Bull. Kyushu Inst. Tech.
Pure Appl. Math. No. 60, 2013, pp. 1–29

ROUGH NON-DETERMINISTIC INFORMATION ANALYSIS AND ITS SOFTWARE TOOL: AN OVERVIEW*

Mao Wu, Naoto Yamaguchi and Hiroshi Sakai

Abstract

Rough Non-deterministic Information Analysis (*RNIA*) is a rough sets-based framework for handling tables with exact and inexact data. Under this framework, we investigated *possible equivalence relations*, *data dependencies*, *rule generation*, *rule stability*, *question-answering systems*, as well as *missing* and *interval values* as special cases of non-deterministic values. In this paper, we briefly survey *RNIA*, and report the state of its underlying software implementation.

1. Introduction

Rough set theory offers a mathematical approach to vagueness and uncertainty, and the rough sets-based concepts have been recognized to be very useful [19, 28, 34–38, 40, 41, 70, 74]. This theory usually handles tables with deterministic information, which we call *Deterministic Information Systems* (*DISs*). Many applications of this theory to classification analysis [3, 23, 42, 63, 72], data mining [11, 26], reduction [8, 18, 20, 65], rule generation [4, 12, 14, 64, 66, 71], machine learning [6] and incomplete and non-deterministic information systems [9, 15, 21, 22, 24, 25, 27, 29–33, 68, 69] have been investigated.

Non-deterministic Information Systems (*NISs*) and *Incomplete Information Systems* (*IISs*) have been proposed for handling information incompleteness in *DISs* [24, 25, 32, 33]. *NISs* have been recognized to be the most important framework for handling information incompleteness in tables, and several theoretical works have been reported [9, 15, 21, 22, 24, 25, 27, 29–33, 68, 69]. We follow this robust framework, and we have been developing algorithms and a software tool, which can handle rough sets-based concepts in *NISs*. We are simply calling this work *Rough Non-deterministic Information Analysis* (*RNIA*) [43–62].

RNIA is a framework for discrete data analysis, which will take the complementary role in statistical data analysis. In *RNIA*, there is no concept of *mean* nor *variance*, but there exists the concept of *consistency*. In statistical analysis, we may obtain a regression line from data sets, and the role of an association rule in *RNIA* corresponds

---

Key words and phrases. Table data analysis, Rough sets, Data dependencies, Data mining, Rules, Non-deterministic information, Implementation, Prolog.

* The third author is supported by the Grant-in-Aid for Scientific Research (C) (No. 22500204), Japan Society for the Promotion of Science.

to a regression line. We know the tendency and the property of each data set by using regression lines and association rules. *RNIA* is a new attempt to analyze data sets in addition to statistical data analysis.

In this paper, we survey our previous work, and show our implemented software tool on *RNIA*. This paper is organized as follows: Section 2 recalls the foundations of rough sets in *DISs*. Section 3 and 4 introduce the frame work of *RNIA*, and survey several extended concepts from *DISs* to *NISs*. In Section 5, we describe our implemented software tool. Section 6 concludes this survey.

2. Foundations of rough sets in DISs

This section recalls the foundations of rough sets in *DISs*.

2.1. Some definitions and concepts in DISs

A *Deterministic Information System* (*DIS*) ψ is a quadruplet [35, 37]

$$\psi = (OB, AT, \{VAL_A \mid A \in AT\}, f),$$

where OB is a finite set whose elements are called *objects*, AT is a finite set whose elements are called *attributes*, VAL_A is a finite set whose elements are called *attribute values* and f is such a mapping:

$$f : OB \times AT \to \bigcup_{A \in AT} VAL_A.$$

We usually consider a table instead of this quadruplet ψ. A *DIS* ψ_1 in Table 1 is an exemplary deterministic information system. We employ it for showing each concept.

In each ψ and a subset $ATR \subseteq AT$, we employ a notation $ATR = \{A_1, \ldots, A_n\}$. Each index i at A_i is the tentative ordinal number in a set ATR, and is not the ordinal number in the original data set. For a set of attributes ATR and an object x, $(f(x, A_1), \ldots, f(x, A_n))$ is a *tuple* of x.

Table 1. An exemplary *DIS* ψ_1 for the suitcase data set. Here, $VAL_{Color} = \{red, blue, green\}$, $VAL_{Size} = \{small, medium, large\}$, $VAL_{Weight} = \{light, heavy\}$, $VAL_{Price} = \{high, low\}$.

Object	*Color*	*Size*	*Weight*	*Price*
x_1	*red*	*small*	*light*	*low*
x_2	*red*	*medium*	*light*	*high*
x_3	*blue*	*medium*	*light*	*high*
x_4	*red*	*medium*	*heavy*	*low*
x_5	*red*	*large*	*heavy*	*high*
x_6	*blue*	*large*	*heavy*	*high*

If $f(x, A_i) = f(y, A_i)$ holds for every $A_i \in ATR \subseteq AT$, we see there is a relation between x and y for ATR. This relation is an *equivalence relation* over OB [37]. Let $eq(ATR)$ denote a set of the equivalence classes with respect to ATR, and let $[x]_{ATR} \in eq(ATR)$ denote an *equivalence class* below:

$$[x]_{ATR} = \{y \in OB \mid f(y, A_i) = f(x, A_i) \text{ for every } A_i \in ATR\}.$$

In rough sets, we effectively employ equivalence classes.

According to ψ_1, let us consider four cases (A), (B), (C) and (D) of ATR.

(A) For $ATR = \{Size, Weight\}$, a tuple of x_1 is $(small, light)$. $eq(\{Size, Weight\}) = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}, \{x_5, x_6\}\}$, $[x_1]_{\{Size,\, Weight\}} = \{x_1\}$, $[x_2]_{\{Size,\, Weight\}} = \{x_2, x_3\}$.

(B) For $ATR = \{Color, Size, Weight\}$, a tuple of x_1 is $(red, small, light)$, $eq(\{Color, Size, Weight\}) = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}\}$, $[x_i]_{\{Color,\, Size,\, Weight\}} = \{x_i\}$ $(i = 1, 2, \ldots, 6)$.

(C) For $ATR = \{Weight\}$, a tuple of x_1 is $(light)$, $eq(\{Weight\}) = \{\{x_1, x_2, x_3\}, \{x_4, x_5, x_6\}\}$, $[x_1]_{\{Weight\}} = \{x_1, x_2, x_3\}$.

(D) For $ATR = \{Price\}$, a tuple of x_1 is (low), $eq(\{Price\}) = \{\{x_1, x_4\}, \{x_2, x_3, x_5, x_6\}\}$, $[x_1]_{\{Price\}} = \{x_1, x_4\}$.

Now, let us sequentially consider five rough sets-based concepts by using the above four cases (A), (B), (C) and (D).

(Concept 1) *The Definability of a Set in a DIS* ψ: If a set $X \subseteq OB$ is the union of some equivalence classes in $eq(ATR)$, we say X is *definable* (for ATR) in ψ. Otherwise, we say X is *rough* (for ATR) in ψ.

In case (B), any set $X \subseteq OB$ is definable for $ATR = \{Color, Size, Weight\}$, because $X = \bigcup_{x \in X} [x]_{\{Color,\, Size,\, Weight\}}$ holds. However, $X = \{x_1, x_2\}$ is not definable for $ATR = \{Size, Weight\}$ in case (A). Both in cases neither (C) nor (D), $X = \{x_1, x_2\}$ is not definable.

(Concept 2) *The Consistency of an Object:* Let us consider two disjoint sets $CON \subseteq AT$ which we call *condition attributes* and $DEC \subseteq AT$ which we call *decision attributes*. An object $x \in OB$ is *consistent*, if $f(x, A) = f(y, A)$ holds for every $A \in CON$ implies $f(x, A) = f(y, A)$ holds for every $A \in DEC$.

Let CON be $\{Weight\}$ and DEC be $\{Price\}$. Then, $[x_1]_{\{Weight\}} = \{x_1, x_2, x_3\}$ holds, but $f(x_1, Price) \neq f(x_3, Price)$ holds. Thus, object x_1 is not consistent. Similarly, all 6 objects are not consistent.

Rough set theory makes use of equivalence classes for solving problems. Here, let us show the most important proposition, which connects two equivalence classes $[x]_{CON}$ and $[x]_{DEC}$ with the consistency of an object x.

Proposition 1 [37]. *For each DIS,* (1) *and* (2) *in the following are equivalent.*

(1) *An object* $x \in OB$ *is consistent for* CON *and* DEC.

(2) $[x]_{CON} \subseteq [x]_{DEC}$.

(Concept 3) *The Degree of Dependency:* The *degree of dependency* for CON and DEC is a ratio,

$$deg(CON, DEC) = |\{x \in OB \mid x \text{ is consistent for } CON \text{ and } DEC\}|/|OB|.$$

Clearly, $deg(CON, DEC) = 1.0$ holds if and only if every object $x \in OB$ is consistent. For $CON = \{Weight\}$ and $DEC = \{Price\}$ in ψ_1, any object is not consistent. Therefore,

$$deg(\{Weight\}, \{Price\}) = 0/6 = 0.0.$$

For $CON = \{Color, Size, Weight\}$ and $DEC = \{Price\}$ in ψ_1, any object is consistent. Therefore,

$$deg(\{Color, Size, Weight\}, \{Price\}) = 6/6 = 1.0.$$

(Concept 4) *Reduction of Condition Attributes:* Let us consider a consistent object x for CON and DEC. An attribute $A \in CON$ is *dispensable* in CON, if x is also consistent for $CON \setminus \{A\}$.

Object x_1 is consistent for $CON = \{Color, Size, Weight\}$ and $DEC = \{Price\}$. However, x_1 is also consistent for $CON = \{Size\}$. Namely, both *Color* and *Weight* are dispensable for x_1.

