*Fundamenta Informaticae 51 (2002) 157–173*
*IOS Press*


# Towards an Ontology of Approximate Reason

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

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

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

**Sheela Ramanna**§
*Department of Electrical and Computer Engineering*
*University of Manitoba*
*Winnipeg Manitoba R3T 5V6 Canada*

**Abstract.** This article introduces structural aspects in an ontology of approximate reason. The basic assumption in this ontology is that approximate reason is a capability of an agent. Agents are designed to classify information granules derived from sensors that respond to stimuli in the environment of an agent or received from other agents. Classification of information granules is carried out in the context of parameterized approximation spaces and a calculus of granules. Judgment in agents is a faculty of *thinking about* (classifying) the particular relative to decision rules derived from data. Judgment in agents is reflective, but not in the classical philosophical sense (e.g., the notion of judgment in Kant). In an agent, a reflective judgment itself is an assertion that a particular decision rule derived from data is applicable to an object (input). That is, a reflective judgment by an agent is an assertion that a particular vector of attribute (sensor) values matches to some degree the conditions for a particular rule. In effect, this form of judgment is an assertion that a vector of sensor values reflects a known property of data expressed by a decision rule. Since the reasoning underlying a reflective judgment is inductive and surjective (not based on a priori conditions or universals), this form of judgment is reflective, but not in the sense of Kant. Unlike Kant, a reflective judgment is surjective in the sense that it maps experimental attribute values onto the most closely

*This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 194376
Address for correspondence: Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg Manitoba R3T 5V6 Canada
†This work was supported by the State Committee for Scientific Research of the Republic of Poland grant 8 T11C 025 19 and by the Wallenberg Foundation grant
‡This work was supported by the State Committee for Scientific Research of the Republic of Poland grant 8 T11C 025 19
§This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grant 185986

matching descriptors (conditions) in a derived rule. Again, unlike Kant's notion of judgment, a reflective judgment is not the result of searching for a universal that pertains to a particular set of values of descriptors. Rather, a reflective judgment by an agent is a form of recognition that a particular vector of sensor values pertains to a particular rule in some degree. This recognition takes the form of an assertion that a particular descriptor vector is associated with a particular decision rule. These considerations can be repeated for other forms of classifiers besides those defined by decision rules.

**Keywords:** approximation neuron, approximate reason, parameterized approximation space, reflective judgment, pattern recognition, rough sets

# 1. Introduction

Approximate reason is a capability (*faculty*) in an agent designed to classify information granules received from other agents or to classify signals from sensors connected to the agent in the context of an approximation space. An ontology of approximate reason is a study of the nature, constitution and structure of an agent that engages in approximate reasoning. Of particular interest in this ontology is the approximation space that facilitates and underlies this form of reasoning by an agent. Several senses of the term nature apply to the study of an approximate reason, namely, its type, disposition (basic features), capacity, and inherent character. First, the nature of approximate reason refers to its type or class. We make the assumption that each type of approximate reason is tailored relative to a particular form of pattern recognition. This typing of an approximate reason results from its inherent character, i.e., it depends on the sensors available to it in classifying sensor signals and available neurons used to discover patterns in input signals. Second, the capacity of an approximate reason is measured relative to the number of sensors that are its source of input, its particular universe of objects U, its measures of information granules such as rough inclusion, closeness and size defined relative to $U$. Finally, its nature can also be characterized by its parameters, which serve to define its disposition. The structure of an approximation reason refers to the approximation space that underlies it. That is, its structure depends on how such a space is put together (i.e., the choice of universe, operations to construct and measure granules of information, and operation parameters). Approximation spaces have been shown to provide a basis for the design of an approximation neuron [21, 24, 31, 33, 34], and to support a significant form of approximate reasoning [25, 31, 33, 34]. A sensor called an approximation sensor that supplies input to an approximate reason can also be defined with a particular parameterized approximation space (see, e.g., [33, 38, 34]). The distinction between approximation sensors and approximation neurons is based on the source of input. The source of input for an approximation sensor is the environment for an agent. By contrast, the source of input of an approximation neuron is either a sensor or another neuron. The study of approximation sensors and neurons is part of research in rough neurocomputing [8, 12, 21, 24, 26, 31, 32, 33, 34, 36, 38, 44, 47, 48] and its applications [9, 10, 11, 12, 19, 20, 33, 40, 42, 43, 46, 49]. There is still the issue of the constitution of approximate reasoning in this ontology. An approximate reason is constituted by the make-up of its approximation space, by its faculty of judgment, by what can be termed its swarm intelligence (its social make-up or disposition to interact and cooperate with other agents to achieve a goal) and indirectly by its communicative capacity, its wiring to its environment (i.e., its sensor inputs, the make-up, arrangement, quality and quantity of its sensors, its agent input ports that are sources of inputs from other agents, and its output ports with connections that make it possible to send granules of information to other agents in the same *colony*). This communication feature of an agent has been inspired by the Milner model [7]. The *swarming* feature of an approximate reason is a direct outcome of the distributed nature of the calculus of granules introduced in [8, 41] and elaborated in the context of intelligent systems in [31]. The idea of a cooperative approximate reason is akin to what is known as swarm intelligence [2]. The constitution of approximate reason (i.e., the members of a colony of reasoning agents where the communication ability of each agent is facilitated by a set of sensors and a set of ports) has a great deal to do with what is termed its faculty of judgment.

