Rough Mereological Calculi of Granules: A Rough Set Approach to Computation

Lech Polkowski[1], *Andrzej Skowron*[2]

[1] Polish–Japanese Institute of Information Technology
Koszykowa 86, 02-008 Warsaw, Poland
and
Department of Mathematics and Information Sciences
Warsaw University of Technology
Pl. Politechniki 1, 00-665 Warsaw, Poland
email: polkow@pjwstk.waw.pl

[2] Institute of Mathematics, Warsaw University
Banacha 2, 02-097 Warsaw, Poland
email: skowron@mimuw.edu.pl

Abstract. Rough Mereology is a paradigm allowing for a synthesis of main ideas of two potent paradigms for reasoning under uncertainty: Fuzzy Set Theory and Rough Set Theory. Approximate reasoning is based in this paradigm on the predicate of being a *part to a degree*. We present applications of Rough Mereology to the important theoretical idea put forth by Lotfi Zadeh [28], [29], i.e., Granularity of Knowledge: we define granules of knowledge by means of the operator of mereological class and we extend the idea of a granule over complex objects like decision rules as well as decision algorithms. We apply these notions and methods in the distributed environment discussing complex problems of knowledge and granule fusion. We express the mechanism of complex granule formation by means of a formal grammar called Synthesis Grammar defined over granules of knowledge, granules of classifying rules, or over granules of classifying algorithms. We finally propose hybrid rough–neural schemes bridging rough and neural computations.[3]

Keywords: *rough mereology, rough inclusion, granularity of knowledge, distributed systems, rough–neural computation*

1 Introduction

1.1 Motivation

The concept of a granule of knowledge goes back to Zadeh [27] where concepts of a linguistic variable and a granule were first discussed. To recall the Zadeh program in a nutshell, a *granule* is a collection of objects drawn together by similarity or functionality and considered therefore as a whole. In Zadeh's program, granules served as denotations for phrases of a natural language, allowing

[3] This article is an extension of the communication [17].

thus for "computing with words". As pointed by Zadeh elsewhere and recently cf.[30], [31], in granule calculus, an important role is played by propagation of constraints which define granules from premises to conclusions (i.e. from problem specifications to the problem solution).

It follows that any calculus of granules should provide the following ingredients:

1. definitions of granules via certain constraints
2. mechanisms for propagating constraints among distinct stages (agents) in the solution finding process
3. mechanisms providing for correctness of solutions

In this work, we propose to base granulation mechanisms and granule calculus on rough mereology, a methodology extending both rough set theory and fuzzy set theory and rooted in classical theories of ontology and mereology proposed by Stanislaw Leśniewski [5], [6]. We give an outline of these theories in respective chapters below. Our approach allows us to define granules as classes of objects close in a given degree one to another (closeness is measured by values of rough inclusion, the basic functor of rough mereology), to introduce mechanisms of propagation of closeness by means of rough mereological connectives, and to study stability of resulting reasoning schemes. This fulfills the Zadeh program. As all ingredients of our formalization may be extracted from data, this approach is of special interest to applications and thus to researchers occupying themselves with problems related to data analysis, feature extraction, data mining, knowledge discovery etc. etc..

We also show that in rough mereological setting, one may address the Computing with Words paradigm, by defining Synthesis Grammars and resulting formal Synthesis Languages, which provide a linguistic rendering of rough mereological schemes for reasoning.

Finally, motivated by an analogy between rough mereological schemes and neuro - computing schemes, we state a program of research towards creating a high–level neuro–computing with granules of knowledge.

Now, we begin with basic notions of Rough Set Theory [9], [13], [14], [15], [16] whose ideas have been a motivation for Rough Mereology.

1.2 Rough Set Theory: An Introduction

Knowledge is represented in rough set approach by means of an information system $\mathsf{A} = (U, A)$ where U is a (current) set of objects, A is a (current) set of (conditional) attributes and each attribute $a \in A$ is a mapping on the set U, i.e., $a : U \to V_a$ where V_a is the set of values of a. The information system A may be regarded as an available to us subsystem of an information system $\mathsf{A}^{\infty} = (U^{\infty}, A^{\infty})$ providing the complete knowledge about the world in question at the cost of U^{∞}, A^{∞} being possibly (countably) infinite. As we know, like in machine learning, A only, our knowledge is incomplete and inferences from it are uncertain. Rough set theory proposes a formal mechanism by means of which one may exhibit and discuss analytically these features of knowledge.

Indiscernibility. We assume thus that only A provides us with knowledge about the world. In consequence, objects with identical descriptions are not discernible: for each $u \in U$ and a set $B \subseteq A$ of attributes, we define the *information set* $Inf_B(u)$ of u over the set$B$ by

$$Inf_B(u) = \{(a, a(u)) : a \in B\}$$

and we express indiscernibility of objects with respect to B by the relation IND_B of *B–indiscernibility*:

$$(u, w) \in IND_B \Longleftrightarrow Inf_B(u) = Inf_B(w).$$

The indiscernibility relation IND_B partitions the set U into indiscernibility classes $[u]_B$. They may be represented in propositional calculus of *descriptors*, i.e., pairs (a, v) where $a \in A$ and $v \in V_a$. The meaning $[\alpha]$ of any formula α from this language (in A) is defined inductively by $[(a, v)] = \{u \in U : a(u) = v\}$, $[\alpha \wedge \beta] = [\alpha] \cap [\beta]$ and $[\neg\alpha] = U - [\alpha]$. Given $[u]_B$ with $B = \{a_{i_1}, a_{i_2}, .., a_{i_k}\}$ and $a_{i_j}(u) = v_{i_j}$, we find that the formula $\wedge_j(a_{i_j} = v_{i_j})$, called a *B-template* and denoted by (B, v) is satisfied in A by and only by elements of $[u]_B$. A *B-elementary atomic granule* in A is any pair $((B, v), [(B, v)])$ where $B \subseteq A$ and an *elementary atomic granule* (in A) is any B-elementary atomic granule (in A).

Approximations. Some *concepts* X (i.e., subsets X of the universe U) may be described *exactly* in terms of *attribute–value descriptors* of the form (a, v) where $a \in A$ and $v \in V_a$. We will say that, in general, a concept X is *exact* (in A) if it is the meaning of a formula in the descriptor language. When it is not the case, the concept X is said to be *rough*. Hence it follows that any exact set can be described as a union of some indiscernibility classes.

A rough concept X may be described approximately only by means of its *approximations*; given a set $B \subseteq A$ of attributes, we define two approximations to X viz.

1. the *B–lower approximation*: $\underline{B}X = \{u \in U : [u]_B \subseteq X\}$;
2. the *B–upper approximation*: $\overline{B}X = \{u \in U : [u]_B \cap X \neq \emptyset\}$.

It follows that

$$\underline{B}X \subseteq X \subseteq \overline{B}X$$

is the description of X over B, classifying $u \in \underline{B}X$ as belonging necessarily in X and $u \in \overline{B}X$ as belonging possibly in X.

Dependencies. Dependencies between templates in A, say (B, v) and (C, w), are expressed by formulae of the form

$$\wedge_j(a_{i_j} = v_{i_j}) \Longrightarrow \wedge_k(b_{i_k} = w_{i_k})$$

where $B = \{a_{i_j}\}$, $C = \{b_{i_k}\}$. These dependencies in turn may be regarded as *classifying rules* allowing to define C–values of an object in terms of its B–values.[4]

The power (quality) of such classifying rule R in A is characterized by two parameters viz. (see e.g., [3], [11])

$$c(R) = \frac{|[\alpha] \cap [\beta]|}{|[\alpha]|} \ (\textit{classification accuracy or confidence});$$

$$\kappa(R) = \frac{|[\alpha] \cap [\beta]|}{|[\beta]|} \ (\textit{coverage});$$

where $[\alpha], [\beta]$ denote the meaning of the templates on the left and on the right hand side of the rule R, respectively and $|X|$ is the cardinality of the set X.