(Concept 5) *Rules and Criteria (Support and Accuracy):* For any object $x \in OB$, let τ^x denote a formula called an *implication* related to CON and DEC.

$$\tau^x : \bigwedge_{A \in CON}[A, f(x, A)] \Rightarrow \bigwedge_{A \in DEC}[A, f(x, A)],$$

where a formula $[A, f(x, A)]$ implies that $f(x, A)$ is the value of the attribute A. This is called a *descriptor* [19, 24, 35, 63]. In most of work on rule generation, a rule is defined by an implication τ^x satisfying some constraints. A constraint, such that $deg(CON, DEC) = 1.0$ holds for CON and DEC, has been proposed in [35–37].

In ψ_1, we know the following implication is a rule,

$$[Size, small] \wedge [Weight, light] \Rightarrow [Price, low],$$

because $deg(\{Size, Weight\}, \{Price\}) = 1.0$. Furthermore, we apply the reduction to this implication, and have the following *minimal* rule from object x_1,

$$[Size, small] \Rightarrow [Price, low].$$

Another familiar constraint [1, 12, 15, 36–38, 70, 74] is defined by two values in the following:

$$support(\tau^x) = |[x]_{CON} \cap [x]_{DEC}|/|OB|,$$

$$accuracy(\tau^x) = |[x]_{CON} \cap [x]_{DEC}|/|[x]_{CON}|.$$

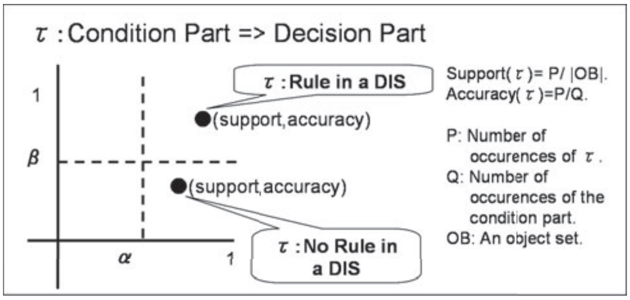

Fig. 1. A pair (*support*, *accuracy*) corresponding to the implication τ.

Since $[x]_{CON}$, $[x]_{DEC}$ and $[x]_{CON} \cap [x]_{DEC}$ are also equivalence classes for attributes CON, DEC and $CON \cup DEC$, the following holds.

$$support(\tau^y) = support(\tau^x), \qquad accuracy(\tau^y) = accuracy(\tau^x)$$

for any $y \in [x]_{CON} \cap [x]_{DEC}$.

Therefore, we may handle τ instead of τ^x in each ψ. However in *NISs*, this property is not assured. Here, we clarify two standard rule generation tasks.

Definition 1 (*Specification of rule generation tasks in a DIS*). *For threshold values α and β $(0 < \alpha, \beta \leq 1)$, find each implication τ satisfying $support(\tau) \geq \alpha$ and $accuracy(\tau) \geq \beta$. We say this is criterion-based rule generation in a DIS. Especially, if $\beta = 1.0$, we say this is consistency-based rule generation in a DIS.*

The *Apriori* algorithm [1, 2] proposed to search for such criterion-based rules by Agrawal is now one of the most representative methods in data mining [5]. As for the consistency-based rule generation, a *discernibility function* method [63] by Skowron is known well.

3. Foundations of rough non-deterministic information analysis

This section surveys a framework of *RNIA* (*Rough Non-deterministic Information Analysis*) and *possible equivalence relations* in *NISs*.

3.1. Some definitions and concepts in NISs

A *Non-deterministic Information System* (*NIS*) Φ is also a quadruplet [32, 36, 37]

$$\Phi = (OB, AT, \{VAL_A \mid A \in AT\}, g),$$

$$g : OB \times AT \to P\left(\bigcup_{A \in AT} VAL_A\right) \qquad \text{(a power set of } \bigcup_{A \in AT} VAL_A\text{)}.$$

Every set $g(x, A)$ is interpreted as that there is an actual value in this set but this value is not known [32, 36, 37]. Especially if the real value is not known at all, $g(x, A)$ is equal to VAL_A. This is called the *null value* interpretation [7] or missing value [15, 22, 68]. We usually consider a table instead of this quadruplet Φ. Let us consider an exemplary NIS Φ_1 in Table 2.

In Φ_1, $g(x_1, Color) = VAL_{Color}$ holds, and this means there is no information about this attribute value, namely we identify Φ_1 with Table 3.

Table 2. An exemplary NIS Φ_1 for the suitcase data set. Here, $VAL_{Color} = \{red, blue, green\}$, $VAL_{Size} = \{small, medium, large\}$, $VAL_{Weight} = \{light, heavy\}$, $VAL_{Price} = \{high, low\}$.

Object	*Color*	*Size*	*Weight*	*Price*
x_1	$\{red, blue, green\}$	$\{small\}$	$\{light, heavy\}$	$\{low\}$
x_2	$\{red\}$	$\{small, medium\}$	$\{light, heavy\}$	$\{high\}$
x_3	$\{red, blue\}$	$\{small, medium\}$	$\{light\}$	$\{high\}$
x_4	$\{red\}$	$\{medium\}$	$\{heavy\}$	$\{low, high\}$
x_5	$\{red\}$	$\{small, medium, large\}$	$\{heavy\}$	$\{high\}$
x_6	$\{blue, green\}$	$\{large\}$	$\{heavy\}$	$\{low, high\}$

Table 3. A table with non-deterministic information and null values. The $*$ symbol means a null value, and we identify $*$ with a set of attribute value.

Object	*Color*	*Size*	*Weight*	*Price*
x_1	$*$	*small*	$*$	*low*
x_2	*red*	$\{small, medium\}$	$*$	*high*
x_3	$\{red, blue\}$	$\{small, medium\}$	*light*	*high*
x_4	*red*	*medium*	*heavy*	$*$
x_5	*red*	$*$	*heavy*	*high*
x_6	$\{blue, green\}$	*large*	*heavy*	$*$

In the previous work, non-deterministic information seems to be identified with a null value. However, each $g(x_3, Color)$, $g(x_6, Color)$, $g(x_2, Size)$ and $g(x_3, Size)$ is different from neither a null value nor a missing value. We have clarified the property of non-deterministic information, and we are proposing a new framework [51, 52].

3.2. A basic chart and two modalities

Now, we introduce a *derived DIS* from a NIS, and show the basic chart in $RNIA$. Since each VAL_A $(A \in AT)$ is finite, we can generate a DIS by replacing each non-deterministic information $g(x, A)$ with an element in $g(x, A)$. We named such a DIS a *derived DIS* from a NIS, and define the following:

$$DD(\Phi) = \{\psi \mid \psi \text{ is a derived } DIS \text{ from a } NIS\ \Phi\}.$$

In Φ_1, there are 2304 $(= 3^2 \times 2^8)$ derived *DISs*, and $\psi_1 \in DD(\Phi_1)$ holds. Due to the interpretation of non-deterministic information, we see an actual ψ^{actual} exists in these 2304 derived *DISs*. Like this, we always consider the *basic chart* and the following two modalities.

(Certainty) If a formula α holds in every $\psi \in DD(\Phi)$, α also holds in ψ^{actual}. In this case, we say α *certainly holds* in ψ^{actual}.

(Possibility) If a formula α holds in some $\psi \in DD(\Phi)$, there exists such a possibility that α holds in ψ^{actual}. In this case, we say α *possibly holds* in ψ^{actual}.

Even if there exists the information incompleteness in Φ, we can have the following decision making.
(1) If a formula α certainly holds, we think α holds under the uncertainty.
(2) If a formula α possibly holds, we think α may hold under the uncertainty.
(3) Otherwise, we think α does not hold under the uncertainty.

In *RNIA*, we follow the rough sets-based concepts in *DISs*, and reconsider the certainty and the possibility of rough sets-based concepts in *NISs*.

3.3. Possible equivalence classes in NISs

For a *NIS*, we call an equivalence relation in a derived *DIS* a *possible equivalence relation* (*pe-relation*) in a *NIS*. A *pe*-relation defines a set $peq(ATR)$ of all *possible equivalence classes* (*pe-class*) in a *NIS*. For example in Fig. 2, we obtain

$$peq(\{Color, Size\}) = \{\{1,2,3\}\} \text{ in } DIS_4,$$
$$peq(\{Color, Size\}) = \{\{1\},\{2\},\{3\}\} \text{ in } DIS_{24}.$$

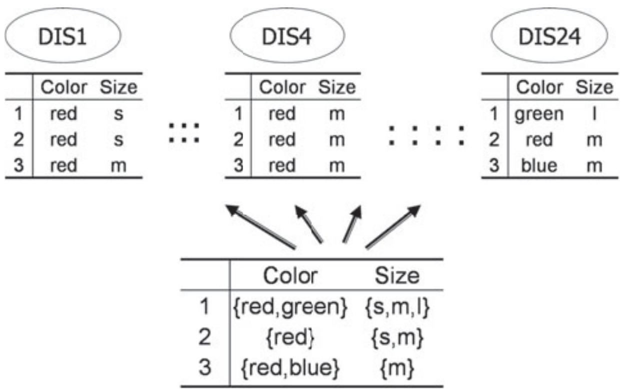

Fig. 2. An example of the basic chart for Φ_2 and a set $DD(\Phi_2)$ of 24 derived DISs.

Both classes $\{1,2,3\}$ and $\{1\}$ are *pe*-classes with object 1. It is necessary to characterize each *pe*-class for handling rough sets-based concepts in Section 2. We first define two sets *inf* and *sup* for each descriptor, which is given in (Concept 5).

