Approximate Reasoning in Distributed Environments

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

Abstract. Information sources provide us with granules of information that must be transformed, analyzed and built into structures that support problem solving. Lotfi A. Zadeh has recently pointed out to the need to develop a new research branch called Computing with Words and Perceptions (CWP). One way to achieve CWP is through Granular Computing (GC). The main concepts of GC are related to information granule calculi. One of the main goals of information granule calculi is to develop algorithmic methods for construction of complex information granules from elementary ones by means of available operations and inclusion (closeness) measures. These constructions can also be interpreted as approximate schemes of reasoning (AR-schemes). The constructed complex granules represent a form of information fusion. Such constructed granules should satisfy some constraints like quality criteria or/and degrees of granule inclusion in (closeness to) a given information granule. In the chapter we discuss the idea of the rough neurocomputing paradigm for inducing AR-schemes based on rough sets and, in particular, on rough mereology. Information granule decomposition methods are important components of methods for AR-schemes induced from data and background knowledge. We report some recent results on information granule decomposition.

Keywords: approximate reasoning by agents, rough sets, rough mereology, information granulation, pattern, granular computing, approximate reasoning schemes , decomposition, information fusion, rough neuron, rough neurocomputing

1 Introduction

Lotfi A. Zadeh has recently pointed out to necessity of developing a new research branch called Computing with Words and Perceptions (CWP) (see, e.g., [58, 60, 59]). The goal of this new research direction is to build foundations for future intelligent computers and information systems performing computations on words rather than on numbers. In this new paradigm different soft computing tools like neural networks, fuzzy sets, rough sets or genetic algorithms should work in a complementary not in a competitive fashion. A great challenge is to develop the foundations for this new computing paradigm and to show that they can help to demonstrate new applications.

Information granulation belongs to intensively studied topics in soft computing (see, e.g., [58, 60, 59, 17, 54, 36, 25]). One of the recently emerging approaches to deal with information granulation is based on information granule calculi (see, e.g., [35, 40, 41, 45, 24]). The development of such calculi is important for making progress in many areas like object identification by autonomous systems (see, e.g., [4, 56]), Web mining (see, e.g., [13]) or spatial reasoning (see, e.g., [55, 6, 7]). In particular, reasoning methods using background knowledge as well as knowledge extracted from experimental data (e.g., sensor measurements) represented by concept approximations [4] are important for making progress in such areas. Moreover, such calculi are important for the development of sensor fusion strategies (see, e.g., [3, 8, 29, 30]). One should take into account that modeling complex phenomena entails the use of local models (captured by local agents) which next should be fused. This process involves the negotiations between agents [12] to resolve contradictions and conflicts in local modeling. This kind of modeling will become more and more important in solving complex real-life problems which we are unable to model using traditional analytical approaches. The latter approaches lead to exact models; however, the necessary assumptions used to create them are causing the resulting solutions to be *too far* from reality to be accepted (see Figure 1).

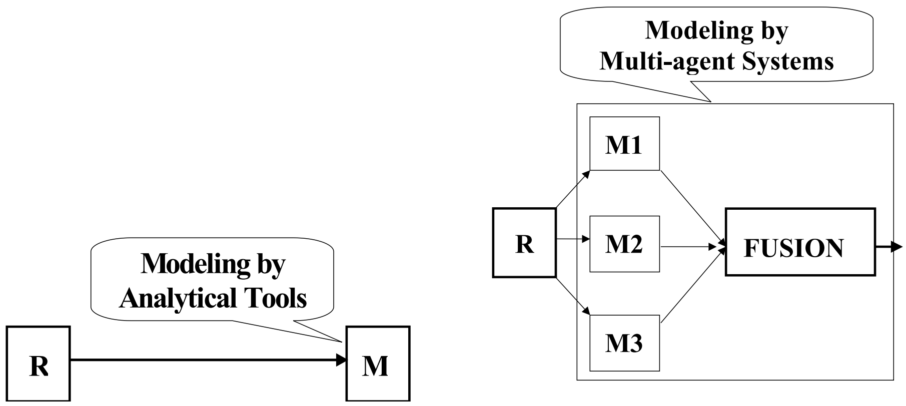

Figure 1: Traditional analytical vs multi-agent systems modeling (R is for reality, M,M1,M2 denote models)

One way to achieve CWP is through Granular Computing (GC). The main concepts of GC are related to information granulation and in particular to information granules. Information granules, due to Zadeh [58], are clumps of objects (points) which are drawn together by indistinguishability, similarity or functionality. Several approaches concerning formulation of information granule concept have been proposed.

Any approach to information granulation should make it possible to define complex information granules (e.g., in spatial and temporal reasoning, one should be able to determine if the situation on the road (see Figure 2) is safe on the basis of sensor measurements [55] or to classify situations in complex games, like soccer [50]. These complex information granules consitute a form of information fusion. Any calculus of complex information granules should allow to: (i) deal with vagueness of information granules; (ii) develop strategies of inducing multi-layered schemes of complex granule construction; (iii) construct robust information granules with respect to deviations of granules from which they are constructed; (iv) develop adaptive strategies for reconstruction of induced schemes of complex information granule synthesis.

To deal with vagueness, one can adopt fuzzy set theory [57] or rough set theory [27] either separately or in combination [22]. The second requirement is related to the problem of understanding of reasoning from measurements to perception (see, e.g., [59]) and to con-

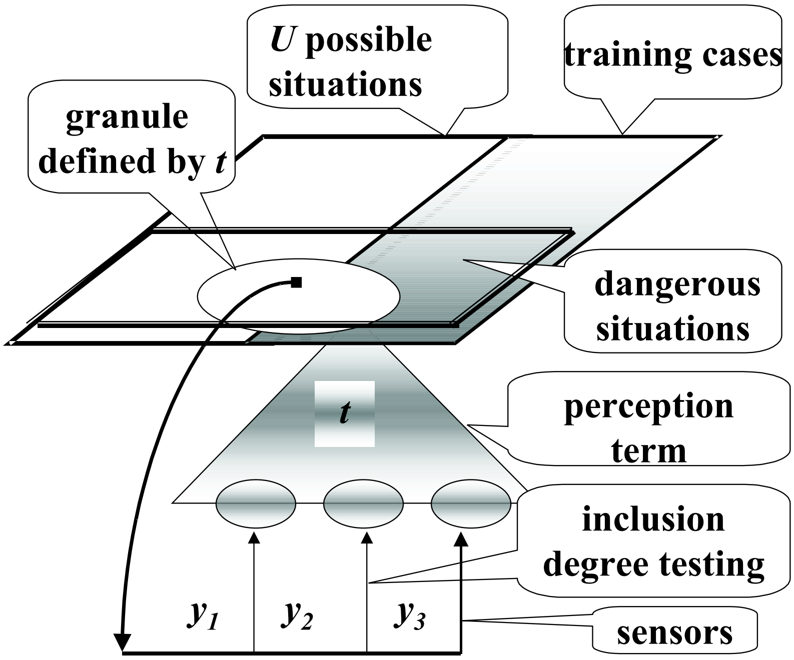

Figure 2: Classification of situations

cept approximation learning in layered learning [50] as well as to fusion of information from different sources (see., e.g., [58, 60, 59]). The importance of searching for Approximate Reasoning Schemes (AR-schemes, for short) as schemes of new information granule construction, is stressed also in rough mereology. In general, this leads to hierarchical schemes of new information granule construction. This is closely related to ideas of co-operation and conflict resolution in multi-agent systems [12]. Among important topics studied are methods for specifying operations on information granules; in particular, for their construction from data and background knowledge, and methods for inducing the hierarchical schemes of information granule construction. One of possible approaches is to learn such schemes using evolutionary strategies. Robustness of the scheme means that any scheme produces rather a higher order information granule that is a clump (e.g., a set) of close information granules rather than a single information granule. Such a clump is constructed, e.g., by means of the scheme from the Cartesian product of input clumps (e.g., clusters) satisfying some constraints. The input clumps are defined by deviations (up to acceptable degree) of input information granules. Using multi-agent terminology, let us also observe that local agents perform operations on information granules from granule sets ”understandable” by them. Hence, granules submitted as arguments by other agents should be approximated by means of properly tuned approximation spaces creating interfaces between agents. These interfaces can be, in the simplest case, constructed on the basis of exchanged information about agents stored in the form of decision data tables. From these tables the approximations of concepts can be constructed using rough set approach [44]. In our model we assume that for any agent ag and its operation $o(ag)$ of arity n there are approximation spaces $AS_1(o(ag), in), \ldots, AS_n(o(ag), in)$ which will filter (approximately) the granules received by the agent for performing the operation $o(ag)$. In turn, the granule sent by the agent after performing the operation is filtered (approximated) by the approximation space $AS(o(ag), out)$. These approximation spaces are parameterized with parameters allowing to optimize the size of neighborhoods in these spaces as well as the inclusion relation [36] using as a criterion for optimization the granule approximation quality. Approximation spaces attached to an operation correspond to neuron weights in neural networks whereas the operation performed by the agent corresponds to the operation realized on the vector of real numbers by the neuron. The generalized scheme of agents is returning a granule in response to input information granules. It can be for example a cluster of elementary granules. Hence, our schemes (being extensions of schemes for synthesis of complex objects (or granules) developed in [35] and [33]) realize much more general computations then neural networks operating on vectors of real numbers. The question, if such schemes can be efficiently simulated by classical neural networks is open. We would like to call extended AR-schemes for complex object construction *rough neuroschemes* (for complex object construction). The stability of such schemes corresponds to the resistance to noise of classical neural networks.

The methods of inducing AR-schemes transforming information granules into information granules studied using rough set methods in hybridization with other soft computing approaches create a core for Rough Neurocomputing (RNC) (see, e.g., [23, 37, 48, 24]).

Another important problem concerns relationships between information granules and words (linguistic terms) in a natural language and also a possibility to induce AR-schemes as schemes approximating reasoning in natural language. This can strengthen the links between RNC and CWP. It is of a great importance for many applications. For example, in case of Web mining (see Figure 3) one is interested to extract documents relevant for the user query or dialog with the user. Hence, e.g., the problem arises, how to construct information granule describing a clump of documents the most relevant to the user query.

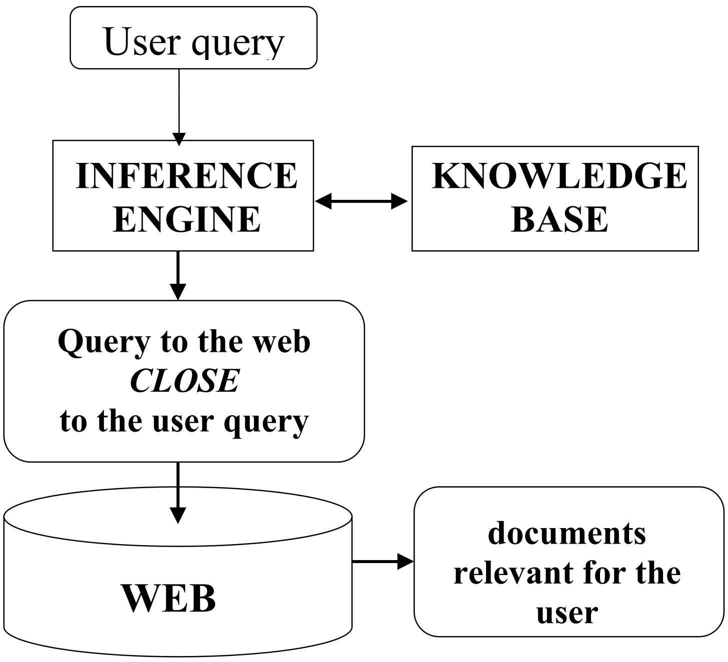

Figure 3: Web mining

Among approaches to information granulate calculi is RNC attempting to define information granules using rough sets and rough mereology (introduced to deal with vague concepts) in hybridization with methods of constructing more complex information granules by schemes analogous to neural networks.

In RNC, computations are performed on information granules. Building the foundations of RNC requires the theory of artificial neural networks, rough set theory [27] and its extensions like rough mereology (see, e.g. [31, 32, 33, 36, 37, 38]) as well as some tools created by hybridization with other soft computing approaches, in particular with fuzzy set theory [22, 57, 60] and evolutionary programming [15, 23].

In developing of RNC, special role have different hybrid methods using soft computing tools used to induce the robust AR-schemes for complex information granule construction and object classification as well as methods based on integration of rough sets with neural network techniques because they are crucial for developing the theory of RNC for synthesis of approximate schemes of reasoning.

We outline a rough neurocomputing model as a basis for GC. Our approach is based on rough sets, rough mereology and information granule calculus.

Rough Mereology [32, 35] is a paradigm allowing for a synthesis of main ideas of two paradigms for reasoning under uncertainty: fuzzy set theory and rough set theory. We present applications of rough mereology to the important theoretical idea put forth by Lotfi Zadeh [58, 59], i.e., granularity of knowledge by presenting the idea of the rough neurocomputing paradigm.

Information granule decomposition methods are important components of methods for AR-scheme inducing from data and background knowledge. In the chapter we discuss some information granule decomposition methods.

AR-schemes are obtained by means of relevant patterns for a given task decomposition of the identified or classified complex objects. The problem of deriving such schemes is closely related to perception [10, 2, 59, 53, 39].

In the chapter we assume AR-schemes define parameterized operations on information granules. We discuss problems of tuning these parameters to derive from them relevant granules included in (or close to) target concepts to a satisfactory degree. Target concepts are assumed to be incompletely specified and/or vague.

We emphasize an important property of GC related to the necessity of lossless compression tuning for complex object constructions. It means that we map a cluster of constructions into one representation. Any construction in the cluster is delivering objects satisfying the specification to a satisfactory degree if only input objects to synthesis are sufficiently close to selected standards (prototypes). In rough mereological approach clusters of constructions are represented by the so–called stable AR-schemes (of co–operating agents), i.e., schemes robust to some deviations of parameters of transformed granules. In consequence, the stable AR-schemes are able to return objects satisfying to a satisfactory degree the specification not only from standard (prototype) objects but also from objects sufficiently close to them [32, 33]. In this way any stable scheme of complex object construction is a representation of a cluster of similar constructions from clusters of elementary objects.