These parameters express what part of $[\alpha]$ is contained in $[\beta]$ and vice versa (one may give these parameters also a probabilistic interpretation as unbiased estimators of the corresponding conditional probabilities cf. [11]). [5]

1.3 Granulation: an orthodox way

Any non-empty set $\mathcal{F}$ of atomic elementary granules in A of the form

$$((B, v), [(B, v)])$$

may be treated as the set of atomic granules from which an algebra of elementary granules is formed by means of operations $\oplus, \odot, \ominus$ defined by

1. $(\alpha, [\alpha]) \oplus (\beta, [\beta]) = (\alpha \vee \beta, [\alpha] \cup [\beta])$;
2. $(\alpha, [\alpha]) \odot (\beta, [\beta]) = (\alpha \wedge \beta, [\alpha] \cap [\beta])$;
3. $\ominus(\alpha, [\alpha]) = (\neg\alpha, U - [\alpha])$.

These elementary granules are defined in terms of indiscernibility relations and may be called *indiscernibility granules.*

Let us mention that we have chosen elementary atomic granules as atomic ones only to simplify our presentation. One can choose more complex granules as atomic granules, e.g., a predefined set $\mathcal{F}$ of indiscernibility granules.

Together with operations on granules one should define a relation of granule inclusion. We will see later that this is a very basic notion in our approach. Now we would like to give an example of such relation.

[4] Let us observe that dependencies above have a general format of *association rules* cf. [1] i.e. the format of an implication $P \longrightarrow Q$ where P, Q are sets of symbolic conditions interpreted in a universe of objects. In our case, these conditions are descriptors related to templates.

[5] Let us observe that in case of association rules of the form $P \longrightarrow Q$, a parameter called *support* is used, along confidence, defined as the number of objects in the universe which satisfy the condition $P \wedge Q$ (cf. [1]); our terminology follows that of Tsumoto [26]. The formula defining confidence of a rule goes back to the Łukasiewicz idea of associating a probability with an implication [8].

One can choose a syntactical definition of the inclusion relation μ assuming that a granule $g = ((C, w), [(C, w)])$ is included in a granule $g' = ((B, v), [(B, v)])$, in symbols $\mu(g, g')$, if and only if (B, v) is a subformula of (C, w). From the definition we have that if $\mu(g, g')$ holds then $[g] \subseteq [g']$. This is a crisp inclusion relation. In the sequel we will discuss also soft inclusion relations, called *rough inclusions* specifying that granules are included one in another in a degree. In some cases degrees can be interpreted as numbers in the interval [0,1] of real numbers; in general, the degrees are also information granules.

Let us consider one more example of granule construction. We assume the following rule, called *the granule collection rule* of new granule formation:

if $g_1, ..., g_n$ are granules **then** $(\{g_1, ..., g_n\}, [\{g_1, ..., g_n\}])$ is a granule

where $[\{g_1, ..., g_n\}] = \{[g_1], ..., [g_n]\}$.

In particular, let us consider the following granule:

$$g = \{\{g_1\}, ..., \{g_r\}\}$$

where $g_i = \{g_1^{(i)}, ..., g_{k_i}^{(i)}\}$ for $i = 1, ..., r$ and all granules $g_j^{(i)}$ are elementary ones.

Sets of decision rules generated from a given decision table can be interpreted as granules of the above form [23]. First, one can construct granules corresponding to each particular decision by taking a collection of left hand sides of decision rules for a given decision. Next, one can construct from them a collection represented by one granule.

Let us consider now an extension of the inclusion relation on elementary granules to the new granules assuming

$\mu_p(((B, v), [(B, v)]), g)$ holds iff

$$p = ((\lambda_1^{(1)}, ..., \lambda_{k_1}^{(1)}), ..., (\lambda_1^{(r)}, ..., \lambda_{k_r}^{(r)}))$$

where $\lambda_j^{(i)} = 1$ iff $\mu(((B, v), [(B, v)]), g_j^{(i)})$ holds, and 0 otherwise.

The above definition has a natural interpretation. An elementary granule $((B, v), [(B, v)])$ corresponding to, e.g., a new tested object, is included in the set of decision rules represented by the granule g in a degree p (which as above is a binary sequence) depending on whether the elementary granule is included in the left hand sides of decision rules or not.

Assuming a function $Conflict\_Resolving$ transforming such degrees into values in $\{1, ..., r\}$ (i.e., the set of possible decisions) is given, one can construct a new granule $Classifier$ by

$Classifier(((B, v), [(B, v)]), g) = i$ iff

$Conflict\_Resolving(p) = i$ and $\mu_p(((B, v), [(B, v)]), g)$ holds.

Hence one can see that classifiers can be treated as special cases of granules. In the next section we will consider another example.

1.4 Similarity approach

Both accuracy and coverage are based on the rough membership function [10] and the representation of concepts in terms of indiscernibility classes. In recent applications, the need has been stressed for more relaxed approach based not only on indiscernibility but also on a variety of tolerance (similarity) relations [16], [24], [4]. We recall that a *tolerance relation* τ on U is a binary relation satisfying the conditions

1. $u\tau u$ (reflexivity);
2. $u\tau u' \longrightarrow u'\tau u$ (symmetry).

   We may be interested in tolerance relations extending indiscernibility, i.e., we may add the condition
3. $uIND_A u' \longrightarrow u\tau u'$.

In some applications, it may be convenient to relax conditions on τ and give up the symmetry property; in that case we will call τ a *similarity relation*(cf. [24]).

Once a relation τ is established, concepts may be approximated via τ; to this end, respective definitions need to be modified slightly only: each indiscernibility class $[u]_A$ is replaced with the tolerance (or, similarity) class $\tau(u) = \{u' : u\tau u'\}$ (notice that in case of a similarity, classes $\tau(u)$ split into two classes, respectively, left and right, so we have a richer approximation structure). Several authors [3], [4] reported results of experiments indicating that these approximations lead to classification rules better suited to the task of new object classification.

Similarity relations in U are often induced by similarity relations defined on the attribute value vectors. More precisely, let $\tau_B \subseteq INF(B) \times INF(B)$ be a tolerance relation where $INF(B)$ is the Cartesian product $\times_{a\in B} V_a$. The relation τ_B induces a tolerance relation $\tau \subseteq U \times U$ defined by $x\tau y$ if and only if $Inf_B(x)\tau_B Inf_B(y)$.

1.5 Similarity granules

The notion of a granule does undergo a revision when indiscernibility is replaced with a similarity: *elementary similarity granules* become now similarity classes of the form $\tau(u)$.

We assume that a fixed family T of tolerance relations τ_B for $B \subseteq A$ is given where A is a given set of attributes of the system A.

An expression $(\tau_B : (B, v))$ is called an *atomic tolerance formula* if (B, v) is a template and $\tau_B \subseteq INF(B) \times INF(B)$ is a tolerance relation from a given family T of tolerance relations. Its semantics $[\tau_B : (B, v)]$ in A is defined by $\{[(B, w)] : w\tau_B v\}$. Any pair of the form

$$((\tau_B : (B, v)), [\tau_B : (B, v)])$$

is called a *B-tolerance atomic granule* in A. B-tolerance granules represent clusters of granules defined by the tolerance relation τ_B.

Let $\mathcal{F}$ be a set of tolerance atomic formulas. $\mathcal{F}$-tolerance propositional formulas and their semantics $[\cdot]$ are defined inductively by

1. any atomic tolerance formula $(\tau_B : (B, v))$ from $\mathcal{F}$ where $B \subseteq A$ and $v \in INF(B)$ is an $\mathcal{F}$-tolerance formula; its semantics is defined by $\{[(B, w)] : w \tau_B v\}$;
2. if α, β are $\mathcal{F}$-tolerance formulas then $(\alpha \vee \beta)$ and $(\alpha \wedge \beta)$ are $\mathcal{F}$-tolerance formulas; their semantics is defined by $[\alpha] \cup [\beta]$ and $[\alpha] \cap [\beta]$, respectively;
3. if α is an $\mathcal{F}$-tolerance formula then $\neg\alpha$ is an $\mathcal{F}$-tolerance formula; its semantics is defined by $\{[(B, v)] : [(B, v)] \notin [\alpha]\}$.

Now we can introduce an example of the algebra of $\mathcal{F}$-tolerance (similarity) granules.