Definition 2. *For a NIS with a function* $g : OB \times AT \to P(\bigcup_{A \in AT} VAL_A)$ *and a set of descriptors* $[A_i, \zeta_i]$ $(A_i \in ATR \subseteq AT)$*, we define two sets inf and sup.*
(1) *For a descriptor* $[A_i, \zeta_i]$,

$$inf([A_i, \zeta_i]) = \{x \in OB \mid g(x, A_i) = \{\zeta_i\}\},$$

$$sup([A_i, \zeta_i]) = \{x \in OB \mid \zeta_i \in g(x, A_i)\}.$$

(2) *For a compound descriptor* $[ATR, \zeta_{ATR}]$ $(= [\{A_1, \dots, A_n\}, (\zeta_1, \dots, \zeta_n)])$,

$$inf([ATR, \zeta_{ATR}]) = \{x \in OB \mid g(x, A_i) = \{\zeta_i\} \text{ for each } i\},$$

$$sup([ATR, \zeta_{ATR}]) = \{x \in OB \mid \zeta_i \in g(x, A_i) \text{ for each } i\}.$$

We can directly obtain the next proposition from Definition 2.

Proposition 2 [46, 51]. *For descriptors* $[A_i, \zeta_i]$ $(A_i \in ATR \subseteq AT)$, $ATR = \{A_1, \dots, A_n\} \subseteq AT$ *and a tuple* $\zeta_{ATR} = (\zeta_1, \dots, \zeta_n)$*, we obtain the following:*

$$inf([ATR, \zeta_{ATR}]) = \bigcap_i inf([A_i, \zeta_i]),$$

$$sup([ATR, \zeta_{ATR}]) = \bigcap_i sup([A_i, \zeta_i]).$$

According to Proposition 2, we can easily calculate two sets *inf* and *sup* for each compound descriptor $[ATR, \zeta_{ATR}]$. For example in Fig. 2, the following holds:

$$inf([Color, red]) = \{1\}, \qquad sup([Color, red]) = \{1,2,3\},$$

$$inf([Size, m]) = \{3\}, \qquad sup([Size, m]) = \{1,2,3\},$$

$$inf([\{Color, Size\}, (red, m)]) = \{1\} \cap \{3\} = \varnothing,$$

$$sup([\{Color, Size\}, (red, m)]) = \{1,2,3\} \cap \{1,2,3\} = \{1,2,3\}.$$

These *inf* and *sup* in Definition 2 and Proposition 2 are key information for *RNIA*. The set *sup* is semantically equal to a set defined by the similarity relation *SIM* [21, 22]. In [21, 22], some theorems are presented based on the relation *SIM*, and our theoretical results are closely related to those theorems. However, the set *sup* causes new properties, which hold just in *NISs*.

Now, let us consider a relation between each *pe*-class and each compound descriptor $[ATR, \zeta_{ATR}]$ $(= [\{A_1, \dots, A_n\}, (\zeta_1, \dots, \zeta_n)])$.

Definition 3. *For a NIS, a compound descriptor* $[ATR, \zeta_{ATR}]$ *and a derived DIS* ψ*, let* $pec_{[ATR, \zeta_{ATR}], \psi}$ *denote a pe-class defined by* $[ATR, \zeta_{ATR}]$ *and* ψ*.*

In Fig. 2, $pec_{[Color, red], DIS_1} = \{1, 2, 3\}$ and $pec_{[Size, s], DIS_1} = \{1, 2\}$ hold. If we see a *DIS* ψ is a *NIS* with only singleton sets,

$$pec_{[ATR, \zeta_{ATR}], \psi} = inf([ATR, \zeta_{ATR}]) = sup([ATR, \zeta_{ATR}])$$

holds. However in every *NIS*, $pec_{[ATR, \zeta_{ATR}], \psi}$ depends upon a derived *DIS* ψ, and generally

$$inf([ATR, \zeta_{ATR}]) \subseteq pec_{[ATR, \zeta_{ATR}], \psi} \subseteq sup([ATR, \zeta_{ATR}])$$

holds. Proposition 3 connects a *pe*-class $pec_{[ATR, \zeta_{ATR}], \psi}$ with $inf([ATR, \zeta_{ATR}])$ and $sup([ATR, \zeta_{ATR}])$.

PROPOSITION 3 [51, 54]. *The conditions* (1) *and* (2) *in the following are equivalent.*
(1) *X is a pe-class* $pec_{[ATR, \zeta_{ATR}], \psi}$.
(2) $inf([ATR, \zeta_{ATR}]) \subseteq X \subseteq sup([ATR, \zeta_{ATR}])$.
Namely, we can express any pe-class X by

$$X = inf([ATR, \zeta_{ATR}]) \cup M,$$
$$(M \subseteq sup([ATR, \zeta_{ATR}]) \backslash inf([ATR, \zeta_{ATR}])).$$

3.4. Computational complexity in NISs

Here, we must pay attention to the computational complexity related to a *NIS*. For a *NIS* Φ, the number of derived *DISs* increases in the exponential order. Therefore, it will be hard to apply the *explicit method* such that we sequentially examine each concept in $\psi \in DD(\Phi)$.

In each concept, we do not employ this explicit method. In the subsequent sections, we show methods depending upon equivalence classes. Especially in rule generation, we had an algorithm which does not depend upon $|DD(\Phi)|$ at all.

4. Extended concepts from DISs to NISs

Now, we sequentially consider rough sets-based concepts in *NISs*.

4.1. The definability of a set in NISs

We can extend (Concept 1) in Section 2 to the concept of a *NIS* as follows:

(Certainly definable) A set X is *certainly definable*, if X is definable in each $\psi \in DD(\Phi)$.

(Possibly definable) A set X is *possibly definable*, if X is definable in some $\psi \in DD(\Phi)$.

Let $PT(x, ATR)$ be a set of tuples for an object x and $ATR \subseteq AT$. For example,

$$PT(1, \{Color\}) = \{(red), (green)\},$$
$$PT(2, \{Color, Size\}) = \{(red, s), (red, m)\}.$$

Here, $X = \bigcup_{x \in X}\{x\}$ and $x \in [x]_{AT} = pec_{[AT, \zeta_{AT}], \psi}$ ($\zeta_{AT} \in PT(x, AT)$) hold according to Proposition 3, therefore we clearly conclude $X \subseteq \bigcup_{x \in X}[x]_{AT}$. On the other hand, if each x and each $\zeta_{AT} \in PT(x, AT)$ satisfies $sup([AT, \zeta_{AT}]) \subseteq X$, we conclude $[x]_{AT} \subseteq X$ in every $\psi \in DD(\Phi)$, because

$$[x]_{AT} \subseteq sup([AT, \zeta_{AT}]) \subseteq X.$$

In this case, we have the following:

$$\bigcup_{x \in X}[x]_{AT} \subseteq \bigcup_{x \in X} X = X.$$

Therefore, we conclude $X = \bigcup_{x \in X}[X]_{AT}$ in every $\psi \in DD(\Phi)$. This means X is certainly definable. As for the possibility, we employ $inf([AT, \zeta_{AT,x}])$ instead of $sup([AT, \zeta_{AT}])$, and we have Proposition 4.

Proposition 4 [51].
(1) *X is certainly definable for $ATR \subseteq AT$, if and only if $sup([ATR, \zeta_{ATR}]) \subseteq X$ for each $x \in X$ and each $\zeta_{ATR} \in PT(x, AT)$.*
(2) *X is possibly definable for $ATR \subseteq AT$, if and only if $inf([ATR, \zeta_{ATR}]) \subseteq X$ for each $x \in X$ and a tuple $\zeta_{ATR} \in PT(x, AT)$.*

4.2. The consistency of an object in NISs

We can extend (Concept 2) in Section 2 to the concept of a *NIS*. Let CON be a set of condition attributes and DEC be a set of decision attributes.

(Certainly consistent) An object x is *certainly consistent* in a *NIS*, if x is consistent for CON and DEC in each $\psi \in DD(\Phi)$.

(Possibly consistent) An object x is *possibly consistent* in a *NIS*, if x is consistent for CON and DEC in some $\psi \in DD(\Phi)$.

We can also characterize the above modalities by using *inf* and *sup*. This is an extension of Proposition 1 in *DISs* to *NISs*.

Proposition 5 [51]. *Let us suppose an object x and its tuples $PT(x, CON)$, $PT(x, DEC)$.*
(1) *An object x is certainly consistent, if and only if $sup([CON, \zeta_{CON}]) \subseteq inf([DEC, \eta_{DEC}])$ holds for each $\zeta_{CON} \in PT(x, CON)$ and each $\eta_{DEC} \in PT(x, DEC)$.*
(2) *An object x is possibly consistent, if and only if $inf([CON, \zeta_{CON}]) \subseteq sup([DEC, \eta_{DEC}])$ holds for some $\zeta_{CON} \in PT(x, CON)$ and some $\eta_{DEC} \in PT(x, DEC)$.*

4.3. Data dependency in NISs

As for the data dependency, we can extend (Concept 3) in Section 2 to the *minimum data dependency* and the *maximum data dependency* in a *NIS*.

DEFINITION 4. *In a NIS, let us consider a set of condition attributes CON and a set of decision attributes DEC. For any derived DIS* ψ*, let* $deg(CON, DEC, \psi)$ *denote the data dependency* $deg(CON, DEC)$ *in* ψ.