One can distinguish two kinds of considered parts (represented, e.g., by sub-formulas or sub-terms) of AR-schemes. Parts of the first type are represented by expressions from a language, called the *domestic* language L_d, that has known semantics (consider, for example, a given information system [27]). Representations of parts of the second type of AR-scheme are from a language, called *foreign* language L_f (e.g., natural language, that has semantics definable only in an approximate way (e.g., by means of patterns extracted using rough, fuzzy, rough–fuzzy or other approaches). For example, the parts of the second kind of scheme can be interpreted as soft properties of sensor measurements [4].

For a given expression e, representing a given scheme that consists of sub-expressions from L_f we propose to search for relevant approximations in L_d of the foreign parts from L_f and next to derive global patterns from the whole expression after replacing the foreign parts by their approximations. This can be a multilevel process, i.e., we are facing problems of discovered pattern propagation through several domestic-foreign layers.

Let us consider some strategies of patterns construction from schemes. The first strategy entails searching for relevant approximations of parts using a rough set approach. This means that each part from L_f can be replaced by its lower or upper approximation with respect to a set B of attributes. The approximation is constructed on the basis of relevant data table [4]. With the second strategy parts from L_f are partitioned into a number of sub-parts corresponding to cuts (or the set theoretical differences between cuts) of fuzzy sets representing vague concepts and each sub-part is approximated by means of rough set methods. The third strategy is based on searching for patterns sufficiently included in foreign parts. In all cases, the extracted approximations replace foreign parts in the scheme and candidates for global patterns are derived from the scheme obtained after the replacement. Searching for relevant global patterns is a complex task because many parameters should be tuned, e.g., the set of relevant features used in approximation, relevant approximation operators, the number and distribution of objects from the universe of objects among different cuts and so on. One can use evolutionary techniques for relevant pattern searching to obtain optimal parameters with respect to the quality of synthesized patterns.

We propose an approach for extracting from data patterns relevant to a given target concept. The approach is based on information granule decomposition strategies. It is shown that the discussed strategies can be based on the developed rough set methods for decision rules generation and Boolean reasoning [14]. We discuss in particular methods for decomposition which can be based on background knowledge. In [20, 45], the reader can find another approach to decomposition.

The chapter is structured as follows. Section 2 includes examples of information granules and operations on granules. In particular, in Section 2.4 we discuss parameterized rough and fuzzy information granules. Sections 3, 4 are dedicated to rough neurocomputing paradigm. In Section 5 methods for decomposition of information granules are outlined.

The chapter summarizes and extends some results presented in our previous works, presented in particular in [31, 32, 33, 34, 35, 36, 37, 38, 43, 44, 45, 46, 47, 49].

2 Information Granules

2.1 Rough Sets and Approximation Spaces

We recall general definition of approximation space [43].

A *parameterized approximation space* (with parameters $\#$ and $\$$) is a system

$$AS_{\#,\$} = (U, I_{\#}, \nu_{\$}) \tag{1}$$

where

- U is a non-empty set of objects;

- $I_{\#} : U \to P(U)$, where $P(U)$ denotes the powerset of U, is an *uncertainty function*;
- $\nu_{\$} : P(U) \times P(U) \to [0,1]$ is a rough *inclusion function*.

We write I, ν instead of $I_{\#}, \nu_{\$}$ to simplify notation. If $X = \{x\}$ and $Y = \{y\}$ we also write $I(x)$ and $\nu(x,y)$ instead of $I(X)$ and $\nu(X,Y)$, respectively. If $p \in [0,1]$ then $\nu_p(x,y)$ ($\underline{\nu}_p(x,y)$) denotes the following condition $\nu(x,y) \geq p$ ($\nu(x,y) \leq p$) holds.

The uncertainty function defines for every object x a set of similarly described objects, i.e., the *neighborhood* $I_{\#}(x)$ of x. A constructive definition of uncertainty function can be based on the assumption that some metrics (distances) are given on attribute values. By ν_{SRI} we denote the *standard rough inclusion* function defined by

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

A set $X \subseteq U$ is *definable in* $AS_{\#,\$}$, if it is a union of some values of the uncertainty function.

The inclusion function defines the degree of inclusion between two subsets of U [43]. The inclusion function definition has been generalized in rough mereology to the *rough inclusion*.

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

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

$$\begin{aligned} LOW\left(AS_{\#,\$}, X\right) &= \{x \in U : \nu_{\$}\left(I_{\#}(x), X\right) = 1\} \\ UPP\left(AS_{\#,\$}, X\right) &= \{x \in U : \nu_{\$}\left(I_{\#}(x), X\right) > 0\}, \text{ respectively.} \end{aligned} \quad (2)$$

The set approximations can be defined in a more soft way by allowing to classify an object x to the lower approximation of X if the neighborhood $I_{\#}(x)$ is included in X at least to a given a priori satisfactory degree $p \in (0,1]$ and to the complement of X if $I_{\#}(x)$ is included in X at most to a degree $q < p$ where q is also given a priori degree. Moreover, one can observe that the neighborhood $I_{\#}(x)$ can be usually defined as a collection of close objects, i.e., it can be defined using the rough inclusion. Sets of objects being definable in a given language collections of objects from data table can be treated as examples of information granules. We can conclude then that the very primitive notions are information granules and inclusion (closeness) relations between them. This has been the starting point for investigating rough mereology.

Approximations of concepts (sets) are constructed on the basis of background knowledge. Obviously, concepts are also related to new (unseen) objects. Hence it is very useful to define parameterized approximations with parameters tuned in the searching process for approximations of concepts. This idea is crucial for methods of construction of concept approximations in particular for rough set methods. In our notation $\#, \$$ denote vectors of parameters which can be tuned in the process of concept approximation. For the discussion on rough sets in inductive reasoning the reader is referred to [42].

Let us now recall some basic definitions [14]. If $IS = (U, A)$ is an *information system* then (a, v) denotes a *descriptor* defined by the attribute a and its value v, α denotes Boolean combination of descriptors, and $[\alpha]_{IS}$ (or $[\alpha]_A$) its meaning in IS, i.e., the set of all objects from U satisfying α. The A-*lower* and A-*upper approximation* of $X \subseteq U$ with respect to A are denoted by $\underline{A}X$ and $\overline{A}X$, respectively. By $Inf_A^{IS}(u)$ (or by $Inf_A(u)$) we denote the *signature* of x in IS (or in A), i.e., the set $\{(a, a(x)) : a \in A\}$ and by $INF^{IS}(A)$ the set $\{Inf_A(u) : u \in U\}$. If $DT = (U, A, d)$ is a *decision table* then we assume the set V_d of values of the *decision* d to be equal to $\{1, \ldots, r(d)\}$ for some positive integer $r(d)$ called *the range of* d. The decision d determines a partition $\{C_1, \ldots, C_{r(d)}\}$ of the universe U, where $C_k = \{x \in U : d(x) = k\}$ for $1 \leq k \leq r(d)$. The set C_k is called the *k-th decision class* of DT.

2.2 Syntax and Semantics of Information Granules

Usually, together with an approximation space, there is also specified a set of formulas Φ expressing properties of objects. Hence, we assume that together with the approximation space $AS_{\#,\$}$ there are given

- a set of formulas Φ over some language;
- semantics Sem of formulas from Φ, i.e., a function from Φ into the power set $P(U)$.

Let us consider an example [27]. We define a language L_{IS} used for elementary granule description, where $IS = (U, A)$ is an information system. The syntax of L_{IS} is defined recursively by

$$(a \in V) \in L_{IS}, \text{ for any } a \in A \text{ and } V \subseteq V_a \tag{3}$$
$$\text{if } \alpha \in L_{IS} \text{ then } \neg\alpha \in L_{IS} \tag{4}$$
$$\text{if } \alpha, \beta \in L_{IS} \text{ then } \alpha \wedge \beta \in L_{IS} \tag{5}$$
$$\text{if } \alpha, \beta \in L_{IS} \text{ then } \alpha \vee \beta \in L_{IS}. \tag{6}$$

The semantics of formulas from L_{IS} with respect to an information system IS is defined recursively by

$$Sem_{IS}(a \in V) = \{x \in U : a(x) \in V\} \tag{7}$$
$$Sem_{IS}(\neg\alpha) = U - Sem_{IS}(\alpha) \tag{8}$$
$$Sem_{IS}(\alpha \wedge \beta) = Sem_{IS}(\alpha) \cap Sem_{IS}(\beta) \tag{9}$$
$$Sem_{IS}(\alpha \vee \beta) = Sem_{IS}(\alpha) \cup Sem_{IS}(\beta). \tag{10}$$

A typical method used by the rough set approach [27] for constructive definition of the uncertainty function is the following: for any object $x \in U$, there is given information $Inf_A(x)$ (information signature of x in A) which can be interpreted as a conjunction $EF_B(x)$ of selectors $a = a(x)$ for $a \in A$ and the set $I_\#(x)$ is equal to

$$Sem_{IS}(EF_B(x)) = Sem_{IS}\left(\bigwedge_{a \in A} a = a(x)\right). \tag{11}$$

One can consider a more general case taking as possible values of $I_\#(x)$ any set $\|\alpha\|_{IS}$ containing x. Next from the family of such sets the resulting neighborhood $I_\#(x)$ can be selected. One can also use another approach by considering more general approximation spaces in which $I_\#(x)$ is a family of subsets of U (see, e.g., [16]).

We present now syntax and semantics of examples of information granules. These granules are constructed by taking collections of already specified granules. They are parameterized by parameters which can be tuned in applications. In the following sections we discuss some other kinds of operations on granules as well as the inclusion and closeness relations for such granules.

Let us note that any information granule g formally can be defined by a pair

$$(Syn(g), Sem(g)) \tag{12}$$

consisting of the granules syntax $Syn(g)$ and semantics $Sem(g)$. However, for simplicity of notation we often use only one component of the information granules to denote it. One can consider another model in which these components are treated as separate granules and their fusion produces the above pair of information granules.

Elementary granules. In an information system $IS = (U, A)$, *elementary granules* are defined by $EF_B(x)$, where EF_B is a conjunction of selectors (descriptors) of the form $a = a(x)$, $B \subseteq A$ and $x \in U$. For example, the meaning of an elementary granule $a = 1 \wedge b = 1$ is defined by $Sem_{IS}(a = 1 \wedge b = 1) = \{x \in U : a(x) = 1 \;\&\; b(x) = 1\}$. The number of conjuncts in the granule can be taken as one of parameters to be tuned what is well known as the drooping condition technique in machine learning [18].

One can extend the set of elementary granules assuming that if α is any Boolean combination of descriptors over A, then $(\overline{B}\alpha)$ and $(\underline{B}\alpha)$ define syntax of elementary granules too, for any $B \subseteq A$. The reader can find more details on granules defined by rough set approximations in [46, 48].

Sequences of granules. Let us assume that S is a sequence of granules and the semantics $Sem_{IS}(\bullet)$ in IS of its elements have been defined. We extend $Sem_{IS}(\bullet)$ on S by

$$Sem_{IS}(S) = \{Sem_{IS}(g)\}_{g \in S}. \tag{13}$$

Example 1. *Granules defined by rules in information systems are examples of sequences of granules. Let IS be an information system and let (α, β) be a new information granule received from the rule **if** α **then** β where α, β are elementary granules of IS. The semantics $Sem_{IS}((\alpha, \beta))$ of (α, β) is the pair of sets $(Sem_{IS}(\alpha), Sem_{IS}(\beta))$. If the right hand sides of rules represent decision classes then among parameters to be tuned in classification is the number of conjuncts on the left hand sides of rules. Typical goal is to search for minimal (or less than minimal) number of such conjuncts (corresponding to the largest generalization) which still guarantee the satisfactory degree of inclusion in a decision class [14, 18, 42].*

Sets of granules. Let us assume that a set G of granules and the semantics $Sem_{IS}(\bullet)$ in IS for granules from G have been defined. We extend $Sem_{IS}(\bullet)$ on the family of sets $H \subseteq G$ by $Sem_{IS}(H) = \{Sem_{IS}(g) : g \in H\}$. One can consider as a parameter of any such granule its cardinality or its size (e.g., the length of such granule representation). In the first case, a typical problem is to search in a given family of granules for a granule of the smallest cardinality sufficiently close to a given one.

Example 2. *One can consider granules defined by sets of rules [27, 14]. Assume that there is a set of rules $Rule\_Set = \{(\alpha_i, \beta_i) : i = 1, \ldots, k\}$. The semantics of $Rule\_Set$ is defined* *by*

$$Sem_{IS}\left(Rule\_Set\right) = \left\{Sem_{IS}\left((\alpha_i, \beta_i)\right) : i = 1, \ldots, k\right\}. \tag{14}$$

The above mentioned searching problem for a set of granules corresponds in the case of rule sets to searching for the simplest representation of a given rule collection by another set of rules (or a single rule) sufficiently close to the collection [5, 52].

Example 3. *Let us consider a set G of elementary information granules – describing possible* situations *together – with decision table DT_α representing decision tables for any situation $\alpha \in G$. Assume $Rule\_Set\left(DT_\alpha\right)$ to be a set of decision rules generated from decision table DT_α (e.g., in the minimal form)[14]. Now let us consider a new granule*

$$\left\{(\alpha, Rule\_Set\left(DT_\alpha\right)) : \alpha \in G\right\} \tag{15}$$

with semantics defined by

$$\begin{aligned}\left\{Sem_{DT}\left((\alpha, Rule\_Set\left(DT_\alpha\right))\right) : \alpha \in G\right\} = \\ \left\{\left(Sem_{IS}\left(\alpha\right), Sem_{DT}\left(Rule\_Set\left(DT_\alpha\right)\right)\right) : \alpha \in G\right\}.\end{aligned} \tag{16}$$

An example of a parameter to be tuned is the number of situations represented in such granule. A typical task is to search for a granule with the minimal number of situations creating together with the corresponding to them rule sets a granule sufficiently close to the original one.

Extension of granules defined by tolerance relation. Now we present examples of granules obtained by application of a tolerance relation (i.e., reflexive and symmetric relation; for more information see, e.g., [43], and [11] for clustering methods based on similarity).