Any pair

$$((\tau_B : (B, v)), [\tau_B : (B, v)])$$

where $(\tau_B : (B, v)) \in \mathcal{F}$ is called an *$\mathcal{F}$-tolerance atomic granule* in A. From these granules an algebra of granules is formed by means of operations $\oplus, \odot, \ominus$ defined by

1. $(\alpha, [\alpha]) \oplus (\beta, [\beta]) = (\alpha \vee \beta, [\alpha] \cup [\beta])$;
2. $(\alpha, [\alpha]) \odot (\beta, [\beta]) = (\alpha \wedge \beta, [\alpha] \cap [\beta])$;
3. $\ominus(\alpha, [\alpha]) = (\neg\alpha, [\neg\alpha])$.

Analogously as before, we define a granule inclusion relation between elementary granules and tolerance granules. One can choose a syntactical definition of granule inclusion μ_p assuming that an elementary granule $g = ((C, w), [(C, w)])$ is included in a granule $g' = ((\tau_B : (B, v)), [\tau_B : (B, v)])$ in degree p, in symbols $\mu_p(g, g')$, if and only if $(B, w \mid B)\tau_B^l(B, v)$ for some $l \leq p$ where $w \mid B$ denotes the restriction of w to B. Hence, $g = ((C, w), [(C, w)])$ is included in degree p in g' if and only if g belongs to a *neighborhood of* (B, v) *of dimension at most* p *defined by the tolerance relation* τ_B, i.e., it can be reached from (B, v) by an iteration at most p–fold of the tolerance relation τ_B.

Assuming the rule for granule collection formation one can consider again granules corresponding to decision rule sets. However, now we have an additional parameter for tuning, namely the degree p of inclusion between elementary granules and tolerance granules.

An extension of the inclusion relation to granules representing sets of decision rules can be now defined as follows:

$\mu_p((B, v), [(B, v)]), g)$ holds if and only if

$$p = ((\lambda_1^{(1)}, ..., \lambda_{k_1}^{(1)}), ..., (\lambda_1^{(r)}, ..., \lambda_{k_r}^{(r)}))$$

where $\lambda_j^{(i)} = l$ iff l is the minimal integer such that $\mu_l((B, v), [(B, v)]), g_j^{(i)})$ holds.

The above definition has a natural interpretation. An elementary granule $(B, v), [(B, v)])$ corresponding to, e.g., a new tested object is included in the set of decision rules represented by the granule g in a degree p depending on how close this elementary granule is to the left hand sides of decision rules.

Assuming – as in case of nearest neighbor classifiers [7] – that a function

$$Conflict\_Resolving$$

transforming any such degree into a value from $\{1, ..., r\}$ (i.e., the set of possible decisions) is given, one can construct a new granule $Classifier$ by

$Classifier((B,v),[(B,v)]),g) = i$ if and only if

$Conflict\_Resolving(p) = i$ and $\mu_p((B,v),[(B,v)]),g)$ holds.

The classifier is *close in degree* q to a given partition $X_1, ..., X_r$ of the universe of objects U into decision classes if and only if for any $x \in U$ the set

$$\{x \in U : \text{Classifier}(((B,v),[(B,v)]),g) = i\}$$

is *ν-close* (under a chosen measure ν) in degree at least q to the decision class X_i for any $i = 1, ..., r$.

The examples given above demonstrate that information granules can model quite complex objects. However, one could ask if we really need more. The existing approaches in soft computing help to model soft concepts and also attempt to model the human thought process in a decision–making system. However, there is no relation of this process to the architecture of the human information processing system [12]. What is missed in existing approaches? Let us consider two aspects. Existing tools for reasoning offered by logic can not mimic the strategy of reasoning which seems to be so basic for human beings. It is based on a possibility to extract very efficiently from complex objects their soft parts and soft relationships between them which are sufficient to carry out reasoning with acceptable precision (possibly not exactly). Thanks to this ability of perception, schemes of reasoning based on soft patterns are representing huge clumps of crisp reasoning schemes. Moreover, these schemes, being of small size, are describing soft concepts with a sufficient quality. This allows for carrying out the reasoning efficiently.

In the next section we discuss shortly rough mereological approach [13], [14], [15], [18], [17] introduced to deal with such problems. In particular, we will discuss how complex granules and reasoning based on information granulation can be modeled using this approach.

1.6 Rough Mereology: graded families of granules

We propose in this paper the *rough mereological approach* to the granulation problem in which similarity classes are induced by means of the mereological predicate of *being a part in a degree* (called also a *rough inclusion*)as *mereological collective classes* indexed by degrees of being a part (i.e. real numbers from the interval $[0, 1]$). Rough Mereology is a simultaneous extension of the mereological theory of approximate reasoning, fuzzy set theory, and rough set theory.

In order to discuss Rough Mereology, we need to introduce basic notions of Ontology and Mereology.

Ontology (i.e. Theory of Names (Concepts)) [20] sets a mechanism for creating names and introducing relationships of inclusion among them; the primitive predicate of Ontology is the copula ***is*** occurring in relations of the form X ***is*** Y meaning that any object called X is also called Y. We denote the copula ***is*** with the symbol ε.

Mereology is a theory of objects based on the opposition *part–whole* and employing a primitive notion of *being a part*. Set in the framework of Ontology, Mereology provides us with a mechanism of forming collective classes for names of Ontology. The precise meaning of this assertion will become clear in the sequel, in respective sections devoted to mereology.

2 Ontology in Information Systems

Our ontology is adopted from Ontology of S. Leśniewski [5]. Ontology is a theory of the copula ***is*** and we render it here as the predicate ε. We first introduce formally this predicate. To this end, we adopt the Ontology Axiom of Leśniewski.
The Ontology Axiom

$$X\varepsilon Y \iff \exists Z.Z\varepsilon X \wedge \forall U,W.(U\varepsilon X \wedge W\varepsilon X$$

$$\implies U = W) \wedge \forall T.(T\varepsilon X \implies T\varepsilon Y).$$

These three conjuncts express respectively that

1. $\exists Z.Z\varepsilon X$ (X is a non–empty name);
2. $\forall U,W.(U\varepsilon X \wedge W\varepsilon X \implies U = W)$ (X is a singleton (i.e. an *individual*));
3. $\forall T.(T\varepsilon X \implies T\varepsilon Y)$ (any entity called X is also called Y).

This defines the meaning of the copula ε. In particular, it follows from the Ontology Axiom that $X\varepsilon X$ states that X is an *individual* object. The meaning of the copula ε may be now stated concisely as follows: $X\varepsilon Y$ means that (i) X is an individual object (e.g. "Socrates") (ii) Y is a name (distributive class: a set, a list, a collection), e.g., "man". Thus, our setting may be also expressed in the language of naive set theory: X is a singleton, Y is a set and $X \in Y$.

2.1 Examples: Rough Set Ontology in Information Systems

As individuals, we consider elementary granules.

We have introduced operations on elementary granules denoted by $\oplus, \odot, \ominus$ corresponding to the boolean structure in Calculus of Names. For example

1. $(C,v) \vee (D,w)$ is the name of the granule $((C,v),[(C,v)]) \oplus ((D,w),[(D,w)])$ and $[(C,v) \vee (D,w)]$ consists of those objects which are from $[(C,v)]$ or from $[(D,w)]$;
2. $(C,v) \wedge (D,w)$ is the name of the granule $((C,v),[(C,v)]) \odot ((D,w),[(D,w)])$ and $[(C,v) \wedge (D,w)]$ consists of those objects which fall both in $[(C,v)]$ and $[(D,w)]$;

3. $\neg(D, w)$ is the name of the granule $\ominus((D, w), [(D, w)])$ where $[\neg(D, w)]$ consists of those objects which fall in U but not in $[(D, w)]$.

One may see that sets definable by introduced Boolean formulas may be described by formulas in DNF form $\vee_{i=1}^{k}(C_i, v_i)$ for an appropriate k and a choice of $C_i \subseteq A, v_i = Inf_{C_i}(u_i), u_i \in U$ where $i = 1, 2, .., k$. For any individual being an elementary granule $g = (\alpha, [\alpha])$ we introduce the name $Name(g)$ of the form $\vee_{i=1}^{k}(C_i, v_i)$ as the representation of α in DNF.