(1) *Let* $Min\_deg(CON, DEC)$ *be* $Min_{\psi}\{deg(CON, DEC, \psi)\}$*, and we call it the minimum degree of data dependency for CON and DEC.*

(2) *Let* $Max\_deg(CON, DEC)$ *be* $Max_{\psi}\{deg(CON, DEC, \psi)\}$*, and we call it the maximum degree of data dependency for CON and DEC.*

4.4. The minimum and the maximum of criterion values in NISs

At (Concept 5) in Section 2, we have shown criteria $support(\tau^x)$ and $accuracy(\tau^x)$ in *DISs*. This $\tau^x(= \bigwedge_{A \in CON}[A, f(x, A)] \Rightarrow \bigwedge_{A \in DEC}[A, f(x, A)])$ is an implication from an object x. In a *NIS* Φ, we usually consider $DD(\Phi)$ and each $\psi \in DD(\Phi)$. Here, the tuple of an object x in ψ' and the tuple of an object x in ψ'' may be different. Namely, τ^x in ψ' may not exist in ψ''. For example in Fig. 2, $\tau^1 : [Color, red] \Rightarrow [Size, s]$ in DIS_1 does not exist in DIS_4. If τ^x does not exist in ψ, we define $support(\tau^x) = 0.0$ and $accuracy(\tau^x) = 0.0$ in ψ. We also define

$$DD(\tau^x) = \{\psi \in DD(\Phi) \mid support(\tau^x) > 0\}.$$

Furthermore, if $DD(\tau^x) = DD(\Phi)$, we say τ^x is *definite*. Otherwise, we say τ^x is *indefinite*. In Fig. 2, there is no definite τ^x, and each τ^x is indefinite.

DEFINITION 5. *For a NIS* Φ*, each* $\psi \in DD(\Phi)$ *and an implication* τ^x*, let* $support(\tau^x, \psi)$ *and* $accuracy(\tau^x, \psi)$ *be the support and accuracy values in* ψ*. We give the following definition.*

$$minsupp(\tau^x) = Min_{\psi \in DD(\tau^x)}\{support(\tau^x, \psi)\},$$

$$maxsupp(\tau^x) = Max_{\psi \in DD(\tau^x)}\{support(\tau^x, \psi)\},$$

$$minacc(\tau^x) = Min_{\psi \in DD(\tau^x)}\{accuracy(\tau^x, \psi)\},$$

$$maxacc(\tau^x) = Max_{\psi \in DD(\tau^x)}\{accuracy(\tau^x, \psi)\}.$$

In Definition 5, we may employ $DD(\Phi)$ instead of $DD(\tau^x)$. For a definite τ^x, $DD(\Phi) = DD(\tau^x)$ holds, so we may employ either $DD(\Phi)$ or $DD(\tau^x)$. However, if τ^x is indefinite, we directly obtain $minsupp(\tau^x) = 0.0$ and $minacc(\tau^x) = 0.0$, because there is a ψ where τ^x does not appear. Even though we may employ $DD(\Phi)$, however we think that $DD(\tau^x)$ is more appropriate than $DD(\Phi)$ in Definition 5. We have obtained the formula to calculate each criterion value in Definition 5. This calculation does not depend upon $|DD(\tau^x)|$.

Proposition 6 [51, 54]. *Let us define $OUTACC$ and $INACC$ as follows:*

$$OUTACC = (sup([CON,\zeta]) \backslash inf([CON,\zeta])) \backslash inf([DEC,\eta]),$$

$$INACC = (sup([CON,\zeta]) \backslash inf([CON,\zeta])) \cap sup([DEC,\eta]).$$

If an implication $\tau^x : [CON,\zeta] \Rightarrow [DEC,\eta]$ is definite, the following holds.

$$minsupp(\tau^x) = |inf([CON,\zeta]) \cap inf([DEC,\eta])|/|OB|,$$

$$minacc(\tau^x) = \frac{|inf([CON,\zeta]) \cap inf([DEC,\eta])|}{|inf([CON,\zeta])| + |OUTACC|},$$

$$maxsupp(\tau^x) = |sup([CON,\zeta]) \cap sup([DEC,\eta])|/|OB|,$$

$$maxacc(\tau^x) = \frac{|inf([CON,\zeta]) \cap sup([DEC,\eta])| + |INACC|}{|inf([CON,\zeta])| + |INACC|}.$$

Proposition 6 shows a case of definite τ^x, and we can obtained similar formulas for indefinite τ^x. The details are in [54]. In Definition 5, each criterion value depends upon $DD(\tau^x)$, however each formula in Proposition 6 does not depend upon the size of $DD(\tau^x)$. We have also obtained the next proposition.

Proposition 7 [56, 58]. *Let us consider a NIS Φ and any τ^x.*

(1) *There is a $\psi' \in DD(\tau^x)$ such that $support(\tau^x,\psi')$ and $accuracy(\tau^x,\psi')$ are both minimums. Namely, both $minsupp(\tau^x)$ and $minacc(\tau^x)$ occur in this ψ'. We employ a notation ψ_{min} for this ψ'.*

(2) *There is a $\psi'' \in DD(\tau^x)$ such that $support(\tau^x,\psi'')$ and $accuracy(\tau^x,\psi'')$ are both maximums. Namely, both $maxsupp(\tau^x)$ and $maxacc(\tau^x)$ occur in this ψ''. We employ a notation ψ_{max} for this ψ''.*

4.5. Rule generation tasks in a NIS

In Section 2, we have surveyed two types of rule generation in *DISs*. The one is the criterion-based rule generation and the other is the consistency-based rule generation. This section focuses on rule generation in *NISs*, and proposes an extended *Apriori* algorithm named *NIS-Apriori*. A *NIS-Apriori* based rule generation is applicable to several types of rule generation.

Definition 6 (*Specification of the rule generation tasks in a NIS*). *Let us consider the threshold values α and β $(0 < \alpha, \beta \leq 1)$.*

(The lower system) *Find each implication τ such that $support(\tau^x) \geq \alpha$ and $accuracy(\tau^x) \geq \beta$ (for an object x) hold in each $\psi \in DD(\tau^x)$. We say this is a* criterion-based certain rule generation *in a NIS. Especially, if* $\beta = 1.0$, *we say this is a* consistency-based certain rule generation *in a NIS.*

(The upper system) *Find each implication* τ *such that* $support(\tau^x) \geq \alpha$ *and* $accuracy(\tau^x) \geq \beta$ *(for an object x) hold in some* $\psi \in DD(\tau^x)$. *We say this is a* criterion-based possible rule generation *in a NIS. Especially, if* $\beta = 1.0$, *we say this is a* consistency-based possible rule generation *in a NIS.*

These two systems are natural extensions from rule generation tasks in a *DIS*, and we need to see that these two systems depend upon $DD(\tau^x)$. The number of derived *DISs* increases in the exponential order. However, we can solve this problem. Namely, we apply Proposition 6 and 7, and we obtain a result illustrated by Fig. 3. Therefore, we have the next equivalent specification.

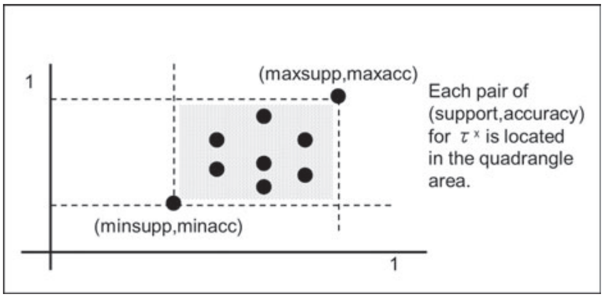

Fig. 3. A distribution of pairs (*support*, *accuracy*) for τ^x. There exists $\psi_{min} \in DD(\tau^x)$ which makes both $support(\tau^x)$ and $accuracy(\tau^x)$ the minimum. There exists $\psi_{max} \in DD(\tau^x)$ which makes both $support(\tau^x)$ and $accuracy(\tau^x)$ the maximum. We denote such quantities as *minsupp*, *minacc*, *maxsupp* and *maxacc*, respectively.

THEOREM 1 (*Equivalent specification of the rule generation tasks in a NIS*). *Let us consider the threshold values* α *and* β $(0 < \alpha, \beta \leq 1)$.

(The lower system) *Find each implication* τ *such that* $minsupp(\tau^x) \geq \alpha$ *and* $minacc(\tau^x) \geq \beta$ *for an object x (see Fig. 4).*

(The upper system) *Find each implication* τ *such that* $maxsupp(\tau^x) \geq \alpha$ *and* $maxacc(\tau^x) \geq \beta$ *for an object x (see Fig. 5).*

For implementing this equivalent specification, we take the similar method as *Apriori* algorithm. We identify an item [1, 2] with a descriptor $[A, \zeta]$. We always assign $inf([ATR, \zeta_{ATR}])$ and $sup([ATR, \zeta_{ATR}])$ to each descriptor $[ATR, \zeta_{ATR}]$ by Definition 2. Since each τ^x is a conjunction of descriptors, we sequentially generate τ^x. In the lower system, we check $minsupp(\tau^x) \geq \alpha$ and $minacc(\tau^x) \geq \beta$ in Fig. 4. In the upper system, we check $maxsupp(\tau^x) \geq \alpha$ and $maxacc(\tau^x) \geq \beta$ in Fig. 5. We are calling

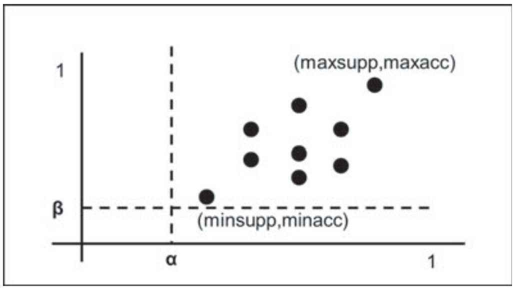

Fig. 4. A characterization of a criterion-based rule by the lower system.

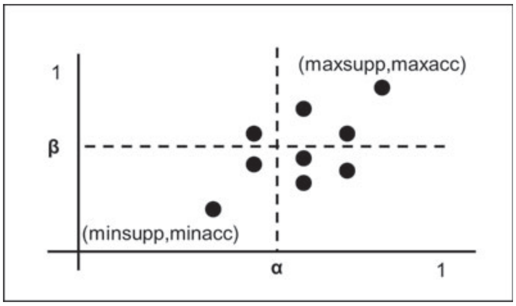

Fig. 5. A characterization of a criterion-based rule by the upper system.

the above steps *NIS-Apriori* algorithm. Clearly, *NIS-Apriori* does not depend upon $|DD(\tau^x)|$. The details are in [54, 67].