In the context of approximate reasoning by an agent, the term faculty refers to a certain power of reasoning (e.g., searching for an appropriate rule based on pattern matching-feature values with descriptors) that has been endowed in an agent by its designer. Judgment in agents is reflective, but not in the classical philosophical sense (e.g., the notion of judgment proposed by Kant [5]). Judgment in agents is a faculty of *thinking about* (classifying) the particular relative to decision rules derived from data. In an agent, a reflective judgment itself is an assertion that a particular decision rule derived from data is applicable to an object (input). The exercise of judgment by an approximate reason may entail a consequent classification of a particular disposition of features (i.e., conjunction of descriptors) or a consequent response or action (e.g., a moral decision or beginning of an action sequence). It is appropriate to consider systems of cooperating agents that collectively exhibit what can be described as swarm intelligence. In the context of an approximate reason, swarm intelligence refers to the positioning of an approximate reason in an agent in a distributed, problem-solving system of agents. In effect, an agent is designed to mimic the behavior of cooperating individuals (e.g., humans, insects such as bees and ants) that live in a colony [2]. In considering an ontology of approximate reason, an individual agent is viewed as a part of a distributed system of communicating agents. An agent that engages in approximate reasoning is an independent process that interacts with its environment and other agents in its system by constructing information granules that approximate the information received from its sensor inputs or from other agents and by transmitting its constructed granules to other agents. Information granule approximation provides an agent with a means of knowledge discovery. For simplicity, the ontology of approximate reason is restricted to very simple forms of agents (approximation neurons) that synthesize elementary granules from their inputs. An elementary granule conveys a single *piece* or *clump* of information (e.g., a condition for a rule, a measurement of rough inclusion).

In this article, the consideration of an ontology of approximate reason is limited to the structural aspects. That is, the classification of an approximate reason (its type, capacity) and of the faculty of judgment of an approximate reason are outside the scope of this article. A study of the structure of approximate reason leads to a consideration of an underlying calculus of granules that makes it possible for an agent to engage in approximate reasoning about information granules. A framework for approximate reasoning is briefly presented in Section 2. The idea of an approximation neuron is considered in Section 3.

## 2. Framework for Approximate Reasoning

In laying the groundwork for approximate reasoning about information granules in the context of rough set theory [13]-[17], a brief introduction to an adaptive calculus of granules is given in this section. Information granule construction and parameterized approximation spaces provide the foundation for a model of rough neurocomputing [12, 31]. A fundamental feature of this model is the design of neurons that engage in knowledge discovery. Mechanically, such a neuron returns granules (synthesized knowledge) derived from input granules.

### 2.1. A Calculus for Intelligent Systems

In working towards a design of intelligent systems, an adaptive calculus of granules has been introduced [29, 33, 34] based on rough mereology [28], and elaborated in [27, 30, 31, 33, 35, 36, 37, 38, 34, 32]. A calculus of granules is a system for approximating, combining, describing, measuring, reasoning about, and performing operations on granules by intelligent computing units called agents. In the calculus of granules, the term granule denotes an assemblage of objects aggregated together by virtue of their indistinguishability, similarity, or functionality. Intuitively, a granule is also called a clump [48]. The term calculus has been attributed to G.W. v. Leibniz [3]. Leibniz thought of a calculus as an instrument of discovery inasmuch as it provides a system for combining, describing, measuring, reasoning about and performing operations on objects of interest such as terms in a logical formula in a logical calculus or infinitesimally small quantities in differential calculus ([1, 6]). The calculus of classes described by Alfred Tarski [47] shares some of the features found in the calculus of granules. The term class is synonymous with set, an assemblage of distinct entities, either individually specified or which satisfy certain specified conditions (e.g., equivalence class of $y$ consisting of all objects equivalent to $y$). It is Georg Cantor's description of how one constructs a set that comes closest to what we have in mind when we speak of a granulation. That is, a set is the result of collecting together certain well-determined objects of our perception or our thinking into a single whole (the objects are called elements of a set) [3]. In a calculus of classes, the kinds of classes (e.g., the empty class and the universal class), relations between classes (e.g., inclusion, overlap, identity), and operations on classes $(\cup, \cap, -)$ are specified. Similarly, the calculus of granules distinguishes between kinds of granules (e.g., elementary granules, set-, concept-, and granule-approximations), relations between granules (e.g., inclusion, overlap, closeness), and operations on granules (e.g., granule approximation, decomposition). It should be observed that in the case of information granules, we can not use crisp equality in comparing granules. Instead, we are forced to deal with the concepts of similarity, closeness, and being a part to a degree when considering relations between granules.