Having defined individual entities i.e. names of elementary granules we may proceed to more complex (compound) entities. Their names may be formed by means of set constructors applied to elementary granules. For instance, any decision algorithm regarded as a collection of granules is such a compound. We return to this example in Sections on Mereology and Rough Mereology.

3 Mereology in Information Systems

Our Mereology is an adaptation of Mereology proposed by Leśniewski [5], [6] which has offered a formal treatment of the predicate of being a *part* (cf. [2] for an alternative Mereology based on Connection predicate). We begin with the notion of a *part* predicate (*pt*, for short). The predicate *pt* is introduced into Ontology by means of additional axioms. We stress that $pt(Y)$ is defined only for an individual object Y.

3.1 Mereology Axioms

(ME1) $X\varepsilon pt(Y) \wedge Y\varepsilon pt(Z) \Longrightarrow Xpt(Z)$ (symmetry of *pt*);
(ME2) $non(Xpt(X))$ (non–reflexivity of *pt*).

It follows that the predicate *pt* defines the notion of a *proper part*.

The concept of an *improper part* is reflected in the notion of an *element* denoted *el* and defined as follows:

$$X\varepsilon el(Y) \Longleftrightarrow X\varepsilon pt(Y) \vee X = Y.$$

Basic feature of Mereology of Leśniewski is the presence of the *class predicate*, denoted Kl, making distributive classes (lists, collections, properties) into individual objects (*collective classes*) and defined as follows.

3.2 Collective Classes: The Class Predicate

The class predicate is defined by means of the following formula

$$X\varepsilon Kl(Y) \Longleftrightarrow$$
$$\exists Z.Z\varepsilon Y \wedge \forall Z.(Z\varepsilon Y \Longrightarrow Z\varepsilon el(X))\wedge$$
$$\forall Z.(Z\varepsilon el(X) \Longrightarrow \exists U, W.U\varepsilon Y \wedge W\varepsilon el(U) \wedge W\varepsilon el(Z)).$$

Let us look at the subsequent conjuncts in the defining formula above.

1. $\exists Z.Z\varepsilon Y$ (Y is a non–empty name);
2. $\forall Z.(Z\varepsilon Y \Longrightarrow Z\varepsilon el(X))$ (meaning that any individual listed in Y is an element of $Kl(Y)$);
3. $\forall Z.(Z\varepsilon el(X) \Longrightarrow \exists U,W.(U\varepsilon Y \wedge W\varepsilon el(U) \wedge W\varepsilon el(Z))$ (any element of $Kl(Y)$ has an element in common with an individual in Y).

Thus, the class functor does paste together individuals in Y by means of their common elements. The reader has undoubtedly noticed a formal analogy of the class operator to the union of sets operator in naive set theory: interpreting the element as a subset, we get from the class operator the union (of a family of sets) operator.

One also requires of the class operator:

(ME3) $X\varepsilon Kl(Y) \wedge Z\varepsilon Kl(Y) \Longrightarrow Z = X$ ($Kl(Y)$ is an individual);
(ME4) $\exists Z.Z\varepsilon Y \Longleftrightarrow \exists Z.Z\varepsilon Kl(Y)$ (the class exists for each non-empty Y).

The class operator will be used by us as a granule–forming tool in the sequel.

3.3 Mereology in Information Systems

Recall that individuals are elementary granules. Let $g = (\alpha, [\alpha])$ and $g' = (\beta, [\beta])$ be individuals.

We define the functor *pt* in this case: if the name of g' is $\vee_{i=1}^{k}[C_i, v_i]$, we let

$$Name(g)\varepsilon pt(Name(g'))$$

if and only if one of the following conditions is satisfied:

1. $k \geq 2$ and $\alpha = (D, w)$ (for some D, w) and $C_i \subseteq D$, $w|C_i = v_i$ (for some i);
2. $k = 1$ and $\alpha = (D, w)$ (for some D, w) and $C_1 \subset D$ and $w|C_1 = v_1$;

Let us observe that: given a name $Y = \vee_{i=1}^{k}(C_i, v_i)$, we have: $Kl(Y) = Name(g)$ where g is an elementary granule with the name Y, indeed, by the class definition, it suffices to check that any part $g' = ((D, w), [(D, w)])$ of g, has an element (e.g., g' itself) which is in turn an element of an individual $(((C_i, v_i), [(C_i, v_i)])$ in the definition) which is in g.

In particular, we have: $Kl((B, v)) = ((B, v), [(B, v)])$ for any (B, v), where we identify the atomic elementary granule $((B, v), [(B, v)])$ with its name (B, v).

4 Rough Mereology

Rough Mereology [13], [14], [21] has been proposed and studied as a tool for approximate reasoning. Its primitive notion is that of a *rough inclusion* i.e. a family of functors μ_r of being a part in degree at least r for $r \in [0, 1]$.

The following is a list of basic postulates about Rough Mereology. We introduce a graded family μ_r, where $r \in [0, 1]$ is a real number from the unit interval,

of predicates which would satisfy the following conditions ($\mu_r(Y)$ is a new property derived from Y via μ_r and we use the relational notation $X\varepsilon\mu_r(Y)$ for the statement: *X is a part of Y in degree at least r*).
(RM1) $X\varepsilon\mu_1(Y) \Longleftrightarrow X\varepsilon el(Y)$ (any part in degree 1 is an element);
(RM2) $X\varepsilon\mu_1(Y) \Longrightarrow \forall Z.(Z\varepsilon\mu_r(X) \Longrightarrow Z\varepsilon\mu_r(Y))$ (monotonicity);
(RM3) $X = Y \wedge X\varepsilon\mu_r(Z) \Longrightarrow Y\varepsilon\mu_r(Z)$ (identity is a μ–congruence);
(RM4) $X\varepsilon\mu_r(Y) \wedge s \leq r \Longrightarrow X\varepsilon\mu_s(Y)$ (degree at least r).

4.1 Rough Inclusions in Information Systems

We may define rough inclusions on individual objects, i.e. exact sets, directly or we may induce rough measures of closeness on rows of an information systems and then extend them to rough inclusions on exact sets. We include some examples on this topic.

Rough inclusions on rows. The following procedure may be used to define an exemplary rough inclusion on rows in an information system (U, A).
Procedure 1

1. Consider a partition $P = \{A_1, A_2, ..., A_k\}$ of A;
2. Select a family of coefficients: $W = \{w_1, w_2, ..., w_k\}$ where $w_i \geq 0$, any i, and $\sum_{i=1}^{k} w_i = 1$;
3. Define $IND(A_i)(u, v) = \{a \in A_i : u(a) = v(a)\}$ for $u, v \in INF(A)$;
4. Let $r = \sum_{i=1}^{k} w_i \frac{card(IND(A_i)(u,v))}{card(A_i)}$;
5. Assume $u\varepsilon\mu_r(v)$
6. Let $x\varepsilon\mu_r(y)$ if and only if $Inf_A(x)\varepsilon\mu_r(Inf_A(y))$.

We now propose a method for extending a measure defined for elements of two concepts to a measure on these two concepts.

Assume that we are given two individuals X, Y being classes of (finite) names: $X = Kl(X')$, $Y = Kl(Y')$ and that we have defined values of μ for pairs T, Z of individuals where $T\varepsilon X'$, $Z\varepsilon Y'$.

We extend μ to a measure μ^* on X, Y by letting:

$$Y\varepsilon\mu_r^*(X) \text{ for } r = min_{Z\varepsilon Y'}\{max_{T\varepsilon X'} max\{s : Z\varepsilon\mu_s(T)\}\}.$$

It may be proved straightforwardly that

Proposition 1. *The measure μ^* satisfies (RM1)–(RM4).*

Rough inclusions based on frequency count. In this case our strategy is based on counting frequencies by means of the rough membership function applied to specifically defined counted objects; particular strategies depend on the type of individual objects we consider. We point to few cases.