In this section, we extended each rough set-based concept in *DISs* to *NISs*. In *NISs*, each concept is extended to the certain concept and the possible concept. According to the previous research [15, 21, 22, 24, 25, 32, 33, 68, 69], we knew the necessity of handling *NISs*, and the survey in this section will be a solution for handling *NISs*.

5. A software tool for RNIA

In our research, we have coped with the following challenges described in more detail in the subsequent sections:

(A) Management of possible equivalence relations [46, 49, 51],
(B) The minimum and the maximum degrees of data dependency [47, 50, 52],
(C) Certain and possible rules, and rule generation [51, 53, 54, 57],
(D) Stability factor of rules and calculation [56, 58],
(E) Management of missing values [54, 59, 60],

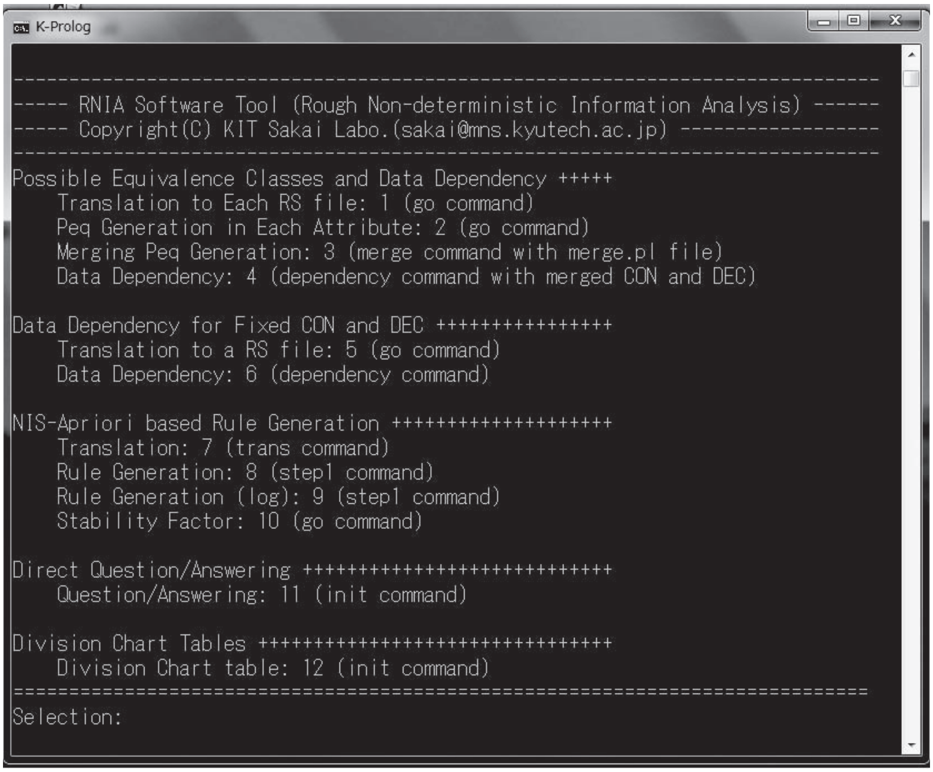

Fig. 6. The menu page of RNIA software tool.

(F) Management of an actual value by intervals [57, 59, 60],
(G) Management of numerical patterns and figures [55],
(H) Direct question-answering [62].

5.1. About a software tool

This software is implemented in C and Prolog, and Prolog program is converted to C. Then, C sources are compiled to object files. For execution, we call each object in Prolog interpreter. This is implemented on Windows 7 PC in Fig. 6. The details of this software tool are stored in [43].

5.2. An exemplary data set

In the subsequent sections, we show an actual execution on Φ_3 in Table 4. Since Φ_3 is an artificial data set, the obtained rule may not coincide with our intuitive knowledge.

The following is an actual data set of Φ_3.

Table 4. An exemplary NIS Φ_3. Here, $VAL_{Temp(erature)} = \{normal, high, very\_high\}$, $VAL_{Head(ache)} = \{yes, no\}$, $VAL_{Nausea} = \{yes, no\}$, $VAL_{Flu} = \{yes, no\}$, $DD(\Phi_3)$ consists of 4608 derived *DISs*.

OB	*Temp*	*Head*	*Nausea*	*Flu*
1	$\{high\}$	$\{yes, no\}$	$\{no\}$	$\{yes\}$
2	$\{high, very\_high\}$	$\{yes\}$	$\{yes\}$	$\{yes\}$
3	$\{normal, high, very\_high\}$	$\{no\}$	$\{no\}$	$\{yes, no\}$
4	$\{high\}$	$\{yes\}$	$\{yes, no\}$	$\{yes, no\}$
5	$\{high\}$	$\{yes, no\}$	$\{yes\}$	$\{no\}$
6	$\{normal\}$	$\{yes\}$	$\{yes, no\}$	$\{yes, no\}$
7	$\{normal\}$	$\{no\}$	$\{yes\}$	$\{no\}$
8	$\{normal, high, very\_high\}$	$\{yes\}$	$\{yes, no\}$	$\{yes\}$

```
object(8,4). /* the number of objects=8, the number of attributes=4 */
support(0.3). /* support value = 0.3 */
accuracy(0.5). /* accuarcy value = 0.5 */
decision(4). /* decision attribute is 4th */

attrib_values(1,temperature,3,[normal,high,very_high]).
attrib_values(2,headache,2,[yes,no]).
attrib_values(3,nausea,2,[yes,no]).
attrib_values(4,flu,2,[yes,no]).

data(1,[high,[yes,no],no,yes]). /* NIS table */
data(2,[[high,very_high],yes,yes,yes]).
data(3,[[normal,high,very_high],no,no,[yes,no]]).
data(4,[high,yes,[yes,no],[yes,no]]).
data(5,[high,[yes,no],yes,no]).
data(6,[normal,yes,[yes,no],[yes,no]]).
data(7,[normal,no,yes,no]).
data(8,[[normal,high,very_high],yes,[yes,no],yes]).
total_cases(4608,nointerval).
```

5.3. (A) Management of possible equivalence relations

In rough sets, we make use of equivalence relations and classes in a *DIS*. However in *NISs*, there may be several derived *DISs*, for example in Φ_3 there are 4608 derived *DISs*. Namely in Φ_3, there are less than 4608 kinds of equivalence relations.

Since the number of derived *DISs* increases exponentially, it seems inappropriate to examine each equivalence relation sequentially. We at first generate *pe*-relations in each attribute. Any *pe*-relation can be obtained as a side effect by solving the definability of a total set *OB*. We have implemented this program as a constraint satisfaction problem [51]. Then, we merge obtained *pe*-relations for generating $peq(ATR)$ ($ATR \subseteq AT$). Like this, we have reduced the computational complexity [52]. Even though, the program for examining the definability is time-consuming.

```
Selection: 2.
yes
?- go.
Original File Name: flu.pl.

Pe-relation (temperature)
[1] [[1,2,4,5],[3,6,7,8]] 1
[2] [[1,4,5],[2],[3,6,7,8]] 1
[3] [[1,2,4,5,8],[3,6,7]] 1
     :      :     :
[17] [[1,4,5,8],[2,3],[6,7]] 1
[18] [[1,4,5],[2,3,8],[6,7]] 1
POSSIBLE CASES 18

Pe-relation (headache)
[1] [[1,2,4,5,6,8],[3,7]] 1
[2] [[1,2,4,6,8],[3,5,7]] 1
[3] [[1,3,5,7],[2,4,6,8]] 1
[4] [[1,3,7],[2,4,5,6,8]] 1
POSSIBLE CASES 4

Pe-relation (nausea)
[1] [[1,3,4,6,8],[2,5,7]] 1
[2] [[1,3,4,6],[2,5,7,8]] 1
     :      :     :
[8] [[1,3],[2,4,5,6,7,8]] 1
POSSIBLE CASES 8

Pe-relation (flu)
[1] [[1,2,3,4,6,8],[5,7]] 1
[2] [[1,2,3,4,8],[5,6,7]] 1
     :      :     :
[8] [[1,2,8],[3,4,5,6,7]] 1
POSSIBLE CASES 8
```

```
EXEC_TIME = 0.0(sec)
yes
```

Now, we generate $peq(\{temperature, headache, nausea\})$. We specify the definition for merging *pe*-relations in a file, and we execute the following.

```
Selection: 3.
Merging 1.pe ...
Merging 2.pe ...
Merging 3.pe ...

EXEC_TIME = 0.000(sec)
yes
```

After this execution, a file 123.pe is stored in a folder. In reality, there are 576 derived *DISs*, but the number of different *pe*-relations is reduced to 51 cases, because some derived *DISs* have the same *pe*-relation.

In the above execution, the program of *pe*-relations is implemented in Prolog, and the execution time is in the form of $*.*$(sec). On the other hand, the program of merging *pe*-relations is implemented in C, and the execution time is in the form of $*.***$(sec).

5.4. (B) The minimum and the maximum degrees of data dependency

In a *NIS*, the degree of dependency is related to each derived *DIS*, therefore we need to consider the minimum and the maximum degrees. The actual degree of dependency is between the minimum and the maximum degrees. If the difference between two degrees is small, the actual degree may not be influenced by the information incompleteness in a *NIS*.