Example 4. *One can consider extension of elementary granules defined by a tolerance relation. Let $IS = (U, A)$ be an information system and let τ be a tolerance relation on elementary granules of IS. Any pair $(\tau : \alpha)$ is called a τ-*elementary granule*. The semantics $Sem_{IS}\left((\tau : \alpha)\right)$ of $(\tau : \alpha)$ is the family $\left\{Sem_{IS}\left(\beta\right) : (\beta, \alpha) \in \tau\right\}$. Parameters to be tuned in searching for relevant tolerance granule can be its support (represented by the number of supporting it objects) and a degree of its inclusion (or closeness) in some other granules as well as parameters specifying the tolerance relation.*

Example 5. *Let us consider granules defined by rules of* tolerance information systems *[43]. Let $IS = (U, A)$ be an information system and let τ be a tolerance relation on elementary granules of IS. If **if** α **then** β is a rule in IS then the semantics of a new information granule $(\tau : \alpha, \beta)$ is defined by $Sem_{IS}\left((\tau : \alpha, \beta)\right) = Sem_{IS}\left((\alpha, \tau)\right) \times Sem_{IS}\left((\beta, \tau)\right)$. Parameters to be tuned are the same as in the case of granules being sets of more elementary granules as well as parameters of the tolerance relation.*

Clustering of decision and association rules is an important problem in data mining. The reader is referred for measures of closeness of such rules to, e.g., [5, 52].

Example 6. *We consider granules defined by sets of decision rules corresponding to a given evidence in tolerance decision tables. Let $DT = (U, A, d)$ be a decision table and let τ be a tolerance on elementary granules of $IS = (U, A)$. Now, any granule $(\alpha, Rule\_Set\left(DT_\alpha\right))$ can be considered as a representative for the information granule cluster*

$$\left(\tau : (\alpha, Rule\_Set\left(DT_\alpha\right))\right) \tag{17}$$

with the semantics

$$Sem_{DT}\left((\tau : (\alpha, Rule\_Set\left(DT_{\alpha}\right)))\right) = \left\{Sem_{DT}\left((\beta, Rule\_Set\left(DT_{\beta}\right))\right) : (\beta, \alpha) \in \tau\right\}. \quad (18)$$

One can see that the considered case is a special case of information granules from Example 3 with G defined by tolerance relation.

Dynamic granules. An elementary granule α of the information system IS is non-empty if $\|\alpha\|_{IS} \neq \emptyset$. A non-empty elementary granule β of IS is an extension of α if $\beta = \alpha \wedge \gamma$, where γ is an elementary granule. Let us consider *dynamic granules* defined by some subsets of

$$\{(\beta, Rule\_Set\left(DT_{\beta}\right)) : \beta \text{ is an extension of } \alpha\}. \quad (19)$$

The semantics of these new granules is defined as in the case of sets of granules. Any set G of elementary granules and a granule α are specifying new granules

$$\{(\beta, Rule\_Set\left(DT_{\beta}\right)) : \beta \text{ is an extension of } \alpha \text{ and } \beta \in G\} \quad (20)$$

important for decision making in dynamically changing environment. Let us consider an example. A DT-path is any sequence $\pi = ((\alpha_1, R_1), \ldots, (\alpha_k, R_k))$ such that α_i is an elementary non-empty granule of IS, $R_i = Rule\_Set(DT_{\alpha_i})$ for $i = 1, \ldots, k$ and $\alpha_i = \alpha_{i-1} \wedge \gamma_{i-1}$ for some elementary atomic granule γ_{i-1} (e.g., selector $a = v$) with an attribute $a \in A$ not appearing in α_{i-1} for $i = 2, \ldots, k$. A granule α_{i-1} is called a *guard* of π if R_{i-1} is not sufficiently close to R_i (what we denote by $non(cl_p(R_{i-1}, R_i))$, where p is the closeness degree). By $Guard(\pi)$ we denote the subsequence of $\alpha_1, \ldots, \alpha_k$ consisting all guards of π. In applications it is important to search for a minimal (in cardinality) set of granules G satisfying the following condition: for any maximal DT-path π of extensions of α all guards β from $Guard(\pi)$ (i.e., all points in which it is sufficient to change the decision algorithm represented by the set of decision rules) are from G.

One can also consider dynamic granules with tolerance relation. Let $DT = (U, A, d)$ be a decision table and let τ be a tolerance relation on elementary granules of $IS = (U, A)$. Two DT-paths $\pi = ((\alpha_1, R_1), \ldots, (\alpha_k, R_k))$ and $\pi' = ((\beta_1, R'_1), \ldots, (\beta_l, R'_l))$ are τ-***similar*** if and only if $(\alpha_{i_s}, \beta_{j_s}) \in \tau$ for $s = 1, \ldots, r$, where $Guard(\pi) = (\alpha_{i_1}, \ldots, \alpha_{i_r})$ and $Guard(\pi') = (\beta_{j_1}, \ldots, \beta_{j_r})$. Let us assume τ has the following property:

if $(\beta, \alpha) \in \tau$ (21)
then the granules $Rule\_Set\left(DT_{\alpha}\right)$ and $Rule\_Set\left(DT_{\beta}\right)$ are sufficiently close.

Having such tolerance relation one can search for a set G of guards of the smaller size than before. To specify the task is enough to change in the above formulated problem the condition for the maximal path to the following one: for any maximal path π of extensions of α there exists a τ-similar path π' to π such that all guards β from $Guard(\pi')$ (i.e., all points where it is sufficient to change the decision algorithm represented by the set of decision rules) are from G.

Labeled graph granules. We discuss *graph granules* and *labeled graph* granules to extend previously introduced granules defined by tolerance relation and dynamic granules.

Example 7. *Let us consider granules defined by pairs* (G, E) *, where* G *is a finite set of granules with semantics in a given information system* $IS = (U, A)$ *and* $E \subseteq G \times G$*. The semantics of a new information granule* (G, E) *is defined by*

$$Sem_{IS}\left((G, E)\right) = \left(Sem_{IS}\left(G\right), Sem_{IS}\left(E\right)\right) \quad (22)$$

where

$$Sem(G)_{IS} = \{Sem(g)_{IS} : g \in G\} \textit{ and} \quad (23)$$
$$Sem(E)_{IS} = \{(Sem(g), Sem(g')) : (g, g') \in E\}.$$

Example 8. *Let* G *be a set of granules with semantics over a given information system* IS*. Labeled graph granules over* G *are defined by* (X, E, f, h) *, where* $E \subseteq X \times X$*,* $f : X \to G$ *and* $h : E \to P\left(G \times G\right)$*. We also assume one additional condition*

$$\textit{if } (x, y) \in E \textit{ then } (f(x), f(y)) \in h(x, y). \quad (24)$$

The semantics of the labeled graph granule (X, E, f, h) *is defined by*

$$\{(Sem(f(x))_{IS}, Sem(h(x, y))_{IS}, Sem(f(y))_{IS}) : (x, y) \in E\}. \quad (25)$$

Let us summarize the above presented considerations. One can define the set of granules G as the least set containing a given set of elementary granules G_0 and closed with respect to operations from a given set of operations on information granules.

We have the following examples of granule construction rules:

$$\frac{\alpha_1, \ldots, \alpha_k\text{-elementary granules}}{\{\alpha_1, \ldots, \alpha_k\}\text{-granule}} \quad (26)$$

$$\frac{\alpha_1, \alpha_2\text{-elementary granules}}{(\alpha_1, \alpha_2)\text{-granule}} \quad (27)$$

$$\frac{\alpha\text{-elementary granule }, \tau}{(\tau : \alpha)\text{-granule}} \quad (28)$$

$$\frac{G\text{-a finite set of granules }, E \subseteq G \times G}{(G, E)\text{-granule}} \quad (29)$$

where τ is a tolerance relation on elementary granules.

Let us observe that in case of granules constructed with application of tolerance relation we have the rule restricted to elementary granules. To obtain a more general rule like

$$\frac{\alpha\text{-graph granule, } \tau}{(\tau : \alpha)\text{-granule}} \quad (30)$$

where τ is a tolerance relation on elementary granules; it is necessary to extend the tolerance (similarity, closeness) relation on more complex objects.

One more interesting class of information granules create classifiers. This example will be discusses in one of the following sections. Parameters to be tuned are voting strategies, matching strategies of objects against rules as well as other discussed above parameters like closeness of granule in the target granule.

In presented examples we have discussed parameterized information granules. We have pointed out that the process of the parameters tuning is used to induce relevant (for a given task) information granules. In particular the process of parameter tuning is performed to obtain a satisfactory degree of inclusion (closeness) of information granules.

In the following section, we discuss inclusion and closeness for information granules.

2.3 Granule Inclusion and Closeness

In this section, we will discuss inclusion and closeness of different information granules introduced in the previous section. The inclusion and closeness are the basic concepts related to information granules [35, 45]. Using them one can measure the closeness of the constructed granule to the target granule and robustness of the construction scheme with respect to deviations of information granules being components of the construction. For details and examples of closeness relations the reader is referred to [35, 45].

Let us mention that the choice of inclusion or closeness definition depends very much on the area of application and data analyzed. This is the reason that we have decided to introduce a separate section with this more subjective (or task oriented) part of granule semantics.

The *inclusion relation* between granules G, G' to degree at least p (i.e., $\nu_p(G, G') \geq p$) will be denoted by $\nu_p(G, G')$. By $\underline{\nu}_p(G, G')$ we denote the inclusion of G in G' to degree at most p, i.e., that $\nu(G, G') \leq p$ holds. Similarly, the *closeness relation* between granules G, G' to degree at least p will be denoted by $cl_p(G, G')$. By p we denote a vector of parameters (e.g., from the interval [0,1] of real numbers). Usually, the set of degrees is assumed to be a lattice with null 0 and unit 1 elements.

A general scheme for construction of *hierarchical granules* and their closeness can be described by the following recursive meta-rule: if granules of order $\leq k$ and their closeness have been defined then the closeness $cl_p(G, G')$ (at least to degree p) between granules G, G' of order $k+1$ can be defined by applying an appropriate operator F to closeness values of components of G, G', respectively. Certainly, the same scheme can be applied to inclusion measures.

Elementary granules. We have introduced the simplest case of granules in information system $IS = (U, A)$. They are defined by $EF_B(x)$, where EF_B is a conjunction of selectors of the form $a = a(x)$, $B \subseteq A$ and $x \in U$. Let $G_{IS} = \{EF_B(x) : B \subseteq A \,\&\, x \in U\}$. In the standard rough set model [27] elementary granules describe indiscernibility classes with respect to some subsets of attributes. In a more general setting see, e.g., [43] tolerance (similarity) classes are described.

The crisp inclusion of α in β, where $\alpha, \beta \in \{EF_B(x) : B \subseteq A \,\&\, x \in U\}$ is defined by $Sem_{IS}(\alpha) \subseteq Sem_{IS}(\beta)$, where $Sem_{IS}(\alpha)$ and $Sem_{IS}(\beta)$ are sets of objects from IS satisfying α and β, respectively. The non-crisp inclusion, known in KDD [1], for the case of association rules is defined by means of two thresholds t and t':

$$support_{IS}(\alpha, \beta) = card(Sem_{IS}(\alpha \wedge \beta)) \geq t \tag{31}$$

$$confidence_{IS}(\alpha, \beta) = \frac{support_{IS}(\alpha, \beta)}{card(Sem_{IS}(\alpha))} \geq t'. \tag{32}$$

Elementary granule inclusion in a given information system IS can be defined using different schemes, e.g., by

$$\nu_t^{IS}(\alpha, \beta) \text{ if and only if } accuracy_{IS}(\alpha, \beta) \geq t. \tag{33}$$

The closeness of granules can be defined by

$$cl_{t,t'}^{IS}(\alpha, \beta) \text{ if and only if } \nu_{t,t'}^{IS}(\alpha, \beta) \text{ and } \nu_{t,t'}^{IS}(\beta, \alpha) \text{ hold.} \tag{34}$$

Decision rules as granules. One can define inclusion and closeness of granules corresponding to rules of the form **if** α **then** β using accuracy coefficients. Having such granules

$g = (\alpha, \beta)$, $g' = (\alpha', \beta')$ one can define inclusion and closeness of g and g' by

$$\nu_{t,t'}^{IS}(g, g') \text{ if and only if } \nu_{t,t'}^{IS}(\alpha, \alpha') \text{ and } \nu_{t,t'}^{IS}(\beta, \beta') . \quad (35)$$

The closeness can be defined by

$$cl_{t,t'}^{IS}(g, g') \text{ if and only if } \nu_{t,t'}^{IS}(g, g') \text{ and } \nu_{t,t'}^{IS}(g', g) . \quad (36)$$

Another way of defining inclusion of granules corresponding to decision rules is as follows

$$\nu_{t}^{IS}((\alpha, \beta), (\alpha', \beta')) \text{ if and only if } \nu_{t_1,t_2}^{IS}(\alpha, \alpha') \text{ and } \nu_{t_1,t_2}^{IS}(\beta, \beta') \text{ and } t = w_1 \cdot t_1 + w_2 \cdot t_2 \quad (37)$$

where w_1, w_2 are some given weights satisfying $w_1 + w_2 = 1$ and $w_1, w_2 \geq 0$.

Measures of closeness of rules are discussed in, e.g., [5, 52].

Extensions of elementary granules by tolerance relation. For extensions of elementary granules defined by similarity (tolerance) relation, i.e., granules of the form (α, τ), (β, τ) one can consider the following inclusion measure:

$$\nu_{t,t'}^{IS}((\alpha, \tau)(\beta, \tau)) \text{ if and only if } \nu_{t,t'}^{IS}(\alpha', \beta') \text{ for any } \alpha', \beta' \text{ such that } (\alpha, \alpha') \in \tau \text{ and } (\beta, \beta') \in \tau \quad (38)$$

and the following closeness measure:

$$cl_{t,t'}^{IS}((\alpha, \tau)(\beta, \tau)) \text{ if and only if } \nu_{t,t'}^{IS}((\alpha, \tau)(\beta, \tau)) \text{ and } \nu_{t,t'}^{IS}((\beta, \tau)(\alpha, \tau)) . \quad (39)$$

It can be important for some applications to define closeness of an elementary granule α and the granule (α, τ) . The definition reflecting an intuition that α should be a representation of (α, τ) sufficiently close to this granule is the following one:

$$cl_{t,t'}^{IS}(\alpha, (\alpha, \tau)) \text{ if and only if } cl_{t,t'}^{IS}(\alpha, \beta) \text{ for any } (\alpha, \beta) \in \tau. \quad (40)$$

Sets of rules. An important problem related to association rules is that the number of such rules generated even from simple data table can be large. Hence, one should search for methods of aggregating close association rules. We suggest that this can be defined as searching for some close information granules.