1. In case our individual objects g, g' are B–elementary granules, we may apply the strategy of counting the number of B–indiscernibility classes in respectively $[g] \cap [g']$ and $[g']$. Accordingly, for $[g] = \bigcup_{i=1}^{k}[(B, v_i)]$ and $[g'] = \bigcup_{j=1}^{m}[(B, w_j)]$, we let $Name(g')\varepsilon\mu_r(Name(g))$ where
$$r = \frac{\mid \{[(B, v_i)] : i \leq k\} \cap \{[(B, w_j)] : j \leq m\} \mid}{m};$$
2. In case our individual objects g, g' are elementary granules in $\mathsf{A} = (U, A)$, we may apply the strategy of counting rows: for two elementary granules g, g' with $[g] = \bigcup_{i=1}^{k}[(B_i, v_i)]$ and $[g'] = \bigcup_{j=1}^{m}[(C_j, w_j)]$, we let
$$Name(g)\varepsilon\mu_r(Name(g'))$$
where
$$r = \frac{\mid [g] \cap [g'] \mid}{\mid [g'] \mid}.$$
We also may apply the strategy of counting indiscernibility classes assuming
$$Name(g)\varepsilon\mu_r(Name(g'))$$
where
$$r = \frac{\mid \{[(B_i, v_i)] : i \leq k\} \cap \{[(C_j, w_j)] : j \leq m\} \mid}{m}.$$
3. We may apply a hybrid approach counting rows for indiscernibility classes and extending the received closeness measure to general individuals. First then, we define μ on atomic elementary granules $g = ((B, v), [(B, v)])$, $g' = ((C, w), [(C, w)])$: first, we define the set $IND(g, g') = \{a \in A : a \in B \cap C \wedge v(a) = w(a)\}$ and then we let $(C, w)\varepsilon\mu_r((B, v))$ where $r = \frac{|IND(g,g')|}{|B|}$. Thus, the degree of partialness of g' in g is determined by frequency count of identical elementary descriptors in templates (B, v) and (C, w).
Now, given individual entities being elementary granules g, g' with $[g] = \bigvee_{i=1}^{k}[(B_i, v_i)]$ and $[g'] = \bigvee_{j=1}^{m}[(C_j, w_j)]$, we let
$$Name(g)\varepsilon\mu_r^*(Name(g'))$$
where
$$r = min_{[(C_j, w_j)]}\{max_{[(B_i, v_i)]}max\{s : [(C_j, w_j)]\varepsilon\mu_s([(B_i, v_i)])\}\}.$$

Example 1. We give a simple example concerning the last method of calculating the measure μ. We begin with an example of an information system presented in Table 1.

Consider $B = \{a_1, a_2\}$, $C = \{a_2, a_3\}$, $v =< 1, 0 >$, $w =< 0, 1 >$. For $g = ((B, v), [B, v])$, $g' = ((C, w), [C, w])$, we have $IND(g, g') = \{a_2\}$ and accordingly, $Name(g')\varepsilon\mu_{0.5}^*Name((g))$.

Let us observe that rough inclusions defined above induce similarity relations in information systems: we may interpret the relation $Name(g)\varepsilon\mu_r(Name(g'))$ as the statement that g *is similar to* g' in degree r. Thus, any rough inclusion μ induces a graded family of similarity relations μ_r.

	a_1	a_2	a_3
u_1	1	0	1
u_2	1	0	0
u_3	1	1	0
u_4	0	1	1
u_5	0	1	0
u_6	1	0	1
u_7	1	1	0

Table 1. *Binary1*: An example of an information table

4.2 Rough Mereological Component of Granulation

Predicates μ_r enter in a natural way our discussion of granules. We will define a granule of knowledge in an information system (U, A) as a class of objects similar to a given X in degree r, i.e., as the object $Kl(Z(X, r))$ where $Y \varepsilon Z(X, r) \iff Y \varepsilon \mu_r(X)$. Tuning the parameter r and other hidden parameters (e.g. weights w_i in the definition of μ in Procedure 1) allows us to define granules optimal in a given application context.

We may also apply rough inclusions to measure the degree of closeness between granules e.g. to select decision rules with sufficient qualities.

We now present a calculus of granules set in the distributed environment of a multi–agent system.

5 Adaptive Calculi of Granules

We construct a mechanism for transferring granules of knowledge among agents by means of transfer functions induced by rough mereological connectives extracted from their respective information systems [14].

We first recall basic ingredients of our scheme of agents [13], [14], [21].

5.1 Distributed Systems of Intelligent Agents

We refer to a model for approximate synthesis in a distributed system proposed in [13]– [15], [16].

Consider a distributed (multi-agent) system $M_A = \{Ag, Link, Inv\}$ where Ag is a set of agents, $Link$ is a finite list of words over Ag and Inv is a set of inventory objects. Each t in $Link$ is a word $ag_1 ag_2 ... ag_k ag$ meaning that ag is the parent node and $ag_1, ag_2, ..., ag_k$ are children nodes in an elementary team t; both parties are related by means of the operation o_t which makes from a tuple $(x_1, ..., x_k)$ of objects, resp. at $ag_1, .., ag_k$ the object $o_t(x_1, .., x_k)$ at ag. Leaf agents $Leaf$ are those ag which are not any parent node. They operate on objects from Inv.

In addition, each agent ag is equipped with an information system A(ag)= $(U(ag), A(ag))$, a rough inclusion $\mu_{r,ag}$ on $U(ag)$ (cf. e.g. Procedure 1) and a set $St(ag) \subseteq U(ag)$ of *standards.*

Reasoning in M_A goes by way of standards and rough inclusions $\mu_{r,ag}$ at any ag. Instrumental in this reasoning process are *rough connectives* $f_{\sigma,t}$ where $\sigma = (st_1, ..., st_k, st)$ is a set of standards such that

1. $st_i \in St(ag_i)$ $(i = 1, 2, .., k)$, $st \in St(ag)$ and $ag_1 ag_2 ... ag_k ag = t \in Link$;
2. $o_t(st_1, .., st_k) = st$

(σ satisfying conditions 1 and 2 is called *admissible*). Any rough connective describes the process of propagation of rough inclusion values from children nodes to the parent node according to the formula

$$\forall i. x_i \varepsilon \mu_{r_i, ag_i}(st_i) \Longrightarrow o_t(x_1, .., x_k) \varepsilon \mu_{f(r_1,...,r_k), ag}(st).$$

Approximate logic of synthesis. We assume for simplicity that the distributed system M_A consists of $Ag=\{ag_1, ag_2, ag\}$ with $Link=\{ag_1 ag_2 ag\}$ and the operation o. This will not restrict the universality of our discussion but it will simplify the notation and make it easier to understand essential features of our approach. In particular, ag_1, ag_2 are (and model) leaf agents of the system M_A and ag is (and models) the head agent (which may be treated e.g. as the interface to an external client) of M_A.

We introduce a simplified logic $L(Ag)$ [14], [21] in which we can express global properties of the synthesis process.

Elementary formulae of $L(Ag)$ are of the form $< st(ag), \varepsilon(ag) >$ where $st(ag) \in St(ag)$, $\varepsilon(ag) \in [0, 1]$ for any $ag \in Ag$. Formulae of $L(ag)$ form the smallest extension of the set of elementary formulae closed under propositional connectives $\vee$, $\wedge$, $\neg$ and under the modal operators $\Box$, $\Diamond$.
For $x \in U(ag)$, we say that x satisfies a formula $< st(ag), \varepsilon(ag) >$, in symbols:

$$x \models < st(ag), \varepsilon(ag) >,$$

iff $x \varepsilon \mu_{\varepsilon(ag),ag}(st(ag))$.

Notice that $st(ag)$ may choose a formula Φ_{ag} (a choice is by no means unique) in descriptor language which it does satisfy and x has to be close enough to $st(ag)$ in order to satisfy the chosen formula Φ_{ag} in degree $\varepsilon(ag)$.

We extend satisfaction over formulae by recursion in usual way.