By merging program, we obtained $peq(\{temperature, headache, nausea\})$ consisting of 51 elements. Since there are 4 *pe*-relations on $DEC = \{flu\}$, we can calculate each degree of dependency for 4806 derived *DISs* by considering 204 ($= 51 \times 4$) combinations. We have the following.

```
Selection: 4.
File Name for Condition: 123.pe
File Name for Decision: 4.pe
--- Dependency Check ------------
CRITERION 1(Num_of_Consistent_DISs/Num_of_All_DISs)
    Number of Derived DISs: 4608
    Number of Derived Consistent DISs: 1812
    Degree of Consistent DISs: 0.393
```

```
CRITERION 2(Total_Min_and_Max_Degrees)
    Minimum Degree of Dependency: 0.125
    Maximum Degree of Dependency: 1.000
--- Consistency Ratio for Every Object ---
Consistent ratio of the object 1: 0.792(= 3648 / 4608)
Consistent ratio of the object 2: 0.688(= 3168 / 4608)
Consistent ratio of the object 3: 0.917(= 4224 / 4608)
Consistent ratio of the object 4: 0.583(= 2688 / 4608)
Consistent ratio of the object 5: 0.656(= 3024 / 4608)
Consistent ratio of the object 6: 0.917(= 4224 / 4608)
Consistent ratio of the object 7: 1.000(= 4608 / 4608)
Consistent ratio of the object 8: 0.771(= 3552 / 4608)

EXEC_TIME = 0.000(sec)
yes
```

According to the above execution, we know there are 1812 derived *DISs*, where each object is consistent. Object 7 is consistent in each $\psi \in DD(\Phi_3)$, and object 4 is consistent in about 58% of all derived *DISs*. Since the difference between the maximum degree and the minimum degree is large, it seems difficult to decide that there is a dependency from $\{temperature, headache, nauses\}$ to $\{flu\}$ in this Φ_3.

5.5. (C) Certain and possible rules, and rule generation

We apply Theorem 1 to *Apriori* algorithm [1], and proposed *NIS-apriori* algorithm [54]. Since $minsupp(\tau^x), \ldots, maxacc(\tau^x)$ do not depend upon the number of derived *DISs*, the computational complexity of *NIS*-apriori is almost the same as the original *Apriori*. The following is real execution report for Φ_3 (decision attribute: *flu*). The threshold values are fixed to $\alpha = 0.3$ and $\beta = 0.5$, namely the following is criterion-based rule generation in Definition 6.

```
File = [tflu|pl] Support= 0.3, Accuracy = 0.5
--- 1st STEP --------------
===== Lower System ============================
The Rest Candidates: []
(Lower System Terminated)
===== Upper System ============================
[1] [temperature,normal]=>[flu,yes] (0.375, 0.75)
Objects: [3,6,8]
[2] [temperature,normal]=>[flu,no] (0.375, 1.0)
Objects: [3,6,7]
      :      :     :
```

```
[13] [nausea,no]=>[flu,yes] (0.625, 1.0)
Objects: [1,3,4,6,8]
[14] [nausea,no]=>[flu,no] (0.375, 0.75)
Objects: [3,4,6]
The Rest Candidates: []
(Upper System Terminated)

EXEC_TIME = 0.0(sec)
yes
```

In this execution, we know there are no certain rule and 12 possible rules. (The assigned number means the ordinal number for each implication, and some of them may not satisfy the constraint. In this case, 6th and 9th implications do not satisfy constraint.) If we change the value of *support* and *accuracy*, we have other implications. Here, we fix $\alpha = 0.0$ and $\beta = 1.0$, namely the following is consistency-based rule generation in Definition 6.

```
File = [tflu|pl] Support= 0.0, Accuracy = 1.0
--- 1st STEP --------------
===== Lower System ============================
The Rest Candidates: [[[1,1],[4,1]],[[1,1],[4,2]], :::
(Next Candidates are Remained)
===== Upper System ============================
[2] [temperature,normal]=>[flu,no] (0.375, 1.0)
Objects: [3,6,7]
[5] [temperature,very_high]=>[flu,yes] (0.375, 1.0)
Objects: [2,3,8]
[6] [temperature,very_high]=>[flu,no] (0.125, 1.0)
Objects: [3]
     :      :     :
[13] [nausea,no]=>[flu,yes] (0.625, 1.0)
Objects: [1,3,4,6,8]
The Rest Candidates: [[[1,1],[4,1]],[[1,2],[4,1]], :::
(Next Candidates are Remained)

EXEC_TIME = 0.0(sec)

--- 2nd STEP --------------
===== Lower System ============================
[30] [headache,no]&[nausea,yes]=>[flu,no] (0.125, 1.0)
Objects: [7]
The Rest Candidates: [[[1,1],[2,1],[4,1]],[[1,1],[2,1],[4,2]], :::
```

```
(Next Candidates are Remained)
===== Upper System ===========================
[1] [temperature,normal]&[headache,yes]=>[flu,yes] (0.25, 1.0)
Objects: [6,8]
[2] [temperature,normal]&[headache,yes]=>[flu,no] (0.125, 1.0)
Objects: [6]
[21] [headache,no]&[nausea,no]=>[flu,yes] (0.25, 1.0)
Objects: [1,3]
[22] [headache,no]&[nausea,no]=>[flu,no] (0.125, 1.0)
Objects: [3]
The Rest Candidates: [[[1,1],[2,2],[4,1]],[[1,1],[3,1],[4,1]], :::
(Next Candidates are Remained)

EXEC_TIME = 0.0(sec)
yes
```

In the above execution, an implication $[headache, no] \wedge [nausea, yes] \Rightarrow [flu, no]$ obtained in the 2nd step lower system is consistent in each of 4608 derived *DISs*.

5.6. (D) Stability factor of rules and its calculation

The lower system detects implications which satisfy *support* and *accuracy* constraint in each $\psi \in DD(\Phi)$. On the other hand, the upper system detects implications which satisfy *support* and *accuracy* constraint in a $\psi \in DD(\Phi)$. Namely, if an implication τ^x (for an $x \in OB$) satisfies constraint in a derived *DIS*, this τ is a possible rule. Even though τ^x satisfies constraint in most of derived *DIS*, this τ is also a possible rule. In order to discriminate such possible rules, we have introduced a degree below:

$$SF(\tau, \Phi) = |\{\psi \in \Phi \mid \tau \text{ satisfies constraint}\}| / |\{\psi \in \Phi \mid \tau \text{ appears in } \psi\}|$$

We name this degree *Stability factor* for τ [58]. For example in Φ_3, let us consider a certain rule in $support = 0.0$ and $accuracy = 1.0$ below:

$[headache, no] \wedge [nausea, yes] \Rightarrow [flu, no]$.

The following is actual execution. Since certain rule is consistent in each derived *DIS*, the stability factor for a certain rule is always 1.0.

```
?- sf([[headache,no],[nausea,yes]],[flu,no]).
[1] PE_CON:[7], PE_DEC:[5,7], Intersection:[7]
Possible Combination:1, Number of DISs in This Case:1
Condition_SUPPORT:0.0, Current_SUPPORT:0.125
Condition_ACCURACY:1.0, Current_ACCURACY:1.0
```

```
both, DENO=1, NUME=1
      :      :     :
[16] PE_CON:[7,5], PE_DEC:[5,7,3,4,6], Intersection:[7,5]
Possible Combination:16, Number of DISs in This Case:1
Condition_SUPPORT:0.0, Current_SUPPORT:0.25
Condition_ACCURACY:1.0, Current_ACCURACY:1.0
both, DENO=16, NUME=16

SF = 1.0 =(16/16)
EXEC_TIME = 1.0(sec)
yes
```

As for possible rules in $support = 0.0$ and $accuracy = 1.0$,
$[temperature, normal] \Rightarrow [flu, no]$ $(*)$,
$[temperature, very\_high] \Rightarrow [flu, yes]$ $(**)$,
we conclude that $(**)$ will be more reliable than $(*)$ according to the following execution.

```
?- sf([[temperature,normal]],[flu,no]).
      :      :     :
SF = 0.2777777778 =(20/72)
EXEC_TIME = 0.0(sec)
yes

?- sf([[temperature,very_high]],[flu,yes]).
      :      :     :
SF = 0.8461538462 =(88/104)
EXEC_TIME = 1.0(sec)
yes
```

5.7. (E) Management of missing values

There are several important directions of research on *DISs* with missing values or *Incomplete Information Systems*. For example, *LERS* system [13, 15] by Grzymała-Busse and a framework of reduction-based rule generation [22] by Kryszkiewicz are well known. In both cases, some interpretations are assumed for missing values, and rule extraction methods are investigated.

In [43], we are showing the execution logs. In *Mammographic* data set in [10], there are 960 objects and 6 attributes (*assessment*, *age*, *shape*, *margin*, *density* and *severity*). The decision attribute is *severity*, and its attribute values are *benign* (0) and *malignant* (1). There are 2, 5, 31, 48 and 67 missing values (? is employed to denote them) on 5 remaining attributes, respectively. Since each set of attribute values is discrete and finite, we can convert this data set to a *NIS* by replacing each ? with a set of attribute values. The number of derived *DISs* is more than 10 power 90, but *NIS-Apriori* could handle such data sets easily [43].

Generally, a *NIS-Apriori* rule generator is also applicable to *DISs* with missing values. In most of tables with categorical data, each domain of attribute values is a finite set. Since any missing value is an element of this finite domain, we replace each missing value with this domain. Then, we can apply our rule generator to such an adjusted *NIS*. In our framework, the interpretation of missing values seems clear, but instead we needed to face the problem of exponential order of the number of derived *DISs*. We have solved this exponential order problem successfully in the *NIS-Apriori* algorithm.

5.8. (F) Management of an actual value by using intervals

We see an interval $[lower, upper]$ [16, 17] takes the role of non-deterministic information in numerical values. Namely, we see an actual value val^{actual} satisfies $lower \leq val^{actual} \leq upper$. By using this consideration, we can handle the information incompleteness in numerical values.