Let us consider two finite sets $Rule\_Set$ and $Rule\_Set'$ of association rules defined by

$$Rule\_Set = \{(\alpha_i, \beta_i) : i = 1, \ldots, k\} \quad (41)$$

$$Rule\_Set' = \{(\alpha_i', \beta_i') : i = 1, \ldots, k'\} . \quad (42)$$

One can treat them as higher order information granules. These new granules $Rule\_Set$, $Rule\_Set'$ can be treated as close to a degree at least t (in IS) if and only if there exists a relation rel between sets of rules $Rule\_Set$ and $Rule\_Set'$ such that:

1. For any $Rule \in Rule\_Set$ there is $Rule' \in Rule\_Set'$ such that $(Rule, Rule') \in rel$ and $Rule$ is close to $Rule'$ (in IS) to degree at least t.

2. For any $Rule' \in Rule\_Set'$ there is $Rule \in Rule\_Set$ such that $(Rule, Rule') \in rel$ and $Rule$ is close to $Rule'$ (in IS) to degree at least t.

Another way of defining closeness of two granules G_1, G_2 represented by sets of rules can be described as follows.

Let us consider again two granules $Rule\_Set$ and $Rule\_Set'$ corresponding to two decision algorithms. By $I(\beta'_i)$ we denote the set $\{j : cl_p^{IS}(\beta'_j, \beta'_i)\}$ for any $i = 1, \ldots, k'$.

Now, we assume $\nu_p^{IS}(Rule\_Set, Rule\_Set')$ if and only if for any $i \in \{1, \ldots, k'\}$ there exists a set $J \subseteq \{1, \ldots, k\}$ such that

$$cl_p^{IS}\left(\bigvee_{j \in I(\beta'_i)} \beta'_j, \bigvee_{j \in J} \beta_j\right) \text{ and } cl_p^{IS}\left(\bigvee_{j \in I(\beta'_i)} \alpha'_j, \bigvee_{j \in J} \alpha_j\right) \tag{43}$$

and for closeness we assume

$$cl_p^{IS}(Rule\_Set, Rule\_Set') \text{ if and only if } \nu_p^{IS}(Rule\_Set, Rule\_Set') \text{ and } \nu_p^{IS}(Rule\_Set', Rule\_Set). \tag{44}$$

For example, if the granule G_1 consists of rules: **if** α_1 **then** $d = 1$, **if** α_2 **then** $d = 1$, **if** α_3 **then** $d = 1$, **if** β_1 **then** $d = 0$, **if** β_2 **then** $d = 0$ and the granule G_2 consists of rules: **if** γ_1 **then** $d = 1$, **if** γ_2 **then** $d = 0$, then

$$cl_p(G_1, G_2) \text{ if and only if } cl_p(\alpha_1 \vee \alpha_2 \vee \alpha_3, \gamma_1) \text{ and } cl_p(\beta_1 \vee \beta_2, \gamma_2). \tag{45}$$

One can consider a searching problem for a granule $Rule\_Set'$ of minimal size such that $Rule\_Set$ and $Rule\_Set'$ are close. Certainly, the above discussed example is only a simple example of closeness measure between rule sets and for a given real-life application one should induce relevant closeness measures.

Granules defined by sets of granules. The previously discussed methods of inclusion and closeness definition can be easily adopted for the case of granules defined by sets of already defined granules. Let G, H be sets of granules.

The inclusion of G in H can be defined by

$$\nu_{t,t'}^{IS}(G, H) \text{ if and only if for any } g \in G \text{ there is } h \in H \text{ for which } \nu_{t,t'}^{IS}(g, h) \tag{46}$$

and the closeness by

$$cl_{t,t'}^{IS}(G, H) \text{ if and only if } \nu_{t,t'}^{IS}(G, H) \text{ and } \nu_{t,t'}^{IS}(H, G). \tag{47}$$

Let G be a set of granules and let φ be a property of sets of granules from G (e.g., $\varphi(X)$ if and only if X is a tolerance class of a given tolerance $\tau \subseteq G \times G$). Then $P_\varphi(G) = \{X \subseteq G : \varphi(X) \text{ holds}\}$. Closeness of granules $X, Y \in P_\varphi(G)$ can be defined by

$$cl_t(X, Y) \text{ if and only if } cl_t(g, g') \text{ for any } g \in X \text{ and } g' \in Y. \tag{48}$$

We have the following examples of inclusion and closeness propagation rules:

$$\frac{\text{for any } \alpha \in G \text{ there is } \alpha' \in H \text{ such that } \nu_p(\alpha, \alpha')}{\nu_p(G, H)} \tag{49}$$

$$\frac{cl_p(\alpha,\alpha'), cl_p(\beta,\beta')}{cl_p((\alpha,\beta),(\alpha',\beta'))} \tag{50}$$

$$\frac{\text{for any } \alpha' \in \tau(\alpha) \text{ there is } \beta' \in \tau(\beta) \text{ s. t. } \nu_p(\alpha',\beta')}{\nu_p((\tau:\alpha),(\tau:\beta))} \tag{51}$$

$$\frac{cl_p(G,G'), cl_p(E,E')}{cl_p((G,E),(G',E'))} \tag{52}$$

where $\alpha, \alpha', \beta, \beta'$ are elementary granules, G, G' are finite sets of elementary granules.

The exemplary rules have a general form, i.e., they are true in any IS (under the chosen definition of inclusion and closeness). Some of them are derivable from others. We will see in the next part of our chapter that there are also some operations of new granules construction specific for a given information system. In this case one should extract inference rules from given data.

Information granules defined by inclusion and closeness measures. Let us observe that inclusion (closeness) measures can be used to define new granules being approximations or generalizations of existing ones. Assume g, h are given information granules, and ν_p is inclusion measure (where $p \in [0,1]$). A (h,p)- approximation of g is an information granule $\nu_{h,p}$ represented by a set $\{h' : \nu_1(h',h) \wedge \nu_p(h',g)\}$. Now the lower and upper approximations of given information granules can be easily defined [43].

2.4 Rough–Fuzzy Granules

In this section, we will discuss briefly approximation schemes of granules and methods for extracting from them relevant patterns in case when they include fuzzy concepts as foreign parts. We propose to use rough set approach to define in a constructive way approximations of fuzzy concepts [46, 47]. The rough set approximations of the fuzzy cuts are used in searching for constructive definition of approximations of fuzzy sets. We use the cut approximations to derive patterns relevant for the target concept approximation. In the process of searching for high quality patterns evolutionary techniques can be used.

Let $DT = (U, A, d)$ be a decision table with the decision being the restriction to the objects from U of the fuzzy membership function $\mu : U \to [0,1]$. Consider reals $0 < c_1 < \ldots < c_k$ where $c_i \in (0,1]$ for $i = 1, \ldots, k$. Any c_i defines c_i-cut by $X_i = \{x \in U : \mu(x) \geq c_i\}$. Assume $X_0 = U, X_{k+1} = X_{k+2} = \emptyset$.

A *rough–fuzzy granule* (*rf–granule*, for short) corresponding to $(DT, c_1, \ldots, c_k)$ is any granule $g = (g_0, \ldots, g_k)$ such that for some $B \subseteq A$

$$Sem_B(g_i) = (\underline{B}(X_i - X_{i+1}), \overline{B}(X_i - X_{i+1})) \text{ for } i = 0, \ldots, k \tag{53}$$

$$\overline{B}(X_i - X_{i+1}) \subseteq (X_{i-1} - X_{i+2}) \text{ for } i = 1, \ldots, k. \tag{54}$$

Any function $\mu^* : U \to [0,1]$ satisfying the following conditions:

$$\mu^*(x) = 0 \text{ if } x \in U - \overline{B}X_1 \tag{55}$$

$$\mu^*(x) = 1 \text{ if } x \in \underline{B}X_k \tag{56}$$

$$\mu^*(x) = c_{i-1} \text{ if } x \in \underline{B}(X_{i-1} - X_i) \text{ for } i = 2, \ldots, k-1 \tag{57}$$

$$c_{i-1} < \mu^*(x) < c_i \text{ if } x \in (\overline{B}X_i - \underline{B}X_i) \text{ for } i = 1, \ldots, k \text{ and } c_0 = 0 \tag{58}$$

is called a B-*approximation* of μ.

Now one can choose of the lower or upper approximations of parts, i.e., the set theoretical differences between successive cuts, and propagate them along the scheme in searching for relevant patterns. Another strategy is to propagate the global approximation of foreign fuzzy concepts through the scheme describing target concept.

This problem is of a great importance in classification of situations by autonomous systems on the basis of sensor measurements [56].

2.5 Classifiers as Information Granules

An important class of information granules create classifiers. One can observe that sets of decision rules generated from a given decision table $DT = (U, A, d)$ (see, e.g., [40] can be interpreted as information granules. The *classifier* construction from DT can be described as follows:

1. First, one can construct granules G_j corresponding to each particular decision $j = 1, \ldots, r$ by taking a collection $\{g_{ij} : i = 1, \ldots, k_j\}$ of left hand sides of decision rules for a given decision.

2. Let E be a set of elementary granules (e.g., defined by conjunction of descriptors) over $IS = (U, A)$. We can now consider a granule denoted by

$$Match(e, G_1, \ldots, G_r) \quad (59)$$

   for any $e \in E$ being a collection of coefficients ε_{ij} where $\varepsilon_{ij} = 1$ if the set of objects defined by e in IS is included in the meaning of g_{ij} in IS, i.e., $Sem_{IS}(e) \subseteq Sem_{IS}(g_{ij})$; and 0, otherwise. Hence, the coefficient ε_{ij} is equal to 1 if and only if the granule e matches in IS the granule g_{ij}.

3. Let us now denote by $Conflict\_res$ an operation (resolving conflict between decision rules recognizing elementary granules) defined on granules of the form

$$Match(e, G_1, \ldots, G_r)$$

   with values in the set of possible decisions $1, \ldots, r$. Hence,

$$Conflict\_res(Match(e, G_1, \ldots, G_r)) \quad (60)$$

   is equal to the decision predicted by the classifier

$$Conflict\_res(Match(\bullet, G_1, \ldots, G_r)) \quad (61)$$

   on the input granule e.

Hence, classifiers are special cases of information granules. Parameters to be tuned are voting strategies, matching strategies of objects against rules as well as other parameters like closeness of granules in the target granule.

The classifier construction is illustrated in Figure 4 where three sets of decision rules are presented for the decision values $1, 2, 3$, respectively. Hence, we have $r = 3$. In figure to omit too many indices we write α_i instead of g_{i1}, β_i instead of g_{i2}, and γ_i instead of g_{i3},

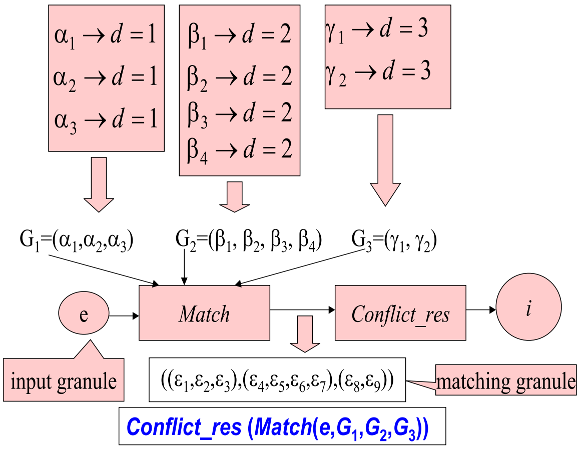

Figure 4: Classifiers as information granules

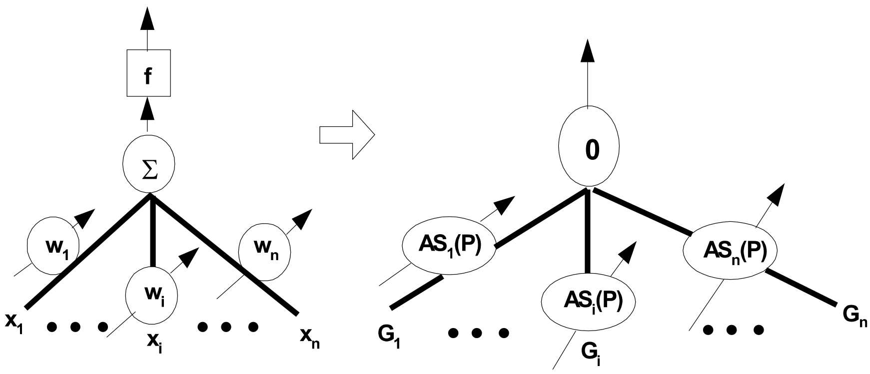

Figure 5: Rough neuron

respectively. Moreover, $\varepsilon_1, \varepsilon_2, \varepsilon_3$, denote $\varepsilon_{1,1}, \varepsilon_{2,1}, \varepsilon_{3,1}$; $\varepsilon_4, \varepsilon_5, \varepsilon_6, \varepsilon_7$ denote $\varepsilon_{1,2}, \varepsilon_{2,2}, \varepsilon_{3,2}, \varepsilon_{4,2}$; and $\varepsilon_8, \varepsilon_9$ denote $\varepsilon_{1,3}, \varepsilon_{2,3}$, respectively.

The reader can now easily describe more complex classifiers by means of information granules. For example, one can consider soft instead of crisp inclusion between elementary information granules representing classified objects and the left hand sides of decision rules or soft matching between recognized objects and left hand sides of decision rules.

Observe that any classifier realizes a kind of $Make\_granule$ operation transforming collections of granules into granules representing decisions.

3 Rough Neurocomputing: Weights Defined by Approximation Spaces

In this section, we will discuss the rough neurocomputing paradigm using model for information granule construction introduced in [44, 45]. First we elaborate a general scheme for information granule construction in distributed systems. Such schemes are parameterized, in particular by local parameterized approximation spaces. These parameterized approximation spaces can be treated as analogy to the neural network weights. The parameters should be learned to induce the relevant information granules (see Figure 5).

We use terminology from multi-agent area to explain the basic constructions [12].