By a *selection* over Ag we mean a function sel which assigns to each agent ag an object $sel(ag) \in U(ag)$. For two selections sel, sel' we say that *sel induces sel'*, in symbols $sel \rightarrow_{Ag} sel'$ when $sel(a) = sel'(a)$ for $a = ag_1, ag_2$ and $sel'(ag) = o(ag)(sel'(ag_1), sel'(ag_2))$.

We extend the satisfiability predicate $\models$ to selections: for an elementary formula $< st(ag), \varepsilon(ag) >$, we let

$$sel \models < st(ag), \varepsilon(ag) > \text{ iff } sel(ag) \models < st(ag), \varepsilon(ag) > .$$

We now let $sel \models \diamond < st(ag), \varepsilon(ag) >$ when there exists a selection sel' satisfying the conditions: $sel \rightarrow_{Ag} sel'; sel' \models < st(ag), \varepsilon(ag) >$.

In terms of $L(Ag)$ it is possible to express the problem of synthesis of an approximate solution to the problem posed to Ag.

In the process of top–down communication, a requirement Ψ received by the scheme from an external source (a client) is decomposed into approximate specifications of the form $< st(ag), \varepsilon(ag) >$ for any agent ag of the scheme. The decomposition process is initiated at the agent ag and propagated along the scheme. We now are able to formulate the synthesis problem.

Synthesis problem

Given $\alpha : < st(ag)), \varepsilon(ag) >$ *find a selection sel with the property* $sel \models \alpha$.

A solution to the synthesis problem with a given formula α is found by negotiations among the agents based on uncertainty rules and their successful result can be expressed by a top–down recursion as follows.

Sufficiency Criterion

To satisfy α, *it is sufficient that each agent* ag_i *choose a standard* $st(ag_i) \in U(ag_i)$ *and a coefficient* $\varepsilon(ag_i) \in [0,1]$ *such that*

1. $\sigma = (st(ag_1), st(ag_2), st(ag))$ is admissible;
2. $f_\sigma(\varepsilon(ag_1), \varepsilon(ag_2)) \geq \varepsilon(ag)$.

We call an α - *scheme* an assignment of a formula $\alpha(ag) : < st(ag), \varepsilon(ag) >$ to each $ag \in Ag$ in such manner that 1, 2 in above Criterion are satisfied and $\alpha(ag))$ is α. We denote this scheme with the symbol $sch(\alpha)$.

We say that a selection sel is *compatible* with a scheme $sch(\alpha)$, in case $sel(a)\varepsilon\mu_{\varepsilon(a),a}(st(a))$ for $a = ag_1, ag_2$.

The goal of negotiations can be summarized now as follows.

Proposition 2. *Given a formula* $\alpha : < st(ag)), \varepsilon(ag)) >$, *if a selection sel is compatible with a scheme* $sch(\alpha)$ *then* $sel \models \diamond\alpha$.

Proposition 2 provides a sufficient condition for leaf agents (here, ag_1, ag_2) in terms of closeness (measured by μ at respective agents) to standards in Inv which would guarantee that the system Ag does satisfy a specification Ψ issued to its head.

6 Calculi of Elementary Granules

We construct in a given system M_A of agents for each agent ag, granules by means of a rough inclusion $\mu_{r,ag}$ of the agent ag.

6.1 Rough Mereological Granules

For a standard $st(ag)$ and a parameter $\varepsilon(ag)$, we denote by the symbol

$$gr(st(ag), \varepsilon(ag))$$

(*the granule of size* $\varepsilon(ag)$ *about* $st(ag)$) the class $Kl_{\varepsilon(ag)}(st(ag))$ of those x for which $x\varepsilon\mu_{\varepsilon(ag),ag}(st(ag))$.

Example 2

We use a hybrid strategy and we define the rough inclusion $\mu_{r,ag}$ according to Procedure 1 with $w_1 = 1$, i.e.,

$$x\varepsilon\mu_{r,ag}(y) \text{ if and only if } r \leq \frac{|IND(x,y)|}{|A_{ag}|}.$$

For $gr(B,v) = ((B,v),[(B,v)])$, an atomic elementary granule, by $st(ag)_{B,v}$ we denote a standard in $gr(B,v)$, i.e., an atomic elementary granule $gr(A,u) = ((A,u),[(A,u)])$ where $u \in INF(A)$ and $u \mid B = v$.

We define the rough inclusion $\mu_{r,ag}$ by

$$x\varepsilon\mu_{r,ag}(y) \Longleftrightarrow r \leq \frac{|\{a \in A_{ag} - B : a(x) = a(y)\}|}{|A_{ag}|}$$

where x, y denote names of atomic elementary granules of the form $gr(A,u) = ((A,u),[(A,u)])$ where $u \in INF(A)$ and $u \mid B = v$.

Thus, given ε, the granule $gr(st(ag)_{B,v}, \varepsilon)$ consists of those x which agree with $st(ag)_{B,v}$ on B and additionally agree with this standard on at most $\varepsilon \times 100$ percent of the remaining attributes. This is in agreement with application contexts where one does search for short templates with a high classification quality.

Granules $gr(st(ag)_{B,v}, \varepsilon)$ provide a covering of the granule $gr(B,v)$ by similarity classes of $\mu_{\varepsilon,ag}$.

6.2 Synthesis in Terms of Granules

We say that $gr(st(ag), \varepsilon(ag))$ satisfies a formula $\alpha :< st(ag), \varepsilon'(ag) >$, in symbols

$$gr(st(ag), \varepsilon(ag)) \models \alpha$$

in case $\varepsilon(ag) \geq \varepsilon'(ag)$.

Given an admissible $\sigma = \{st(ag_1), st(ag_2), st(ag)\}$ and $\varepsilon(ag), \varepsilon(ag_1), \varepsilon(ag_2)$ with $f_\sigma(\varepsilon(ag_1), \varepsilon(ag_2)) \geq \varepsilon(ag)$ (i.e., conditions 1 and 2 in Criterion are satisfied) we observe that:

if

$$x \in gr(st(ag_1), \varepsilon(ag_1)), y \in gr(st(ag_2), \varepsilon(ag_2))$$

then

$$o(x,y) \in gr(st(ag), \varepsilon(ag)).$$

We may state the sufficiency of synthesis condition in terms of granules as follows.

Proposition 3. *For a formula* $\alpha :< st(a), \varepsilon(a) >$

if $x \in gr(st(ag_1), \varepsilon(ag_1))$, $y \in gr(st(ag_2), \varepsilon(ag_2))$

with conditions 1, 2 in Criterion satisfied

then $o(x, y) \models \alpha$.

It thus suffices that for a given granule $gr(st(ag), \varepsilon(ag))$*, agents* ag_1*,* ag_2 *send to ag granules, respectively,* $gr(st(ag_1), \varepsilon(ag_1))$ *and* $gr(st(ag_2), \varepsilon(ag_2))$*.*

Proposition 3 paraphrases Proposition 2 in terms of granules (i.e. similarity sets) at agents and gives sufficient conditions for leaf agents in terms of granules to fulfill by the system Ag the specification Ψ.

7 Associated Synthesis Grammars

The above may be formulated in terms of a grammar Γ and a language $L(\Gamma)$ whose words code sufficient synthesis conditions [15]. With each agent $ag \in Ag$, we associate a grammar $\Gamma(ag) = (N(ag), T(ag), P(ag))$. To this end, we assume that a finite set $\Xi(ag) \subset [0, 1]$ is selected for each ag. We let $N(ag) = \{(s_{st(ag)}, t_{\varepsilon(ag)}) : st(ag) \in St(ag), \varepsilon(ag) \in \Xi(ag)\}$ where $s_{st(ag)}$ is a non–terminal symbol corresponding in a one – to – one way to the standard $st(ag)$ and similarly $t_{\varepsilon(ag)}$ corresponds to $\varepsilon(ag)$.