However, we have a problem for handling numerical attribute values. Namely, the concept defined in Fig. 2 is vague. A set of real numbers is infinite and uncountable. It is necessary to control the figure in a numerical value. We introduced the concept of *resolution* γ (>0) into numerical attributes. An interval $[lower, upper]$ is *definite*, if $(upper - lower) \leq \gamma$. Otherwise, we may have infinite number of derived interval $[lower', upper']$ ($lower \leq lower'$, $upper' \leq upper$ and $(upper' - lower') = \gamma$). By using resolutions, we can have a chart similar to Fig. 2 for numerical values [57], but we have another problem. For each discrete set of attribute values VAL_A, we can naturally define a descriptor $[A, val]$ ($val \in VAL_A$). In a set of numerical attribute values, the definition of descriptors is vague. Even though we are currently specifying descriptors for numerical values, we need to consider what are the proper descriptors in a set of numerical attribute values.

In [43], we are showing an execution log for an exemplary data set, which consists of non-deterministic information and intervals.

5.9. (G) Management of numerical patterns and figures

Now, we consider information incompleteness for numerical values, again. Information incompleteness is a relative concept. For example, let us consider number π. The value 3.14 will be enough for students, but it will be too simple for researcher. This example will also be related to granularity and granular computing [39, 73].

We introduced two symbols @ and #, which represent numeric from 0 to 9. A *numerical pattern* is a sequence of @ and #, for example @@@, @@#, @##, @@.@

and @#.#. Here, '.' denotes a decimal point, and @ does not occur after #. We see @@@, @@#, @## and ### have the same type ???. Three patterns @@.@, @@.# @#.# have the same type ??.?, too. Here, @ denotes a significant figure and # denotes a figure, which we do not care.

For example, students are seeing π by a numerical pattern @.@@##..., and researchers must be seeing π by a numerical pattern @.@@@@.... Furthermore in baseball games, we often see a season batting average higher than .300 is considered to be excellent player. In this case, we are seeing .300 by a numerical pattern .@##. If we see two players' averages 0.309 and 0.310 by .@##, these two players belong to the same equivalence class. However, if we employ a numerical pattern .@@#, the two players belong to the different equivalence class.

If we employ a fine numerical pattern (with much @ symbols), we obtain the large number of equivalence classes. On the other hand, if we employ a coarse numerical pattern (with less @ symbols), we obtain the small number of equivalence classes. Namely, numerical patterns control the size of equivalence classes, *support* and *accuracy* values. In [55], we coped with numerical patterns, and implemented a software tool.

5.10. (H) Direct question-answering

If the condition $\bigwedge_i [A_i, val_i]$ matches with the condition part of an obtained certain or possible rule $\bigwedge_i [A_i, val_i] \Rightarrow [DEC, val_j]$, we have a decision $[DEC, val_j]$ with certainty or possibility.

However, if the condition $\bigwedge_i [A_i, val_i]$ does not match with the condition part of any obtained rules, we may not have decision from the data set, because $\bigwedge_i [A_i, val_i]$ may not conclude unique decision attribute value. In such case, we apply direct question-answering, and we know all $[DEC, val_j]$ with $minsupp(\tau_j)$, $minacc(\tau_j)$, $maxsupp(\tau_j)$ and $maxacc(\tau_j)$ which characterize the validity of val_j. Direct question-answering can provide all information for decision making in such case. The following is the actual execution for Φ_3.

```
?- qa([[temperature,very_high],[headache,yes]]).
----- Direct Question/Answering Mode --------
[1] [temperature,very_high]&[headache,yes]=>[flu,yes]
MINSUPP=0.0, MINACC=0.0
MAXSUPP=0.25, MAXACC=1.0
[2] [temperature,very_high]&[headache,yes]=>[flu,no]
MINSUPP=0.0, MINACC=0.0
MAXSUPP=0.0, MAXACC=0.0

EXEC_TIME = 0.0(sec)
yes
```

For condition $[temperature, very\_high] \wedge [headache, yes]$, there are two decision attribute values, i.e., $[flu, yes]$ and $[flu, no]$. We know there are no objects which support an implication whose decision is $[flu, no]$. Probably, we will have a decision $[flu, yes]$ for this condition.

6. Concluding remarks

This paper surveyed the foundations of *Rough Non-deterministic Information Analysis* (*RNIA*) including *DISs* and *NISs*. As far as authors know, we have not seen any system with specified functionalities in this paper.

RNIA will take the complementary role in statistical data analysis, and *RNIA* is a new attempt to analyze data sets in addition to statistical data analysis. We are currently coping with *RNIA* web version [44].

Acknowledgement. The authors would be grateful to President of International Rough Set Society, Professor Dominik Ślęzak at Warsaw University and Professor Michinori Nakata at Josai International University for their discussion about *RNIA*.

References

[1] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, J. B. Bocca, M. Jarke, C. Zaniolo (eds.) VLDB'94, Morgan Kaufmann (1994), 487–499.

[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo, Fast discovery of association rules, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press (1996), 307–328.

[3] J. Bazan, H. S. Nguyen, S. H. Nguyen, P. Synak, J. Wróblewski, Rough set algorithms in classification problem, Rough Set Methods and Applications, Studies in Fuzziness and Soft Computing, **56**, Springer (2000), 49–88.

[4] J. Blaszczynski, S. Greco, R. Słowiński, Multi-criteria classification—a new scheme for application of dominance-based decision rules, European Journal of Operational Research, **181**(3) (2007), 1030–1044.

[5] A. Ceglar, J. F. Roddick, Association mining, ACM Computing Survey, **38**(2) (2006).

[6] M. R. Chmielewski, J. W. Grzymała-Busse, Global discretization of continuous attributes as preprocessing for machine learning, International Journal of Approximate Reasoning, **15**(4) (1996), 319–331.

[7] E. F. Codd, A relational model of data for large shared data banks, Communication of the ACM, **13**(6) (1970), 377–387.

[8] C. Cornelis, R. Jensen, G. H. Martín, D. Ślęzak, Attribute selection with fuzzy decision reducts, Information Sciences, **180**(2) (2010), 209–224.

[9] S. Demri, E. Orłowska, Incomplete Information: Structure, Inference, Complexity, Monographs in Theoretical Computer Science, An EATCS Series, Springer, 2002.

[10] A. Frank, A. Asuncion, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science (2010) http://mlearn.ics.uci.edu/MLRepository.html.

[11] S. Greco, B. Matarazzo, R. Słowiński, Granular computing and data mining for ordered data: The dominance-based rough set approach, R. A. Meyers (ed.) Encyclopedia of Complexity and Systems Science, Springer (2009), 4283–4305.

[12] J. W. Grzymała-Busse, A new version of the rule induction system LERS, Fundamenta Informaticae, **31**(1) (1997), 27–39.

[13] J. W. Grzymała-Busse, P. Werbrouck, On the best search method in the LEM1 and LEM2 algorithms, E. Orłowska (ed.) Incomplete Information: Rough Set Analysis, Studies in Fuzziness and Soft Computing, **13**, Springer (1998), 75–91.

[14] J. W. Grzymała-Busse, J. Stefanowski, Three discretization methods for rule induction, International Journal of Intelligent Systems, **16**(1) (2001), 29–38.

[15] J. W. Grzymała-Busse, Data with missing attribute values: Generalization of indiscernibility relation and rule induction, Transactions on Rough Sets, **1** (2004), 78–95.

[16] V. N. Huynh, Y. Nakamori, H. Ono, J. Lawry, V. Kreinovich, H. T. Nguyen (eds.), Interval/Probabilistic Uncertainty and Non-Classical Logics, Advances in Soft Computing, **46**, Springer (2008).

[17] V. N. Huynh, Y. Nakamori, C. Hu, V. Kreinovich, On decision making under interval uncertainty: A new justification of hurwicz optimism-pessimism approach and its use in group decision making, ISMVL 2009, IEEE Computer Society (2009), 214–220.

[18] M. Inuiguchi, Y. Yoshioka, Y. Kusunoki, Variable-precision dominance-based rough set approach and attribute reduction, International Journal of Approximate Reasoning, **50**(8) (2009), 1199–1214.

[19] J. Komorowski, Z. Pawlak, L. Polkowski, A. Skowron, Rough sets: a tutorial, S. K. Pal, A. Skowron (eds.) Rough Fuzzy Hybridization: A New Method for Decision Making, Springer (1999), 3–98.

[20] M. Kryszkiewicz, H. Rybinski, Computation of reducts of composed information systems, Fundamenta Informaticae, **27**(2–3) (1996), 183–195.

[21] M. Kryszkiewicz, Rough set approach to incomplete information systems, Information Sciences, **112**(1/4) (1998), 39–49.

[22] M. Kryszkiewicz, Rules in incomplete information systems, Information Sciences, **113**(3/4) (1999), 271–292.

[23] Y. Leung, M. M. Fischer, W. Z. Wu, J. S. Mi, A rough set approach for the discovery of classification rules in interval-valued information systems, International Journal of Approximate Reasoning, **47**(2) (2008), 233–246.

[24] W. Lipski, On semantic issues connected with incomplete information databases, ACM Transactions on Database Systems, **4**(3) (1979), 262–296.

[25] W. Lipski, On databases with incomplete information, Journal of the ACM, **28**(1) (1981), 41–70.

[26] T. Murai, G. Resconi, M. Nakata, Y. Sato, Operations of zooming in and out on possible worlds for semantic fields, L. J. E. Damiani, R. J. Howlett, N. Ichalkaranje (eds.) KES 2002, Frontiers in Artificial Intelligence and Applications, **82**, IOS Press (2002), 1083–1087.

[27] A. Nakamura, A rough logic based on incomplete information and its application, International Journal of Approximate Reasoning, **15**(4) (1996), 367–378.

[28] A. Nakamura, S. Tsumoto, H. Tanaka, S. Kobayashi, Rough set theory and its applications, Journal of Japanese Society for Artificial Intelligence, **11**(2) (1996), 209–215.