Teams of agents are organized, e.g., along the schemes of decomposition of complex objects (e.g., representing situations on the road) into trees. The trees are represented by expressions called *terms*. Two granules are defined for any term t under a given valuation val of leaf agents of t in the set of input granules. They are called the lower and upper approximations of t under val.

The necessity to consider approximation of granule returned by a given term t under a given valuation val rather than the exact value of t under val is a consequence of the ability of agents to perceive in approximate sense only information granules received from other agents. Hence, approximate reasoning in distributed environment requires a construction of interfaces between agents (information sources or units) enabling effective learning by agents of concepts definable by other agents. In the chapter, we suggest a solution based on exchanging views of agents on objects with respect to a given concept. An agent delivering a concept is submitting positive and negative examples (objects) with respect to a given concept. The agent receiving this information describes objects using its own attributes. In this way a data table (called a *decision table*) is created and the approximate description of concept can be extracted by the receiving agent. Our solution is based on the rough set approach. We propose to use the parameterized approximation spaces to allow appropriate tuning of concept perception by agents (see Figures 6 and 7).

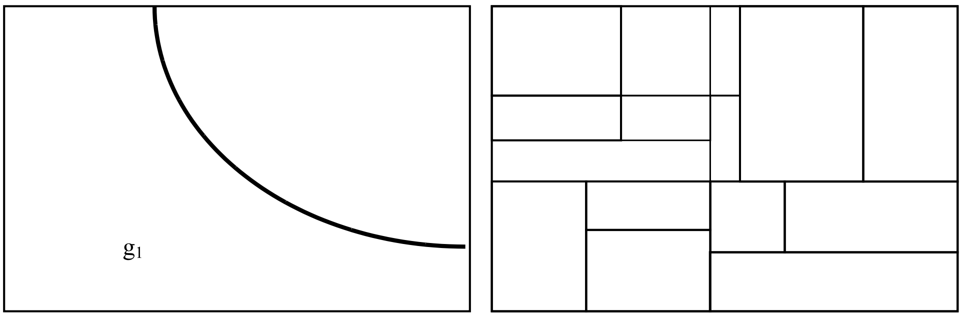

Figure 6: Concept g_1 – information granule of $ag_1 \in Ag$ (left) and communication interface defined by data table (right)

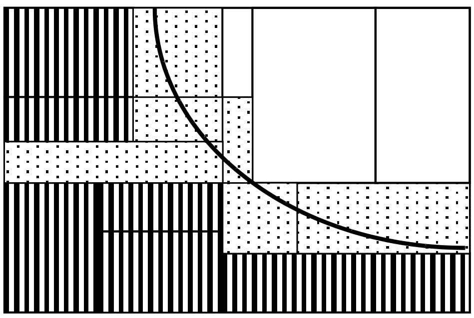

Figure 7: Lower and upper approximation of g_1 by $ag \in Ag$

One can consider different problems related to synthesis of AR-schemes defined by terms. For example, one can look for a strategy returning for a given specification granule a term t and its valuation val such that the granules defined by the lower and upper values of t under val are sufficiently included in the soft specification granule. Moreover, one can require these granules to be of high quality (e.g., supported by many objects) and the term t to be *robust* with respect to the deviations of val, i.e., the lower and upper values of t under val'

sufficiently close to val to be included in the soft specification granule to a satisfactory degree. Observe that such terms define pattern granules sufficiently included in the specification. Moreover, a given object (situation) is covered by this pattern if the valuation defined by this object is sufficiently close to val (see Figure 2).

We assume any non leaf-agent ag is equipped with an operation

$$o\left(ag\right):U_{ag}^{(1)}\times\ldots\times U_{ag}^{(k)}\rightarrow U_{ag}^{(0)} \tag{62}$$

and has different approximation spaces

$$AS_{ag}^{(i)}=\left(U_{ag}^{(i)},I_{ag}^{(i)},\nu_{SRI}\right) \text{ where } i=0,\ldots,k. \tag{63}$$

We assume that the agent ag is perceiving objects by measuring values of some available attributes. Hence, some objects can become indiscernible [27]. This influences the specification of any operation $o\left(ag\right).$ We consider a case when arguments and values of operations are represented by attribute value vectors. Hence, instead of the operation $o\left(ag\right)$ we have its inexact specification $o^{*}\left(ag\right)$ taking as arguments $I_{ag}^{(1)}\left(x_1\right),\ldots,I_{ag}^{(k)}\left(x_k\right)$ for some $x_1\in U_{ag}^{(1)},\ldots,x_k\in U_{ag}^{(k)}$ and returning the value $I_{ag}^{(0)}\left(o(ag)(x_1,\ldots,x_k)\right)$ if $o(ag)(x_1,\ldots,x_k)$ is defined, otherwise the empty set. This operation can be extended to the operation $o^{*}(ag)$ with arguments being definable sets (in approximation spaces attached to arguments) and with values in the family of all non-empty subsets of $U_{ag}^{(0)}$. Let $X_1,\ldots,X_k$ be definable sets. We define

$$o^{*}\left(ag\right)\left(X_1,\ldots,X_k\right)=\bigcup_{x_1\in X_1,\ldots,x_k\in X_k}o^{*}\left(ag\right)\left(I_{ag}^{(1)}\left(x_1\right),\ldots,I_{ag}^{(k)}\left(x_k\right)\right). \tag{64}$$

In the sequel, for simplicity of notation, we write $o\left(ag\right)$ instead of $o^{*}\left(ag\right).$

This idea can be formalized as follows. First we define terms representing agent schemes.

Let $X_{ag},Y_{ag},\ldots$ be agent variables for any leaf-agent $ag\in Ag$. Let $o\left(ag\right)$ denote a function of arity k. We have mentioned that it is an operation from Cartesian product of

$$Def\_Sets(AS_{ag}^{(1)}),\ldots,Def\_Sets(AS_{ag}^{(k)}) \tag{65}$$

into $P\left(U_{ag}^{(0)}\right)$ where $Def\_Sets(AS_{ag}^{(i)})$ denotes the family of sets definable in $AS_{ag}^{(i)}$. Using the above variables and functors we define terms in a standard way, for example

$$t=o\left(ag\right)\left(X_{ag_1},X_{ag_2}\right). \tag{66}$$

Such terms can be treated as description of complex information granules. By a valuation we mean any function val defined on the agent variables with values being definable sets satisfying $val(X_{ag})\subseteq U_{ag}$ for any leaf-agent $ag\in Ag$. Now we can define the lower value $val\left(LOW,AS_{ag}^{(i)}\right)(t)$ and the upper value $val\left(UPP,AS_{ag}^{(i)}\right)(t)$ of any term t under the valuation val with respect to a given approximation space $AS_{ag}^{(i)}$ of an agent ag.

1. If t is of the form X_{ag_i} and $val(t)\subseteq U_{ag}^{(i)}$ then

$$val\left(LOW,AS_{ag}^{(i)}\right)(t)=LOW\left(AS_{ag}^{(i)},val(t)\right) \tag{67}$$

$$val\left(UPP,AS_{ag}^{(i)}\right)(t)=UPP\left(AS_{ag}^{(i)},val(t)\right) \tag{68}$$

else the lower and the upper values are undefined;
where

$$LOW\left(AS_{ag}^{(i)}, val(t)\right) \text{ and } UPP\left(AS_{ag}^{(i)}, val(t)\right) \tag{69}$$

denotes the lower approximation and the upper approximation in $AS_{ag}^{(i)}$ of the set $val(t)$, respectively.

2. If $t = o(ag)(t_1, \ldots, t_k)$, where $t_1, \ldots, t_k$ are terms and $o\,(ag)$ is an operation of arity k, then
   (a) if $val\left(LOW, AS_{ag}^{(i)}\right)(t_i)$ is defined for $i = 1, \ldots, k$
   then

$$\begin{aligned} &val\left(LOW, AS_{ag}^{(0)}\right)(t) = \\ &LOW\left(AS_{ag}^{(0)}, o\,(ag)\left(val\left(LOW, AS_{ag}^{(1)}\right)(t_1),\ \ldots,\ val\left(LOW, AS_{ag}^{(k)}\right)(t_k)\right)\right) \end{aligned} \tag{70}$$

   else $val\left(LOW, AS_{ag}^{(0)}\right)(t)$ is undefined,
   (b) if $val\left(UPP, AS_{ag}^{(i)}\right)(t_i)$ is defined for $i = 1, \ldots, k$
   then

$$\begin{aligned} &val\left(UPP, AS_{ag}^{(0)}\right)(t) = \\ &UPP\left(AS_{ag}^{(0)}, o\,(ag)\left(val\left(UPP, AS_{ag}^{(1)}\right)(t_1), \ldots,\ val\left(UPP, AS_{ag}^{(k)}\right)(t_k)\right)\right) \end{aligned} \tag{71}$$

   else $val\left(UPP, AS_{ag}^{(0)}\right)(t)$ is undefined.

For illustrative example of computation of the lower and upper approximations of terms the reader is referred to [45].

Let us observe that the set

$$val(UPP, AS_{ag}^{(0)})(t) - val(LOW, AS_{ag}^{(0)})(t) \tag{72}$$

can be treated as the boundary region of t under val. Moreover, in the process of term construction we have additional parameters to be tuned for obtaining sufficiently high support and accuracy, namely the approximation operations.

A concept X specified by the customer-agent is *sufficiently close to* t *under a given set* Val *of valuations* if X is included in the upper approximation of t under any $val \in Val$ and X includes the lower approximation of t under any $val \in Val$ as well as the size of the boundary region of t under Val, i.e.,

$$card\left(\bigcap_{val \in Val} val\left(UPP, AS_{ag}^{(0)}\right)(t) - \bigcup_{val \in Val} val\left(LOW, AS_{ag}^{(0)}\right)(t)\right) \tag{73}$$

is sufficiently small relatively to

$$\bigcap_{val \in Val} val\left(UPP, AS_{ag}^{(0)}\right)(t). \tag{74}$$

4 Rough Neurocomputing: Rough Mereological Approach

We now present a conceptual scheme for an adaptive calculus of granules aimed at synthesizing solutions to problems posed under uncertainty. This exposition is based on our earlier analyzes presented in [32, 35, 37]. For recent developments the reader is referred to [24, 25, 48]. We construct a scheme of agents which communicate by relating their respective granules of knowledge by means of transfer functions induced by rough mereological connectives extracted from their respective information systems. Such schemes can be treated as AR-schemes. We assume the notation of [35] where the reader will find all the necessary information.

In the previous section ideas concerning rough neurocomputing based on approximation spaces have been discussed. Now we will present an approach to calculi of granules based on rough mereological approach. We will concentrate on some ideas. The formal details of rough mereology can be found, e.g., in [31, 36, 38].

We now define formally the ingredients of our scheme of agents.

4.1 Distributed Systems of Agents

We assume that a pair (Inv, Ag) is given where Inv is an *inventory of elementary objects* and Ag is a set of intelligent computing units called shortly *agents*.

We consider an agent *ag* $\in Ag$. The agent *ag* is endowed with tools for reasoning about objects in its scope; these tools are defined by components of the agent label. The *label of the agent ag* is the tuple

$$lab(ag) = (\mathcal{A}(ag), M(ag), L(ag), Link(ag), AP\_O(ag), St(ag), \\ Unc\_rel(ag), H(ag), Unc\_rule(ag), Dec\_rule(ag)) \quad (75)$$

where

1. $\mathcal{A}(ag) = (U(ag), A(ag))$ is an information system of the agent ag; we assume as an example that objects (i.e., elements of $U(ag)$) are granules of the form: $(\alpha, [\alpha])$ where α is a conjunction of descriptors (one may have more complex granules as objects).

2. $M(ag) = (U(ag), [0,1], \mu_o(ag))$ is a *pre - model* of L_{rm} with a *pre - rough inclusion* $\mu_o(ag)$ on the universe $U(ag)$ [31].

3. $L(ag)$ is a set of unary predicates (properties of objects) in a predicate calculus interpreted in the set $U(ag)$; we may assume that formulae of $L(ag)$ are constructed as conditional formulae of logics L_B where $B \subseteq A(ag)$.

4. $St(ag) = \{st(ag)_1, \ldots, st(ag)_n\} \subset U(ag)$ is the set of *standard objects* at ag.

5. $Link(ag)$ is a collection of strings of the form $t = ag_1 ag_2 \ldots ag_k ag$; the intended meaning of a string $ag_1 ag_2 \ldots ag_k ag$ is that $ag_1, ag_2, .., ag_k$ are children of ag in the sense that ag can assemble complex objects (constructs) from simpler objects sent by $ag_1, ag_2, \ldots, ag_k$. In general, we may assume that for some agents ag we may have more than one element in $Link(ag)$ which represents the possibility of re - negotiating the synthesis scheme.

We denote by the symbol $Link$ the union of the family $\{Link(ag) : ag \in Ag\}$.

6. $AP\_O(ag)$ consists of pairs of the form:

$$(o(ag,t), ((AS_1(o(ag), in), \cdots, AS_n(o(ag), in)), AS(o(ag), out)) \quad (76)$$

where $o(ag,t) \in O(ag)$, n is the arity of $o(ag,t)$, $t = ag_1 ag_2 \ldots ag_k ag \in Link$, $AS_i(o(ag,t), in)$ is a parameterized approximation space [44] corresponding to the $i-th$ argument of $o(ag,t)$ and $AS(o(ag,t), out)$ is a parameterized approximation space [44] for the output of $o(ag,t)$.

$O(ag)$ is the set of *operations at* ag; any $o(ag,t) \in O(ag)$ is a mapping of the Cartesian product $U(ag) \times U(ag) \times \ldots \times U(ag)$ into the universe $U(ag)$; the meaning of $o(ag,t)$ is that of an operation by means of which the agent ag is able to assemble from objects $x_1 \in U(ag_1), x_2 \in U(ag_2), \ldots, x_k \in U(ag_k)$ the object $z \in U(ag)$ which is an approximation defined by $AS(o(ag,t), out)$ to $o(ag,t)(y_1, y_2, \ldots, y_k) \in U(ag)$ where y_i is the approximation to x_i defined by $AS_i(o(ag,t), in)$. One may choose here either a lower or an upper approximation.

