in a given information system is linked by means of infomorphisms with information about their exact parts. We plan to extend an approach to the case of parts to a degree [18] and to use the rough mereological approach in investigations of information nets. Such an approach has close connections with information granule calculi in distributed systems.

References

[1] Barr M.: *Autonomous Categories*, Lecture Notes in Mathematics **752**, Springer-Verlag, Heidelberg 1979.

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

[3] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software, London, Addison-Wesley, 1995.

[4] Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Combining rough and crisp knowledge in deductive databases (to appear in [13]).

[5] Huhns, M.N., Singh, M.P. (Eds.): *Readings in Agents*. Morgan Kaufmann, San Mateo, 1998.

[6] WEB page for Chu spaces http://chu.stanford.edu/

[7] Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: *Rough Sets: A Tutorial.* In [8], 3–98, 1999.

[8] Pal, S.K., Skowron, A. (Eds.): *Rough-Fuzzy Hybridization A New Trend in Decision Making*, Springer–Verlag, Singapore, 1999.

[9] Pawlak, Z.: *Mathematical Foundations of Information Retrieval*, Proceedings of Symposium of Mathematical Foundations of Computer Science, September 3-8, 1973, High Tartras, 135–136; see also: *Mathematical Foundations of Information Retrieval*, CC PAS Report **101**, 1973.

[10] Pawlak, Z.: *Information Systems*, ICS PAS Report **338**, 1979.

[11] Pawlak, Z.: *Information Systems: Theoretical Foundations*, WNT Warsaw, 1983 (in Polish).

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

[13] Pal, S.K., Polkowski, L., Skowron, A. (Eds.): *Rough-Neuro Computing: Techniques for Computing with Words*. Springer-Verlag, Berlin, 2003. (to appear).

[14] Pal, S.K., Peters, J.F., Polkowski, L., Skowron, A., Rough-Neuro Computing: An Introduction, in: [13], 16–43.

[15] Peters, J.F., Skowron, A., Stepaniuk, J., Ramanna, S.: Towards an Ontology of Approximate Reason, *Fundamenta Informaticae* **51**(1-2), 2002, 157–173.

[16] Polkowski, L., Skowron, A. (Eds.): *Rough Sets in Knowledge Discovery* **1-2**, Studies in Fuzziness and Soft Computing **18-19**, Physica-Verlag, Heidelberg, 1998.

[17] Polkowski, L., Skowron, A.: Rough Mereological Foundations for Design, Analysis, Synthesis, and Control in Distributed Systems, *Information Sciences An International Journal* **104**(1-2), 1998, 129–156.

[18] Polkowski, L., Skowron, A.: Towards Adaptive Calculus of Granules, In: [24] **1**, 201–227.

[19] Pratt, V.R.: Types as Processes, via Chu Spaces, *Electronic Notes in Theoretical Computer Science* **7**, 1997, 1–21, http://www.elsevier.nl/locate/volume7.html.

[20] Skowron, A.: Rough sets in KDD. in: Z. Shi, B. Faltings, and M. Musem (Eds.), *16-th World Computer Congress (IFIP'2000): Proceedings of Conference on Intelligent Information Processing (IIP'2000)*, Publishing House of Electronic Industry, Beijing, 2000, 1–17.

[21] Skowron, A.: Toward Intelligent Systems: Calculi of Information Granules. *Bulletin of the International Rough Set Society* **5**(1-2), 2001, 9–30.

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

[23] Skowron, A., Stepaniuk, J.: Information Granules and Rough-Neuro Computing. (to appear in [13]).

[24] Zadeh, L.A., Kacprzyk, J. (Eds.): Computing with Words in Information/Intelligent Systems **1-2**, *Studies in Fuzziness and Soft Computing* **30**, Physica–Verlag, Heidelberg, 1999.

[25] Zadeh, L.A.: A New Direction in AI: Toward a Computational Theory of Perceptions, *AI Magazine* **22**(1), 2001, 73–84.