The set of terminal symbols $T(ag)$ is defined for ag by letting

$$T(ag) = \cup_{i=1,2}\{\{(s_{st(ag_i)}, t_{\varepsilon(ag_i)}) : st(ag_i) \in St(ag_i); \varepsilon(ag_i) \in \Xi(ag_i)\} : i = 1, 2\}.$$

The set of productions $P(ag)$ contains productions of the form

$$(s_{st(ag)}, t_{\varepsilon(ag)}) \longrightarrow (s_{st(ag_1)}, t_{\varepsilon(ag_1)})(s_{st(ag_2)}, t_{\varepsilon(ag_2)})$$

where $st(ag_1), st(ag_2), st(ag)$, $\varepsilon(ag), \varepsilon(ag_1), \varepsilon(ag_2)$ satisfy conditions 1 and 2 of Sufficiency Criterion.

We define a grammar system

$$\Gamma = (T(ag), (\Gamma(a) : a = ag \vee a = Input), S)$$

by introducing an additional agent $Input$ with the non–terminal symbol S, terminal symbols of $Input$ being non–terminal symbols of ag and productions of $Input$ of the form: $S \Longrightarrow (s_{st(ag))}, t_{\varepsilon(ag)})$.

The meaning of S is that it does code an approximate specification (requirement) for an object to be synthesized; productions of $Input$ code specifications for approximate solutions in the language of the agent ag. Subsequent re–writings produce terminal strings of the form

$$(s_{st(ag_1)}, t_{\varepsilon(ag_1)})(s_{st(ag_2)}, t_{\varepsilon(ag_2)}).$$

Proposition 4. *Suppose* $(s_{st(ag_1)}, t_{\varepsilon(ag_1)})(s_{st(ag_2)}, t_{\varepsilon(ag_2)})$ *is obtained from*

$$S \longrightarrow (s_{st(ag))}, t_{\varepsilon(ag))})$$

by subsequent re–writing by means of productions in Γ. *Then given any selection sel with*

$$sel(ag_i)\varepsilon\mu_{\varepsilon(ag_i),ag_i}(st(ag_i))$$

for $i = 1, 2$ *we have*

$$sel \models \diamond < st(ag), \varepsilon(ag) > .$$

Remarks

1. Synthesis grammars constructed above reflect processes in multi-agent systems which arise in a multi-agent system involved in cooperation, negotiation and conflict–resolving actions when attempting to provide a solution to a specification of a problem posed to its root
2. Complexities of membership problems for languages generated by synthesis grammars may be taken ex definitione as complexities of the underlying synthesis processes

8 Toward Granules of Classifying Rules and Classifying Algorithms

Classifying (decision) rules may be represented as pairs of granules.

Therefore, we may apply to classifying (decision) rules the formula of Section 4.1. Consider atomic elementary granules g, g' and assume a rough inclusion μ_r is given over atomic elementary granules of the form (A, v). Then:

$$Name(g)\varepsilon\mu_r^*(Name(g'))$$

if and only if

$$min_{x\varepsilon Name(g)} max_{y\varepsilon Name(g')}\{s : x\varepsilon\mu_s(y)\} \geq r.$$

This defines a rough inclusion μ_r^*.

Example 3

Consider templates (B, v) and (C, w) in an information system $\mathsf{A} = (U, A)$ with the rough inclusion μ defined in Procedure 1 with $w_1 = 1$.

It may be found straightforwardly that for $g = gr(B, v)$, $g' = gr(C, w)$, we have the worst–case estimate

$$Name(g)\varepsilon\mu_r^*(Name(g')) \Longleftrightarrow r \leq 1 - \frac{k - s}{n}$$

where $k = |C|$, s–the number of indices j with $j\varepsilon B \cap C$, $v_j = w_j$.

We extend this measure to pairs of granules, i.e., classifying rules by letting:

$$Name(g, g_1)\varepsilon\mu^{g*}_{\top(r,s)}(Name(g', g'_1))$$

if and only if

$$Name(g)\varepsilon\mu^*_r(Name(g')) \wedge g_1\varepsilon\mu^*_s(Name(g'_1))$$

where $\top$ is a *t-norm* (e.g. *min* operation, see e.g. [14]) and $Name(g_1, g_2)$ is $(Name(g_1), Name(g_2))$.

Then $\mu^*_{\top(r,s)}$ is a rough inclusion on classifying rules. Observe that the degree of inclusion on pairs is calculated as the value of a chosen *t–norm* on values of inclusions of respective elements in the pair.

8.1 Rough mereological connectives on classifying rules

Given our system M_A, rough mereological connectives f may be extended to connectives propagating uncertainty about classifying rules. Although it would be possible to derive a general formula from our results above, we would rather give an example.

Example 4. Fusion of data

Consider M_A with ag_1, ag_2, ag where information systems of agents are $A_{ag_1} = (U, A_1)$, $A_{ag_2} = (U, A_2)$ where A_1 and A_1 are disjoint, $| A_1 |= n_1$, $| A_2 |= n_2$, and $A_{ag} = (U, A_1 \cup A_2)$, i.e., ag fuses tables of ag_1 and ag_2 by merging their attribute sets. We have $Name(gr(B, v)) = (B, v)$ for any template (B, v) in A.

We find a formula for f, the rough mereological connective for granules generated by templates.

Consider thus templates (B_1, v_1), (C_1, w_1) at ag_1, (B_2, v_2), (C_2, w_2) at ag_2 and $(B = B_1 \cup B_2, v = v_1 \cup v_2)$ and $(C = C_1 \cup C_2, w = w_1 \cup w_2)$ at ag along with the corresponding elementary granules for μ as in Example 3.

We have by Example 3 that :

1. $(B_1, v_1)\varepsilon\mu^*_{r_1}((C_1, w_1))$ with $r_1 \leq 1 - \frac{k_1 - s_1}{n_1}$;
2. $(B_2, v_2)\varepsilon\mu^*_{r_2}((C_2, w_2))$ with $r_2 \leq 1 - \frac{k_2 - s_2}{n_2}$;
3. $(B, v)\varepsilon\mu^*_r((C, w))$ with $r \leq 1 - \frac{(k_1 - s_1) + (k_2 - s_2)}{n_1 + n_2}$.

From the above conditions 1–3 it follows that the connective f satisfies the formula

$$f(1 - \varepsilon_1, 1 - \varepsilon_2) \geq 1 - max(\varepsilon_1, \varepsilon_2) = min(1 - \varepsilon_1, 1 - \varepsilon_2).$$

An Application: a classifier query decomposition. In the setting of Example 4, we may select the template (C, w). Then the query

$$? < (.,.), (C, w), \delta >$$

means that we are searching for (B, v) with $(B, v)\varepsilon\mu^*_\delta((C, w))$, i.e., for a classifier with a sufficient quality (measured by δ).

Then, the problem may be decomposed: it suffices that ag_1, ag_2 find granules $gr(B_1, v_1)$, $gr(B_2, v_2)$, respectively, satisfying:

1. $(B_1, v_1)\varepsilon\mu^*_{1-\varepsilon_1}((C_1, w_1))$;
2. $(B_2, v_2)\varepsilon\mu^*_{1-\varepsilon_2}((C_2, w_2))$

with $max(\varepsilon_1, \varepsilon_2) \leq 1 - \delta$.

The results presented here allow to

1. formulate counterparts of granule calculi for granules of classifiers;
2. define synthesis grammars in terms of classifiers;
3. extend these results to classification algorithms as these are finite collections of classifiers so one more application of our results would yield appropriate formulae in case of classification algorithms.

In consequence of the above results, we may build synthesis schemes in terms of constructs defined in languages of various levels: from the least abstract language of object granules to the most abstract language of decision algorithms granules.

9 A Neural Model of Rough Mereological Computation

There is a parallelism between the proposed above calculi of granules in distributed systems and neural computing. Let us point to some analogies.