7. $Unc\_rel(ag)$ is the set of *uncertainty relations* unc_rel_i of type

$$(o_i(ag,t), \rho_i(ag), ag_1, \ldots, ag_k, ag, \mu_o(ag_1), \mu_o(ag_k), \mu_o(ag), st(ag_1)_i, \ldots, st(ag_k)_i, st(ag)_i) \tag{77}$$

where $t = ag_1 ag_2 \ldots ag_k ag \in Link(ag)$, $o_i(ag,t) \in O(ag)$ and ρ_i is such that

$$\rho_i((x_1,\varepsilon_1),(x_2,\varepsilon_2),.,(x_k,\varepsilon_k),(x,\varepsilon)) \tag{78}$$

holds for $x_1 \in U(ag_1), x_2 \in U(ag_2), .., x_k \in U(ag_k)$ and $\varepsilon, \varepsilon_1, \varepsilon_2, .., \varepsilon_k \in [0,1]$ if and only if $\mu_o(x, st(ag)_i) \geq \varepsilon_j$, and $\mu_o(x_j, st(ag_j)_i) \geq \varepsilon_j$ for $j = 1, 2, .., k$ for the collection of standards $st(ag_1)_i, st(ag_2)_i, .. ., st(ag_k)_i, st(ag)_i$ such that

$$o_i(ag,t)(st(ag_1)_i, st(ag_2)_i, .., st(ag_k)_i) = st(ag)_i. \tag{79}$$

The operation o performed by ag here is more complex then that of [35] as it is composed of three stages: first, approximations to input objects are constructed, next the operation is performed, and finally the approximation to the result is constructed. Relations unc_rel_i provide a global description of this process; in reality, they are composition of analogous relations corresponding to the three stages. As a result, unc_rel_i depend on parameters of approximation spaces. This concerns also other constructs discussed here. It follows that in order to get satisfactory decomposition (similarly, uncertainty and so on) rules one has to search for satisfactory parameters of approximation spaces (this is in analogy to weight tuning in neural computations).

Uncertainty relations express the agents knowledge about relationships among uncertainty coefficients of the agent ag and uncertainty coefficients of its children. The relational character of these dependencies expresses their intensionality.

8. $Unc\_rule(ag)$ is the set of *uncertainty rules* unc_rule_j of type

$$(o_j(ag,t), f_j, ag_1, ag_2, \ldots, ag_k, ag, st(ag_1), st(ag_2), \ldots, st(ag_k), st(ag), \mu_o(ag_1), \ldots, \mu_o(ag_k), \mu_o(ag)) \tag{80}$$

of the agent ag where $t = ag_1 ag_2 \ldots ag_k ag \in Link(ag)$ and $f_j : [0,1]^k \longrightarrow [0,1]$ is a function which has the property that for any $x_1 \in U(ag_1), x_2 \in U(ag_2), \ldots, x_k \in U(ag_k)$

$$\textbf{if } o_j(ag,t)(st(ag_1), st(ag_2), \ldots, st(ag_k)) = st(ag) \text{ and } \mu_o(x_i, st(ag_i)) \geq \varepsilon(ag_i) \text{ for } i = 1, 2, \ldots, k \;\textbf{then } \mu_o(o_j(ag,t)(x_1, x_2, \ldots, x_k), st(ag)) \geq f_j(\varepsilon(ag_1), \varepsilon(ag_2), \ldots, \varepsilon(ag_k)). \tag{81}$$

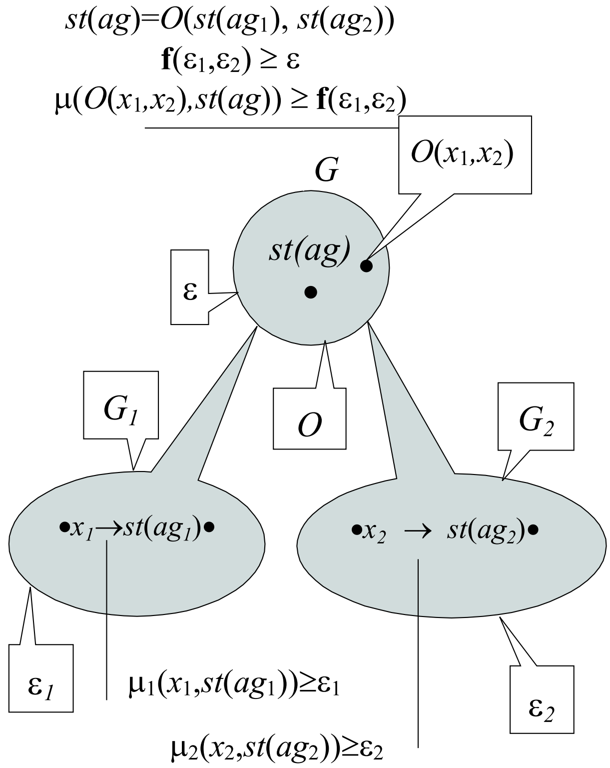

Figure 8: Uncertainty rules

In Figure 8 the idea of uncertainty rules is illustrated. Uncertainty rules provide functional operators, called *rough mereological connectives*, for propagating uncertainty measure values from the children of an agent to the agent; their application is in negotiation processes where they inform agents about plausible uncertainty bounds.

9. $H(ag)$ is a strategy which produces uncertainty rules from uncertainty relations; to this end, various rigorous formulas as well as various heuristics can be applied among them the algorithm presented in Section 2.8 of [35].

10. $Dec\_rule(ag)$ is a set of *decomposition rules* dec_rule_i of type

$$(o_i(ag,t), ag_1, ag_2, \ldots, ag_k, ag) \tag{82}$$

such that

$$(\Phi(ag_1), \Phi(ag_2), .., \Phi(ag_k), \Phi(ag)) \in dec\_rule_i \tag{83}$$

where

$$\Phi(ag_1) \in L(ag_1), \Phi(ag_2) \in L(ag_2), \ldots, \Phi(ag_k) \in L(ag_k), \Phi(ag) \in L(ag) \tag{84}$$

$$t = ag_1 ag_2 \ldots ag_k ag \in Link(ag) \tag{85}$$

and there exists a collection of standards $st(ag_1), st(ag_2), \ldots, st(ag_k), st(ag)$ with the properties that $o_j(ag,t)(st(ag_1), st(ag_2), .., st(ag_k)) = st(ag)$, $st(ag_i)$ satisfies $\Phi(ag_i)$ for $i = 1, 2, .., k$ and $st(ag)$ satisfies $\Phi(ag)$.

Decomposition rules are decomposition schemes in the sense that they describe the standard $st(ag)$ and the standards $st(ag_1), \ldots, st(ag_k)$ from which the standard $st(ag)$ is assembled under o_i in terms of predicates which these standards satisfy.

We may sum up the content of (1) - (10) above by saying that for any agent ag the possible sets of children of this agent are specified and, relative to each team of children, decompositions of standard objects at ag into sets of standard objects at the children, uncertainty relations as well as uncertainty rules, which relate similarity degrees of objects at the children to their respective standards and similarity degree of the object built by ag to the corresponding standard object at ag, are given.

We take rough inclusions of agents as measures of uncertainty in their respective universes. We would like to observe that the mereological relation of being a part is not transitive globally over the whole synthesis scheme because distinct agents use distinct mereological languages.

4.2 Approximate Synthesis of Complex Objects

The process of synthesis of a complex object (signal, action) by the above defined scheme of agents consists in our approach of the two communication stages viz. the top - down communication/negotiation process and the bottom - up communication/assembling process. We outline the AR-scheme construction in the language of approximate formulae.

For simplicity of exposition and to avoid unnecessarily tedious notation, we assume that the relation $ag' \leq ag$, which holds for agents $ag', ag \in Ag$ iff there exists a string $ag_1 ag_2 \ldots ag_k ag \in Link(ag)$ with $ag' = ag_i$ for some $i \leq k$, orders the set Ag into a tree. We also assume that $O(ag) = \{o(ag, t)\}$ for $ag \in Ag$, i.e., each agent has a unique assembling operation for a unique t.

To this end we build a logic $L(Ag)$ [35] in which we can express global properties of the synthesis process. We recall our assumption that the set Ag is ordered into a tree by the relation $ag' \leq ag$.

Elementary formulae of $L(Ag)$ are of the form $\langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$ where $st(ag) \in St(ag), \Phi(ag) \in L(ag), \varepsilon(ag) \in [0,1]$ 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 [ ], <> .

To introduce a semantics for the logic $L(ag)$, we first specify the meaning of satisfaction for elementary formulae. The meaning of a formula $\Phi(ag)$ is defined classically as the set $[\Phi(ag)] = \{u \in U(ag) : u$ has the property $\Phi(ag)\}$; we will denote the fact that $u \in [\Phi(ag)]$ by the symbol $u \models \Phi(ag)$. We extend now the satisfiability predicate $\models$ to approximate formulae: for $x \in U(ag)$, we say that x satisfies an elementary formula $\langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$, in symbols: $x \models < st(ag), \Phi(ag), \varepsilon(ag) >$, iff

$$st(ag) \models \Phi(ag) \text{ and } \mu_o(ag)(x, st(ag)) \geq \varepsilon(ag). \tag{86}$$

We let

$$x \models \neg\langle st(ag), \Phi(ag), \varepsilon(ag)\rangle \text{ iff it is not true that } x \models \langle st(ag), \Phi(ag), \varepsilon(ag)\rangle \tag{87}$$

$$x \models \langle st(ag)_1, \Phi(ag)_1, \varepsilon(ag)_1\rangle \vee \langle st(ag)_2, \Phi(ag)_2, \varepsilon(ag)_2\rangle \text{ iff} \tag{88}$$
$$x \models \langle st(ag)_1, \Phi(ag)_1, \varepsilon(ag)_1\rangle \text{ or } x \models \langle st(ag)_2, \Phi(ag)_2, \varepsilon(ag)_2\rangle.$$

In order to extend the semantics over modalities, we first introduce the notion of a selection: 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' (i.e., $sel, sel' : Ag \rightarrow \bigcup U_{ag}$ and $sel(ag), sel'(ag) \in U_{ag}$ for any $ag \in Ag$) we say that sel *induces* sel', in symbols $sel \rightarrow_{Ag} sel'$ when

$$sel(ag) = sel'(ag) \text{ for any } ag \in Leaf(Ag) \text{ and} \tag{89}$$
$$sel'(ag) = o(ag,t)(sel'(ag_1), sel'(ag_2), \ldots, sel'(ag_k)) \text{ for any } t = ag_1 ag_2 \ldots ag_k ag \in Link.$$

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

$$sel \models \langle st(ag), \Phi(ag), \varepsilon(ag)\rangle \text{ iff } sel(ag) \models \langle st(ag), \Phi(ag), \varepsilon(ag)\rangle. \tag{90}$$

We now let $sel \models <>< st(ag), \Phi(ag), \varepsilon(ag) >$ when there exists a selection sel' satisfying the conditions:

$$sel \rightarrow_{Ag} sel' \tag{91}$$
$$sel' \models \langle st(ag), \Phi(ag), \varepsilon(ag)\rangle. \tag{92}$$

In terms of logic $L(Ag)$ it is possible to express the problem of synthesis of an approximate solution to the problem posed to the team Ag. We denote by $head(Ag)$ the root of the tree $(Ag, \leq)$.

In the process of top - down communication, a requirement Ψ received by the scheme from an external source (which may be called a *customer*) is decomposed into approximate specifications of the form $\langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$ for any agent ag of the scheme. The decomposition process is initiated at the agent $head(Ag)$ and propagated down the tree.

We are able now to formulate the synthesis problem.

Synthesis problem

Given a formula

$$\alpha : \langle st(head(Ag)), \Phi(head(Ag)), \varepsilon(head(Ag))\rangle \tag{93}$$

find a selection sel *over the tree* $(Ag, \leq)$ *with the property* $sel \models <> \alpha$.

A solution to the synthesis problem with a given formula

$$\langle st(head(Ag)), \Phi(head(Ag)), \varepsilon(head(Ag))\rangle \tag{94}$$

is found by negotiations among the agents. Negotiations are based on uncertainty rules of agents and their successful result can be expressed by a top-down recursion in tree $(Ag, \leq)$ as follows: given a local team $ag_1 ag_2 \ldots ag_k ag$ with the formula $\langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$ already chosen in negotiations on a higher tree level, it is sufficient that each agent ag_i choose a standard $st(ag_i) \in U(ag_i)$, a formula $\Phi(ag_i) \in L(ag_i)$ and a coefficient $\varepsilon(ag_i) \in [0, 1]$ such that

$$(\Phi(ag_1), \Phi(ag_2), \ldots, \Phi(ag_k), \Phi(ag)) \in Dec\_rule(ag) \tag{95}$$

with standards $st(ag)$, $st(ag_1), \ldots, st(ag_k)$;

$$f(\varepsilon(ag_1), \ldots, \varepsilon(ag_k)) \geq \varepsilon(ag) \tag{96}$$

where f satisfies $unc\_rule(ag)$ with $st(ag)$, $st(ag_1)$, $\ldots$, $st(ag_k)$ and $\varepsilon(ag_1), \ldots, \varepsilon(ag_k)$, $\varepsilon(ag)$.

For a formula

$$\alpha : \langle st(head(Ag)), \Phi(head(Ag)), \varepsilon(head(Ag))\rangle \tag{97}$$

we call an α - *scheme* an assignment of a formula $\alpha(ag) : \langle st(ag), \Phi(ag), \varepsilon(ag)\rangle$ to each $ag \in Ag$ in such manner that formulas (95) and (96) above are satisfied and $\alpha(head(Ag))$ is

$$\langle st(head(Ag)), \Phi(head(Ag)), \varepsilon(head(Ag))\rangle. \tag{98}$$

We denote this scheme with the symbol

$$sch(\langle st(head(Ag)), \Phi(head(Ag)), \varepsilon(head(Ag))\rangle). \tag{99}$$

We say that a selection sel is *compatible* with a scheme

$$sch(\langle st(head(Ag)), \Phi(head(Ag)), \varepsilon(head(Ag))\rangle) \tag{100}$$

when $\mu_o(ag, t)(sel(ag), st(ag)) \geq \varepsilon(ag)$ for each leaf agent $ag \in Ag$ where

$$\langle st(ag), \Phi(ag), \varepsilon(ag)\rangle \tag{101}$$

is the value of the scheme at ag for any leaf $ag \in Ag$.