[29] M. Nakata, H. Sakai, Rough-set-based approaches to data containing incomplete information: possibility-based cases, K. Nakamatsu, J. M. Abe (eds.) Advances in Logic Based Intelligent Systems, Frontiers in Artificial Intelligence and Applications, **132**, IOS Press (2005), 234–241.

[30] M. Nakata, H. Sakai, Lower and upper approximations in data tables containing possibilistic information, Transactions on Rough Sets, **7**, Springer (2007), 170–189.

[31] M. Nakata, H. Sakai, Applying rough sets to information tables containing possibilistic values, Transactions on Computational Science, **2**, Springer (2008), 180–204.

[32] E. Orłowska, Z. Pawlak, Representation of nondeterministic information, Theoretical Computer Science, **29**(1/2) (1984), 27–39.

[33] E. Orłowska (ed.), Incomplete Information: Rough Set Analysis, Studies in Fuzziness and Soft Computing, **13**, Springer, 1998.
[34] E. Orłowska, Introduction: What you always wanted to know about rough sets, E. Orłowska (ed.) Incomplete Information: Rough Set Analysis, Studies in Fuzziness and Soft Computing, **13**, Springer (1998), 1–20.
[35] Z. Pawlak, Information systems theoretical foundations, Information Systems, **6**(3) (1981), 205–218.
[36] Z. Pawlak, Systemy Informacyjne: Podstawy teoretyczne, Wydawnictwa Naukowo-Techniczne Publishers, 1983.
[37] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, 1991.
[38] Z. Pawlak, Some issues on rough sets, Transactions on Rough Sets, **1** (2004), 1–58.
[39] W. Pedrycz, A. Skowron, V. Kreinovich (eds.), Handbook of Granular Computing, Wiley, 2008.
[40] L. Polkowski, A. Skowron (eds.), Rough Sets in Knowledge Discovery 1: Methodology and Applications, Studies in Fuzziness and Soft Computing, **18**, Springer, 1998.
[41] L. Polkowski, A. Skowron (eds.), Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Studies in Fuzziness and Soft Computing, **19**, Physica-Verlag, 1998.
[42] J. R. Quinlan, Improved use of continuous attributes in C4.5, The Journal of Artificial Intelligence Research, **4** (1996), 77–90.
[43] RNIA software tool: http://www.mns.kyutech.ac.jp/~sakai/RNIA.
[44] RNIA WEB software tool: http://getrnia.appspot.com/.
[45] H. Sakai, On a framework for logic programming with incomplete information, Fundamenta Informaticae, **19**(3/4) (1993), 223–234.
[46] H. Sakai, A. Okuma, An algorithm for finding equivalence relations from tables with non-deterministic information, N. Zhong, A. Skowron, S. Ohsuga (eds.) New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, LNAI, **1711**, Springer (1999), 64–72.
[47] H. Sakai, A. Okuma, An algorithm for checking dependencies of attributes in a table with non-deterministic information: A rough sets based approach, R. Mizoguchi, J. Slaney (eds.) PRICAI 2000 Topics in Artificial Intelligence, LNAI, **1886**, Springer (2000), 219–229.
[48] H. Sakai, A. Okuma, On a theorem prover for variational logic programs with functors setu and sets, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, **8**(1), World scientific (2000), 73–92.
[49] H. Sakai, Effective procedures for handling possible equivalence relations in non-deterministic information systems, Fundamenta Informaticae, **48**(4) (2001), 343–362.
[50] H. Sakai, Effective procedures for data dependencies in information systems, M. Inuiguchi, S. Tsumoto, S. Hirano (eds.) Rough Set Theory and Granular Computing, Studies in Fuzziness and Soft Computing, **125**, Springer (2003), 167–176.
[51] H. Sakai, A. Okuma, Basic algorithms and tools for rough non-deterministic information analysis, Transactions on Rough Sets, **1**, Springer (2004), 209–231.
[52] H. Sakai, Possible equivalence relations and their application to hypothesis generation in non-deterministic information systems, Transactions on Rough Sets, **2**, Springer (2004), 82–106.
[53] H. Sakai, M. Nakata, An application of discernibility functions to generating minimal rules in non-deterministic information systems, Journal of Advanced Computational Intelligence and Intelligent Informatics, **10**(5) (2006), 695–702.
[54] H. Sakai, R. Ishibashi, K. Koba, M. Nakata, Rules and apriori algorithm in non-deterministic information systems, Transactions on Rough Sets, **9**, Springer (2008), 328–350.
[55] H. Sakai, K. Koba, M. Nakata, Rough sets based rule generation from data with categorical and numerical values, Journal of Advanced Computational Intelligence and Intelligent Informatics, **12**(5) (2008), 426–434.

[56] H. Sakai, K. Hayashi, M. Nakata, D. Ślęzak, The lower system, the upper system and rules with stability factor in non-deterministic information systems, H. Sakai, M. K. Chakraborty, A. E. Hassanien, D. Ślęzak, W. Zhu (eds.) Rough Sets, Fuzzy Sets, Data Mining and Granular Computing, LNAI, **5908**, Springer (2009), 313–320.

[57] H. Sakai, M. Nakata, D. Ślęzak, Rule generation in Lipski's incomplete information databases, M. S. Szczuka, M. Kryszkiewicz, S. Ramanna, R. Jensen, Q. Hu (eds.) Rough Sets and Current Trends in Computing, LNAI, **6086**, Springer (2010), 376–385.

[58] H. Sakai, H. Okuma, M. Nakata, D. Ślęzak, Stable rule extraction and decision making in rough non-deterministic information analysis, International Journal of Hybrid Intelligent Systems, **8**(1) (2011), 41–57.

[59] H. Sakai, K. Hayashi, M. Nakata, D. Ślęzak, A mathematical extension of rough set based issues toward uncertain information analysis, New Mathematics and Natural Computation, **7**(3), World Scientific (2011), 543–570.

[60] H. Sakai, H. Okuma, M. Nakata, Rough non-deterministic information analysis: Foundations and its perspective in machine learning, Emerging Paradigms in Machine Learning, Smart Innovation, Systems and Technologies, **13**, Springer (2012), 215–247.

[61] H. Sakai, H. Okuma, M. Wu, M. Nakata, Rough non-deterministic information analysis for uncertain information, The Handbook on Reasoning-based Intelligent Systems, World scientific, Chapter 4 (2013) in press.

[62] H. Sakai, M. Wu, M. Nakata, Association rule-based decision making in table data, International Journal of Reasoning-based Intelligent Systems, **4**(3), Inderscience (2012), 162–170.

[63] A. Skowron, C. Rauszer, The discernibility matrices and functions in information systems, R. Słowiński (ed.) Intelligent Decision Support—Handbook of Advances and Applications of the Rough Set Theory, Kluwer Academic Publishers (1992), pp. 331–362.

[64] D. Ślęzak, J. Wróblewski, Roughfication of numeric decision tables: The case study of gene expression data, J. Yao, P. Lingras, W. Z. Wu, M. Szczuka, N. Cercone, D. Ślęzak (eds.) Rough Sets and Knowledge Technology, LNAI **4481**, Springer (2007), 316–323.

[65] D. Ślęzak, Rough sets and functional dependencies in data: Foundations of association reducts, Transactions on Computational Science, **5** (2009), 182–205.

[66] D. Ślęzak, V. Eastwood, Data warehouse technology by infobright, U. Çetintemel, S. B. Zdonik, D. Kossmann, N. Tatbul (eds.) Proceedings of the ACM SIGMOD International Conference on Management of Data SIGMOD2009, ACM (2009), 841–846.

[67] D. Ślęzak, H. Sakai, Automatic extraction of decision rules from non-deterministic data systems: Theoretical foundations and SQL-based implementation, D. Ślęzak, T. h. Kim, Y. Zhang, J. Ma, K. i. Chung (eds.) Database Theory and Application, Communications in Computer and Information Science, **64**, Springer (2009), 151–162.

[68] J. Stefanowski, A. Tsoukiàs, On the extension of rough sets under incomplete information, N. Zhong, A. Skowron, S. Ohsuga (eds.) New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, LNAI, **1711**, Springer (1999), 73–81.

[69] J. Stefanowski, A. Tsoukiàs, Incomplete information tables and rough classification, Computational Intelligence, **17**(3) (2001), 545–566.

[70] S. Tsumoto, Knowledge discovery in clinical databases and evaluation of discovered knowledge in outpatient clinic, Information Sciences, **124**(1–4) (2000), 125–137.

[71] X. Yang, D. Yu, J. Yang, L. Wei, Dominance-based rough set approach to incomplete interval-valued information system, Data & Knowledge Engineering, **68**(11) (2009), 1331–1347.

[72] Y. Yao, C. J. Liau, N. Zhong, Granular computing based on rough sets, quotient space theory, and belief functions, N. Zhong, Z. W. Ras, S. Tsumoto, E. Suzuki (eds.) ISMIS 2003, LNAI, **2871**, Springer (2003), 152–159.

[73] L. A. Zadeh, Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems, **90**(2) (1997), 111–127.
[74] W. Ziarko, Variable precision rough set model, Journal of Computer and System Sciences, **46**(1) (1993), 39–59.

(M. Wu)
Department of Applied Science for Integrated System Engineering
Graduate School of Engineering
Kyushu Institute of Technology
Tobata, Kitakyushu 804-8550, Japan
E-mail: wumogaku@yahoo.co.jp

(N. Yamaguchi)
Department of Integrated System Engineering
School of Engineering
Kyushu Institute of Technology
Tobata, Kitakyushu 804-8550, Japan
E-mail: KITYN1124@gmail.com

(H. Sakai)
Department of Mathematics
Kyushu Institute of Technology
Tobata, Kitakyushu 804-8550, Japan
E-mail: sakai@mns.kyutech.ac.jp