1. Any elementary team of agents $t = ag$ may be regarded as a model of a neuron with inputs $ag_1, ag_2, ..., ag_k$, the output ag, and a parameterized family of activation functions represented as rough connectives $f_{\sigma,t}$ where $\sigma = (st_1, ..., st_k, st)$ is an admissible set of standards
2. Values of rough inclusions in the formula

$$\forall i.x_i\varepsilon\mu_{r_i,ag_i}(st_i) \Longrightarrow o_t(x_1, .., x_k)\varepsilon\mu_{f(r_1,...,r_k),ag}(st)$$

   are counterparts of weights in a traditional neural network. Let us observe that in our case the resulting network is a parameterized system of simple networks, indexed by synthesis schemes (α–schemes)
3. Learning in this new kind of a neural network is based also on back-propagation mechanisms in which the incoming signal (a specification Ψ) is assigned a proper α–scheme and a proper set of weights set in negotiation and cooperation processes among local teams and agents therein

These processes of learning would require new algorithms and one possible way out here is to base the process of learning on familiar techniques of neural networks by encoding all the constructs in a neural network whose activation functions are tractable (e.g. piece–wise differentiable) approximations to rough mereological connectives. As a result, we would obtain a closed–loop system providing feedback information from the distributed system to the neural network. The theory and practice of such systems is to come in future. In [18] ideas of rough–neuro computing based on approximation spaces [22], [25] are proposed.

10 Conclusions

The reader of this work, after a scrutiny of the content, has undoubtedly noticed and made themselves familiar with the main ingredients of this approach: the formalism based on mereological ideas, the construction of reasoning schemes, the granule–forming mechanism, the analysis of correctness of problem solving procedure along with additional aspects i.e. the linguistic counterpart given in the guise of synthesis grammars and languages and rough–neurocomputing scheme.

In this way we propose to realize programs of *granulation of knowledge* and *computation with granules*. At the same time we propose a way toward *computing with words*: interfaces between natural language corpora and synthesis languages will allow for performing computing with words *sensu stricto*.

Clearly, further intensive research is necessary to bring to life real systems working on the principles exposed in this paper.

Acknowledgement *L. Polkowski has been supported by the Research Grant No 8T11C 024 17 from the State Committee for Scientific Research (KBN) of the Republic of Poland. A. Skowron has been supported by the Research grant T11C02519 from the State Committee for Scientific Research of the Republic of Poland and a grant from the Wallenberg Foundation.*

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo, 1996, Fast discovery of association rules, in: V. M. Fayad, G. Piatetsky–Shapiro, P. Smyth and R. Uthurusamy (eds.), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Cambridge, MA, pp. 307–328.
2. N.Asher, L. Vieu, 1995, Toward a geometry of commonsense: a semantics and a complete axiomatization of mereotopology, in: *Proceedings* IJCAI'95, Morgan Kauffman, San Mateo, CA, pp. 846-852.
3. J. Bazan, Hung Son Nguyen, Sinh Hoa Nguyen, P. Synak, and J. Wróblewski, 2000, Rough set algorithms in classification problem, in: [19], to appear.
4. Sinh Hoa Nguyen, 2000, Regularity analysis and its applications in Data Mining, in: [19], to appear.
5. S. Leśniewski, 1927, O podstawach matematyki (in Polish), *Przegląd Filozoficzny*, 30, pp.164-206; 31, pp. 261-291; 32, pp. 60-101; 33, pp. 77-105; 34, pp. 142-170.
6. S. Leśniewski, 1982, On the foundations of mathematics, Topoi 2, pp. 7–52.
7. H. Liu, H. Motoda, 1998, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Kluwer Academic Publishers, Boston.
8. J. Łukasiewicz, 1913, Die Logischen Grundlagen der Wahrscheinlichtkeitsrechnung, Kraków; also in: L. Borkowski (ed.), 1970, *Jan Łukasiewicz–Selected Works*, North Holland Publ. Co. and Polish Scientific Publ., Amsterdam – Warsaw.
9. Z. Pawlak, 1992, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht.

10. Z. Pawlak and A. Skowron, 1994, Rough membership functions, in: R.R. Yaeger, M. Fedrizzi, J. Kacprzyk (eds.), *Advances in the Dempster Shafer Theory of Evidence*, John Wiley and S., New York, pp. 251-271.
11. Z. Pawlak, 1999, Decision rules, Bayes' rule and rough sets, in: Ning Zhong, A. Skowron, S. Ohsuga (eds.), *New Directions in Rough Sets, Data Mining and Granular-Soft Computing, Lecture Notes in Artificial Intelligence* vol. 1711, Springer-Verlag, Berlin, pp. 1–9.
12. S. K. Pal, A. Gosh, M.K. Kundu, 2000, Soft Computing for Image Processing, in the Series: *Studies in Fuzziness and Soft Computing*, J. Kacprzyk (ed.), Physica-Verlag/ Springer-Verlag, Heidelberg.
13. L. Polkowski and A. Skowron, 1994, Rough mereology, *Lecture Notes in Artificial Intelligence*, vol. 869, Springer-Verlag, Berlin, pp. 85-94.
14. L. Polkowski and A. Skowron, 1996, Rough mereology: a new paradigm for approximate reasoning, *International J. Approximate Reasoning* 15(4), pp. 333-365.
15. L. Polkowski and A. Skowron, 1998, Grammar systems for distributed synthesis of approximate solutions extracted from experience, in: Gh. Paun and A. Salomaa (eds.), *Grammatical Models of Multi-Agent Systems*, Gordon and Breach Sci. Publ., Amsterdam, pp. 316-333.
16. L. Polkowski and A. Skowron (eds.), 1998, *Rough Sets in Knowledge Discovery*, vols. 18, 19 in the Series: *Studies in Fuzziness and Soft Computing*, J. Kacprzyk (ed.), Physica-Verlag/Springer-Verlag, Heidelberg.
17. L. Polkowski and A. Skowron, 1999, Calculi of granules based on rough set theory: Approximate distributed synthesis and granular semantics for computing with words, in: N. Zhong, A. Skowron , and S. Ohsuga (eds.), *New Directions in Rough Sets, Data Mining, and Granular Soft Computing, Lecture Notes in Artificial Intelligence* vol. 1711, Springer-Verlag, Berlin, pp. 20–28.
18. L.Polkowski and A. Skowron, 2000, Rough–Neuro Computing, *Proceedings* RSCTC'2000, Banff, Canada, to appear.
19. L. Polkowski, S. Tsumoto, and T. Y. Lin (eds.), 2000, *Rough Set Methods and Applications. New Developments in Knowledge Discovery in Information Systems*, in the Series: *Studies in Fuzziness and Soft Computing*, J. Kacprzyk (ed.), Physica-Verlag/ Springer-Verlag, Heidelberg, in print.
20. P. Simons, 1987, Parts - A Study in Ontology, Clarendon, Oxford.
21. A. Skowron and L. Polkowski, 1998, Rough mereological foundations for design, analysis, synthesis and control in distributed systems, *Information Sciences. An Intern. J.* 104(1-2), pp. 129-156.
22. A. Skowron, J. Stepaniuk, 1996, Tolerance approximation spaces, *Fundamenta Informaticae* 27(2-3), pp. 245-253.
23. A. Skowron, J. Stepaniuk, S. Tsumoto, 1999, Information granules for spatial reasoning, *Bulletin of the International Rough Set Society* 3/4, pp. 147–154.
24. R. Słowiński and D. Vanderpooten, 2000, A generalized definition of rough approximations based on similarity, *IEEE Transactions on Data and Knowledge Engineering*, to appear.
25. J. Stepaniuk, 2000, Knowledge Discovery by Application of Rough Set Models, in: [19], to appear.
26. S. Tsumoto, 1999, Induction of expert decision rules using rough sets and set–inclusion, in: S. K. Pal and A. Skowron (eds.), *Rough–Fuzzy Hybridization. A New Trend in Decision Making*, Springer–Verlag, Singapore, pp. 316–329.
27. L.A. Zadeh, 1973, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-3,

pp. 28–44.

28. L.A. Zadeh, 1996, Fuzzy logic = computing with words, *IEEE Trans. on Fuzzy Systems* 4, pp. 103-111.
29. L. A. Zadeh, 1997, Toward a theory of fuzzy information granulation and its certainty in human reasoning and fuzzy logic, *Fuzzy Sets and Systems* 90, pp. 111-127.
30. L. A. Zadeh and J. Kacprzyk (eds.), 1999, *Computing with Words in Information/Intelligent Systems 1. Foundations*,Physica–Verlag/Springer–Verlag, Heidelberg.
31. L. A. Zadeh, 1999, Fuzzy logic = Computing with w ords, in [22], pp. 3–23.