Any leaf agent realizes its approximate specification by choosing in the subset $Inv \cap U(ag)$ of the inventory of primitive constructs a construct satisfying the specification.

The goal of negotiations can be summarized now as follows.

Proposition 1. *(Sufficiency Criterion) For a given a requirement*

$$\langle st(head(Ag)), \Phi(head(Ag)), \varepsilon(head(Ag))\rangle \tag{102}$$

we have:
if *a selection sel is compatible with a scheme*

$$sch(\langle st(head(Ag)), \Phi(head(Ag)), \varepsilon(head(Ag))\rangle) \tag{103}$$

then

$$sel \models <> \langle st(head(Ag)), \Phi(head(Ag)), \varepsilon(head(Ag))\rangle. \tag{104}$$

The bottom-up communication consists of agents sending to their parents the chosen constructs. The root agent $root(Ag)$ assembles the final construct.

There is a parallelism between the proposed calculi of granules in distributed systems and neural computing. Let us point to some analogies [36, 38].

1. Any elementary team of agents may be regarded as a model of a neuron with inputs supplied by agents $ag_1, ag_2, \ldots, ag_k$, the output returned by ag, and a parameterized family of activation functions represented as rough mereological connectives.

2. Values of rough inclusions 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.

3. Learning in this new kind of a neural network is based also on back-propagation mechanisms in which the incoming signal (a customer specification) is assigned a proper scheme and a proper set of weights is 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.

5 Extracting of AR-Schemes from Data and Background Knowledge

In this section, we present some methods of information granule decomposition aimed at extracting from data decomposition rules. We restrict our considerations to methods based only on experimental data. This approach can be extended to the case of information granule decomposition methods using background knowledge [46, 47].

The search methods discussed in this section return local granule decomposition schemes. These local schemes can be composed using techniques discussed in the previous section. The received schemes of granule construction (which can be also treated as AR-schemes) have also the following property: if the input granules are sufficiently close to input concepts (standards) then the output granule is sufficiently included in the target concept (standard) provided this property is preserved locally (see Proposition 1 in Section 4.2 and [35]).

The above may be formulated in terms of a synthesis grammar [38] with productions corresponding to the local decomposition rules. The relevant derivations over a given synthesis grammar represent AR-schemes. Let us note that synthesis grammars 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. Complexities of membership problems for languages generated by synthesis grammars may be taken ex definitione as complexities of the underlying synthesis processes.

5.1 Granule Decomposition

In this section, we show that in some cases decomposition can be performed using methods for specific rule generation based on Boolean reasoning [14]. Moreover, we present how the decomposition stable with respect to information granule deviations can be obtained.

First, the representation problem for operations on information granules will be discussed. We assume, any (partial) operation $f : G_1 \times \ldots \times G_k \rightarrow H$ with arguments from the sets $G_1, \ldots, G_k$ of information granules and values in the set H of information granules is partially specified by a data table (information system) [27]. In Figure 9 R denotes constraints specifying the domain of f, i.e., arguments of f are composable by means of f if and only if they satisfy constraints from R.

Any row in the data table corresponds to an object being a tuple $(g_1, \ldots, g_k, f(g_1, \ldots, g_k))$, where $(g_1, \ldots, g_k)$ belongs to the domain of f. The attribute values for a given object consist of

1. values of attributes from sets $A_{G_1}, \ldots, A_{G_k}$ on information granules $g_1, \ldots, g_k$ (attributes are extracted from some preassumed feature languages $L_1, , \ldots, L_k$);
2. values of attributes characterizing relations between information granules $g_1, \ldots, g_k$ specifying conditions under which the tuple $(g_1, \ldots, g_k)$ belongs to a relevant part of the domain of f;
3. values of attributes selected for the information granule $f(g_1, \ldots, g_k)$ description.

In this way, partial information about the function f is given. In our considerations, we assume objects indiscernible by condition attributes are indiscernible by decision attribute, i.e., the considered decision table $DT = (U, A, d)$ is consistent [27]. We assume also the representation is consistent with a given function on information granules, i.e., any image obtained by f of the Cartesian product of indiscernibility classes defined by condition attributes is included in a decision indiscernibility class.

Now we explain in what sense the decision table $DT = (U, A, d)$ can be treated as a partial information about the function $f : G_1 \times \ldots \times G_k \to H$. Let for $i = 1, \ldots, k$

$$G_i^{DT} = \{g_i \in G_i : \text{ there exists in } DT \text{ an object } (g_1, \ldots, g_i, \ldots, g_k, h)\}.$$

One can define H^{DT} in an analogous way. The decision table DT defines a function

$$f_{DT} : G_1/IND(A_{G_1}) \times \ldots \times G_k/IND(A_{G_k}) \to H^{DT}/IND(d) \tag{105}$$

by

$$f_{DT}([g_1]_{IND(A_{G_1})}, \ldots, [g_k]_{IND(A_{G_k})}) = [h]_{IND(d)} \text{ iff } (g_1, \ldots, g_k, h) \text{ is an object of } DT. \tag{106}$$

We assume a consistency modeling condition for f is satisfied, namely

$$f([g_1]_{IND(A_{G_1})} \times \ldots \times [g_k]_{IND(A_{G_k})}) = f_{DT}([g_1]_{IND(A_{G_1})}, \ldots, [g_k]_{IND(A_{G_k})}) \tag{107}$$

for any $(g_1, \ldots, g_k) \in G_1^{DT} \times \ldots \times G_k^{DT}$.

The function description can be induced from such a data table by interpreting it as a decision table with the decision corresponding to the attributes specifying the values of the function f.

We assume a family of inclusion relations $\nu_p^i \subseteq G_i \times G_i$, $\nu_p^H \subseteq H \times H$ and a family of closeness relation $cl_p^1, \ldots, cl_p^k, cl_p^H$ for every $p \in [0, 1]$ and $i = 1, \ldots, k$ are given [35]. Let us assume two thresholds t, p are given. We define a relation $Q_{t,p}^{DT}(Pattern_1, \ldots, Pattern_k, \overline{v})$ between granules called patterns $Pattern_1, \ldots, Pattern_k$ from pattern languages $L_1, \ldots, L_k$ for arguments of f and the target pattern $\overline{v}$ representing the decision value vector in the following way:

$$Q_{t,p}^{DT}(Pattern_1, \ldots, Pattern_k, \overline{v}) \tag{108}$$

if and only if the following two conditions are satisfied

$$\nu_p^H(f(Sem_{DT}(Pattern_1) \times \ldots \times Sem_{DT}(Pattern_k)), [\overline{v}]_{IND(d)}) \tag{109}$$

$$card(Sem_{DT}(Pattern_1) \times \ldots \times Sem_{DT}(Pattern_k)) \geq t. \tag{110}$$

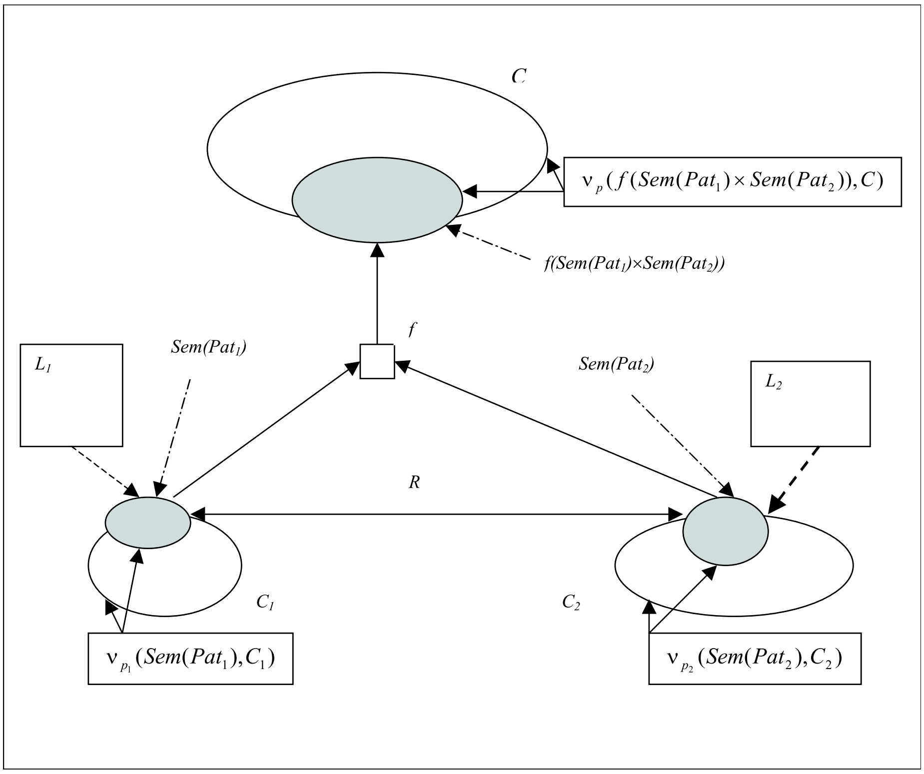

Figure 9: Decomposition of information granule

Let us now consider the following decomposition problem:

Granule decomposition problem

Input:

- two thresholds t, p;
- pattern languages $L_1, \ldots, L_k$;
- a decision table $DT = (U, A, d)$ representing an operation $f : G_1 \times \ldots \times G_k \rightarrow H$ where $G_1, \ldots, G_k$ and H are given sets of information granules;
- a fixed decision value vector $\overline{v}$ represented by a value vector of decision attributes.

Output:

- a tuple $(Pattern_1, \ldots, Pattern_k) \in L_1 \times \ldots \times L_k$ of patterns such that

$$Q^{DT}_{t,p}(Pattern_1, \ldots, Pattern_k, \overline{v}). \tag{111}$$

We consider a description given by means of decision rules extracted from the data table specifying the function f. Any left hand side of decision rule can be divided into parts corresponding to different arguments of the function f. The i-th part, denoted by $Pattern_i$, is specifying a condition which should be satisfied by the i-th argument of f to obtain the function value specified by the decision attributes. For simplicity of considerations we do not consider conditions specifying the relations between arguments. In this way the left hand sides of decision rules describe patterns $Pattern_i$. The semantics of extracted patterns relevant for the target can be defined as the image with respect to f of the Cartesian product of sets $Sem_{DT}(Pattern_i)$, i.e., by $f(Sem_{DT}(Pattern_1) \times \ldots \times Sem_{DT}(Pattern_k))$ (see Figure 9).

One can use one of the methods for decision rule generation, e.g., for generation of minimal rules or their approximations (e.g., in the form of association rules) [14] to obtain such decision rules.

In the former case we receive the most general patterns for function arguments consistent with a given decision table, i.e., the information granules constructed by means of the function f from patterns extracted for arguments are included exactly in the information granule represented by a given decision value vector in the data table (see Figure 9).

In the latter case, we obtain more general patterns for function arguments having the following property: information granules constructed by means of f from such patterns will be included to a satisfactory degree in the information granule represented by a given decision value vector in the data table (see Figure 9).

One of the very important properties of the above discussed operations on information granules is their robustness with respect to the deviations of arguments (see, e.g., [38]). This property can be formulated as follows: if information granule constructed by means of f from the extracted patterns $Pattern_1, \ldots, Pattern_k$ is satisfying the target condition then the information granule constructed from patterns $Pattern'_1, \ldots, Pattern'_k$ sufficiently close to $Pattern_1, \ldots, Pattern_k$, respectively, is satisfying the target condition too. In this way we obtain the following problem:

Robust decomposition problem (RD-problem)

Input:

- thresholds t, p;
- pattern languages $L_1, \ldots, L_k$;
- a decision table $DT = (U, A, d)$ representing an operation $f : G_1 \times \ldots \times G_k \to H$ where $G_1, \ldots, G_k$ and H are given sets of information granules;
- a fixed decision value vector $\overline{v}$ represented by a value vector of decision attributes.

Output:

- a tuple $(p_1, \ldots, p_k)$ of parameters;
- a tuple $(Pattern_1, \ldots, Pattern_k) \in L_1 \times \ldots \times L_k$ of patterns such that

$$Q^{DT}_{t,p}(Pattern'_1, \ldots, Pattern'_k, \overline{v}) \tag{112}$$

if $cl^i_{p_i}(Sem_{DT}(Pattern_i), Sem_{DT}(Pattern'_i))$ for $i = 1, \ldots, k$.

It is possible to search for the solution of the RD-problem by modifying the previous approach of decision rule generation. In the process of rule generation one can impose a stronger discernibility condition by assuming objects to be discernible if their tolerance classes are disjoint. Certainly, one can tune parameters of tolerance relations to obtain rules of satisfactory quality. We would like to stress that efficient heuristics for solving these problems can be based on Boolean reasoning [14].

Searching for relevant patterns for information granule decomposition can be based on methods for tuning parameters of rough set approximations of fuzzy cuts or concepts defined by differences between cuts (see Section 2.4). In this case pattern languages consist of parameterized expressions describing the rough set approximations of *parts* of fuzzy concepts being fuzzy cuts or differences between cuts. Hence, an interesting research direction related to the development of new hybrid rough-fuzzy methods arises aiming at developing algorithmic methods for rough set approximations of such parts of fuzzy sets relevant for information granule decomposition.

An approach presented in this section can be extended on the case of local granule decomposition based on background knowledge [46, 47].

Conclusions

We have outlined a general scheme for rough neuro–computation based on knowledge granulation ideas using rough mereological tools. An important practical problem is a construction of such schemes (networks) for rough neurocomputing and of algorithms for parameter tuning. We now foresee two possible approaches: the one in which we would rely on new, original decomposition, synthesis and tuning methods in analogy to [35] but in the presence of approximation spaces; the second, in which a rough neurocomputing scheme would be encoded by a neural network in such a way that optimization of weights in the neural net leads to satisfactory solutions for the rough neurocomputing scheme (cf. [9] for an attempt in this direction).

We have also discussed an approach for extracting relevant patterns from parameterized schemes of information granule construction consisting of parts from different information sources. The schemes can be also treated as AR-schemes built on the basis of perception by means of information granule calculi. Relevant output patterns (information granules) can be obtained by tuning of the AR-scheme parameters. We have emphasized the necessity of approximation (in an accessible language) of information granules being parts of schemes and expressed in another language called foreign language.

Several research directions are related to the discussed AR-schemes and rough neural networks. We enclose a list of such directions together with examples of problems.

1. *Developing foundations for information granule systems.* Certainly, still more work is needed to develop solid foundations for synthesis and analysis of information granule systems. In particular, methods for construction of hierarchical information granule systems, and methods for representation of such systems should be developed.

2. *Algorithmic methods for inducing parameterized productions.* Some methods have already been reported such as discovery of rough mereological connectives from data (see, e.g., [32]) or methods based on decomposition (see, e.g., [33, 40, 46, 30]). However, these

are only initial steps toward algorithmic methods for inducing of parameterized productions from data. One interesting problem is to determine how such productions can be extracted from data and background knowledge.

3. *Algorithmic methods for synthesis of AR-schemes.* It was observed (see, e.g., [33, 37]) that problems of negotiations and conflict resolutions are of great importance for synthesis of AR-schemes. The problem arises, e.g., when we are searching in a given set of agents for a granule sufficiently included or close to a given one. These agents, often working with different systems of information granules, can derive different granules and their fusion will be necessary to obtain the relevant output granule. In the fusion process, the negotiations and conflict resolutions are necessary. Much more work should be done in this direction by using the existing results on negotiations and conflict resolution. In particular, Boolean reasoning methods seem to be promising [33]. Another problem is related to the size of production sets. These sets can be of large size and it is important to develop learning methods for extracting *small* candidate production sets in the process of extension of temporary derivations out of huge production sets. For solving this kind o problems methods for clustering of productions should be developed to reduce the size of production sets. Moreover, dialog and cooperation strategies between agents can help to reduce the search space for necessary extension of temporary derivations.

4. *Algorithmic methods for learning in rough neural networks.* A basic problem in rough neural networks is related to selecting relevant approximation spaces and to parameter tuning. One can also look up to what extent the existing methods for classical neural methods can be used for learning in rough neural networks. However, it seems that new approach and methods for learning of rough neural networks should be developed to deal with real-life applications. In particular, it is due to the fact that high quality approximations of concepts can be often obtained only through dialog and negotiations processes among agents in which gradually the concept approximation is constructed. Hence, for rough neural networks learning methods based on dialog, negotiations and conflict resolutions should be developed. In some cases, one can use directly rough set and Boolean reasoning methods (see, e.g., [45]). However, more advanced cases need new methods. In particular, hybrid methods based on rough and fuzzy approaches can bring new results [22].

5. *Fusion methods in rough neural neurons.* A basic problem in rough neurons is fusion of the inputs (information) derived from information granules. This fusion makes it possible to contribute to the construction of new granules. In the case where the granule constructed by a rough neuron consists of characteristic signal values made by relevant sensors, a step in the direction of solving the fusion problem can be found in [28].

6. *Adaptive methods.* Certainly, adaptive methods for discovery of productions, for learning of AR-schemes and rough neural networks should be developed [15].

7. *Discovery of multi-agent systems relevant for given problems.* Quite often, the agents and communication methods among them are not given a priori with the problem specification and a challenge is to develop methods for discovery of relevant for given problems multi-agent system structures, in particular methods for discovery of relevant communication protocols.

8. *Construction of multi-agent systems for complex real-life problems.* The challenging problems are related to applying the discussed methodology to real life problems like control of autonomous systems (see, e.g., www page of WITAS project [56]), Web mining problems (see, e.g., [13, 40], sensor fusion (see, e.g., [3, 29, 28]) or spatial reasoning (see, e.g., [7, 6]).

9. *Evolutionary methods.* For all of the above methods it is necessary to develop evolutionary searching methods for (semi-) optimal solutions [15].

10. *Parallel algorithms.* The discussed problems are of high computational complexity. Parallel algorithms searching for AR-schemes and methods for their hardware implementation belong to one important research directions. Moreover, reasoning based on synthesis of AR-schemes using DNA-computing [26] or quantum computing [21] should be developed.

Acknowledgements

I would like to thank to Professor Lech Polkowski for close cooperation on rough mereology, to Professor Jarosław Stepaniuk for the cooperation on information granule models and to Professor James F. Peters for cooperation on sensor fusion methods, insightful comments and a number of clarifying discussions on the presented chapter. Many thanks to Professor Ning Zhong for his kind invitation to write this article.

The research has been supported by the State Committee for Scientific Research of the Republic of Poland (KBN) research grant 8 T11C 025 19 and by the Wallenberg Foundation grant.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. Verkano, Fast discovery of association rules. In: Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. (Eds.), Advances in Knowledge Discovery and Data Mining. The AAAI Press/The MIT Press (1996) 307–328.

[2] L.W. Barsalou, Perceptual symbol systems. Behavioral and Brain Sciences **22** (1999) 577–660.

[3] R.R. Brooks, S.S. Iyengar, Multi-Sensor Fusion. Upper Saddle River, NJ: Prentice-Hall PTR (1998).

[4] P. Doherty, W. Łukaszewicz, A. Skowron, A. Szałas, Combining rough and crisp knowledge in deductive databases. (to appear in [24]).

[5] G. Dong, J. Li.: Interestigness of discovered association rules in terms of neighborhood based unexpectedness, PAKDD (1998) 72–86.

[6] I. Düntsch (Ed.), Spatial Reasoning (special issue). Fundamenta Informaticae **45**(1-2) (2001).

[7] M.T. Escrig, F. Toledo, Qualitative Spatial Reasoning: Theory and Practice. IOS Press, Amsterdam (1998).

[8] J. Fraden, Handbook of Modern Sensors: Physics, Design, and Applications. Springer-Verlag, Berlin (1996).

[9] L. Han, J.F. Peters, S. Ramanna, R. Zhai, Classifying faults in high voltage power systems: A rough–fuzzy neural computational approach. Proceedings of the 7th International Workshop on Rough Sets, Fuzzy Sets, Data Mining and Granular Soft Computing (RSFDGrC'99), Lecture Notes in Artificial Intelligence **1711**, Springer-Verlag, Berlin (1999) 47–54.

[10] S. Harnad (Ed.), Categorical Perception: The Groundwork of Cognition. Cambridge University Press, New York (1987).

[11] T. Hastie, R. Tibshirani, J.H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics, Springer Verlag, Berlin (2001).

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

[13] H. Kargupta, Ph. Chan, Advances in Distributed and Parallel Knowledge Discovery. AIII Press/MIT Press, Cambridge (2001).

[14] J. Komorowski, P. Pawlak, L. Polkowski, A. Skowron, Rough sets: A tutorial. In: S.K. Pal and A. Skowron (Eds.), Rough-Fuzzy Hybridization: A New Trend in Decision-Making, Springer-Verlag, Singapore (1998) 3–98.

[15] J.R. Koza, Genetic Programming II: Automatic Discovery of Reusable Programs. MIT Press, Cambridge, MA (1994).

[16] T.Y. Lin, Granular computing on binary relations I. Data mining and neighborhood systems, In: [34] **1** 107–121.

[17] T.Y. Lin, Y.Y. Yao, L. Zadeh (Eds.), Data Mining, Rough Sets and Granular Computing. Physica-Verlag, Heidelberg (2001).

[18] T.M. Michell, Machine Learning. Mc Graw-Hill, Portland (1997).

[19] H.S. Nguyen, S.H. Nguyen, A. Skowron, Decomposition of task specification. Proceedings of the 11th International Symposium on Foundations of Intelligent Systems, June 1999, Warsaw, Poland, Lecture Notes in Computer Science **1609**, Springer-Verlag, Berlin (1999) 310–318.

[20] H.S. Nguyen, A. Skowron, J. Stepaniuk, Granular computing: A rough set approach. Computational Intelligence **17**(3) (2001) 514–544.

[21] M.A. Nielsen, I.L. Chuang, Quantum Computation and Quantum Information. Cambridge University Press, Cambridge (2000).

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

[23] S.K. Pal, W. Pedrycz, A. Skowron, R. Swiniarski (Eds.), Special Volume on Rough-Neuro Computing. Neurocomputing **36** (2001) 1–262.

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

[25] S.K. Pal, J.F. Peters, L. Polkowski, A. Skowron, Rough-neuro computing: An introduction. In: [24] 16–43.

[26] G. Paun, G. Rozenberg, A. Salomaa, DNA computing. Springer-Verlag, Berlin (1998).

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

[28] Z. Pawlak, J.F. Peters, A. Skowron, Z. Suraj, S. Ramanna, M. Borkowski, Rough measures: Theory and applications. Bulletin of the International Rough Set Society **5**(1–2) (2001) 177–184.

[29] J.F. Peters, S. Ramanna, A. Skowron, M. Borkowski, J. Stepaniuk, Z. Suraj, Sensor fusion: A rough granular approach. In: Proceedings of the Joint 9th International Fuzzy Systems Association (IFSA) World Congress and 20th North American Fuzzy Information Processing Society (NAFIPS) Int. Conf., Vancouver, British Columbia, Canada, 25-28 June 2001, IEEE (2001) 1367–1371.

[30] J.F. Peters, A. Skowron, J. Stepaniuk, Information granules in spatial reasoning. In: Proceedings of the Joint 9th International Fuzzy Systems Association (IFSA) World Congress and 20th North American Fuzzy Information Processing Society (NAFIPS) Int. Conf., Vancouver, British Columbia, Canada, 25-28 June 2001, IEEE (2001) 1355–1361.

[31] L. Polkowski, A. Skowron, Rough mereology: a new paradigm for approximate reasoning. International J. Approximate Reasoning **15**(4) (1996) 333–365.

[32] L. Polkowski, A. Skowron, Rough mereological approach to knowledge-based distributed AI. In: J.K. Lee, J. Liebowitz, and J.M. Chae (Eds.), Critical Technology, Proceedings of the Third World Congress on Expert Systems, February 5-9, Seoul, Korea, Cognizant Communication Corporation, New York (1996) 774–781.

[33] L. Polkowski, A. Skowron, Rough mereological foundations for design, analysis, synthesis, and control in distributed systems. Information Sciences An International Journal **104**(1-2) (1998) 129–156.

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

[35] L. Polkowski, A. Skowron, Towards adaptive calculus of granules. In: [60] **1** 201–227.

[36] L. Polkowski, A. Skowron, Rough mereology in information systems. A case study: Qualitative spatial reasoning. In: L. Polkowski, T.Y. Lin, S. Tsumoto (Eds.), Rough Sets: New Developmentsin Knowledge Discovery in Information Systems, Studies in Fuzziness and Soft Computing **56**, Physica-Verlag, Heidelberg (2000) 89–135.

[37] L. Polkowski, A. Skowron, Rough-neuro computing. Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing (RSCTC'2000), October 16-19, 2000, Banff, Canada, Lecture Notes in Artificial Intelligence **2005**, Springer-Verlag, Berlin (2001) 57–64.

[38] Polkowski L., Skowron A.: Rough mereological calculi of granules: A rough set approach to computation. Computational Intelligence **17**(3) (2001) 472–492.

[39] I. Rock, Indirect Perception, The AAAI Press/ MIT Press, Cambridge (1996).

[40] A. Skowron, Toward intelligent systems: Calculi of information granules. Bulletin of the International Rough Set Society **5**(1–2) (2001) 9–30, see also: [51] 251–260.

[41] A. Skowron, Approximate reasoning by agents in distributed environments. In: [61] 28–39.

[42] A. Skowron, Z. Pawlak, J. Komorowski, L. Polkowski, A rough set perspective on data and knowledge. In: W. Kloesgen, J. Żytkow (Eds.), Handbook of KDD, Oxford University Press (2002) 134–149.

[43] A. Skowron, J. Stepaniuk, Tolerance approximation spaces. Fundamenta Informaticae **27**(2-3) (1996) 245–253.

[44] A. Skowron, J. Stepaniuk, S. Tsumoto, Information granules for spatial reasoning. Bulletin of the International Rough Set Society **3**(4) (1999) 147–154.

[45] A. Skowron, J. Stepaniuk, Information granules: Towards foundations of granular computing. International Journal of Intelligent Systems **16**(1) (2001) 57–86.

[46] A. Skowron, J. Stepaniuk, J.F. Peters, Extracting patterns using information granules. Bulletin of the International Rough Set Society **5**(1) (2001) 135–142.

[47] A. Skowron, J. Stepaniuk, Information granule decomposition. Fundamenta Informaticae **47**(3-4) (2001) 337–350.

[48] A. Skowron, J. Stepaniuk, Information granules and rough-neuro computing. (to appear in [24]).

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

[50] P. Stone, Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer, MIT Press, Cambridge (2000).

[51] T. Terano, T. Nishida, A. Namatame, S. Tsumoto, Y. Ohsawa, T. Washio, New Frontiers in Artificial Intelligence. Lecture Notes in Artificial Intelligence **2253**, Springer-Verlag, Berlin (2001).

[52] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen, H. Mannila, Pruning and grouping discovered association rules. In: MLnet Familiarization Workshop on Statistics, ML and KDD (1995) 47–52.

[53] Waltz D.: The importance of importance. AI Magazine **20**(3) (1999) 18–26.

[54] Wang, P.P. (Ed.): Granular Computing. Wiley, New York (2001).

[55] WWW SPACENET page: http://agora.scs.leeds.ac.uk/spacenet/.

[56] WITAS project Web page: http://www.ida.liu.se/ext/witas/eng.html.

[57] L.A. Zadeh, Fuzzy sets. Information and Control **8** (1965) 333–353.

[58] L.A. Zadeh, Fuzzy logic = Computing with words, IEEE Trans. on Fuzzy Systems **4** (1996) 103–111.

[59] L.A. Zadeh, A new direction in AI: Toward a computational theory of perceptions. AI Magazine **2**(1) (2001) 73–84.

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

[61] N. Zhong, J. Liu, S. Ohsuga, J. Bradshaw (Eds.), Intelligent Agent Technology: Research and Development. Proceedings of the 2nd Asia-Pacific Conference on IAT, Maebashi City, Japan, October 23-26, 2001, World Scientific, Singapore (2001).

[62] N. Zhong, A. Skowron, S. Ohsuga (Eds.), New Directions in Rough Sets, Data Mining, and Granular Soft Computing. Proceedings of the 7th International Workshop (RSFDGr'99), November 1999, Yamaguchi, Japan, Lecture Notes in Artificial Intelligence **1711**, Springer-Verlag, Berlin (1999).