The calculus of granules includes a number of features not found in the calculus of classes, namely, a system of agents, communication of granules of knowledge between agents and the construction of granules by agents. To some extent, the new calculus of granules is similar to the agent-based, value-passing calculus of communicating systems proposed by Robin Milner [7]. In Milner's system, an agent is an independent process possessing input and output ports. Agents communicate via channels connecting the port of one agent with the port of another agent. Milner's calculus is defined by a tuple $(A, L, Act, X, V, K, J, \varepsilon)$ where $A$ is a set of names, $L$, a set of labels, $Act$, a set of actions, $X$, a set of agent variables, $V$, a set of values, $K$, a set of agent constants, $J$, an indexing set; and $\varepsilon$ is a set of agent expressions. This calculus includes a grammar for formulating expressions. Even though adaptivity, granules of knowledge, information granulation, parameterized approximations, and a hierarchy of relations of being a part to a degree (fundamental features of the calculus of granules) are not found in Milner's calculus, it is possible to enrich Milner's system to obtain a variant of the calculus of granules.

The fundamental feature of a granulation system is the exchange of information granules between agents by means of transfer functions induced by rough mereological connectives extracted from information systems.

## 2.2. Calculus of Granules

A calculus of granules has been introduced to provide a foundation for the design of information granulation systems. The keystone in such systems is the granularity of knowledge for approximate reasoning by agents [31]. An agent is modeled as a computing unit that receives input from its sensors and from other agents, acquires knowledge by discovering (constructing) information granules and by granule approximation, learns (improves its skill in acquiring knowledge) and adapts (adjusts in granulation parameters predicates in response to changing sensor measurements and feedback from other agents). Agents engage in approximate reasoning about information granules not only because of inexactness of information in a granule but also because a gain in efficiency in reasoning can result if it is enough to deliver approximate solutions, sufficiently close to the ideal solutions. For two sets $X, Y \subseteq U$ (the universe of an information system), define standard rough inclusion using $\mu(X, Y) = card(X \cap Y)/card(X)$ when $X$ is non-empty and $\mu(X, Y) = 1$, otherwise. A simple granule of knowledge of type $(\mu, B, C, tr, tr')$ is a pair $(\alpha, \alpha')$ of descriptor conjunctions over $B, C$ respectively, where $\mu$ is the standard rough inclusion, $B$ and $C$ are subsets of $A$ (attributes or sensors of an information system), and $tr, tr' \in [0, 1]$ are thresholds on functions defined with respect to $\mu$ such that $\mu([\alpha]_B, [\alpha']_C) \geq tr$ and $\mu([\alpha']_C, [\alpha]_B) \geq tr'$ ($[\alpha]_B, [\alpha']_C$ are sets of objects satisfying $\alpha, \alpha'$, respectively). For example, the assertion that $Gr(\mu, B, C, tr, tr', \alpha, \alpha')$ is true in the case where $(\alpha, \alpha')$ is a $(\mu, B, C, tr, tr')$ granule of knowledge. There are several sources of adaptivity in the scheme defined by a calculus of granules. First, there is the possibility that changes can be made in parameters $\mu, B, C, tr, tr'$ in the granulation predicate $Gr(\mu, B, C, tr, tr', \alpha, \alpha')$ for any agent $ag \in Ag$ (a set of agents). Second, new granules can be constructed by any agent in response to a changing environment. Third, new rough inclusion measures can be instituted by an agent by changing, for example, the parameters in t-norm and s-norm used in defining $\mu$. The possibility that any agent can make one or more of these changes paves the way towards an adaptive calculus of granules [29]. A recently formulated rough-fuzzy neural network has partially realized this idea with an adaptive threshold relative to a set of real-value attributes without employing rough inclusion [8].

Each agent (neuron) distils its knowledge from granulated (fused) sensor measurements, from granulated signals from other agents, and from approximate reasoning in classifying its acquired granules. An agent communicates its knowledge over channels connected to other agents. An agent (neuron) learns by adjusting accessible parameters in response to feedback from other agents. Let $Ag$ be a non-empty set of agents. In describing the elements of a calculus of granules, we sometimes write $U$ instead of $U(ag)$, for example, where $U$ (and $U(ag)$) denotes a non-empty set of granules (universe) known to agent $ag \in Ag$ [31]. Similarly, when it is clear from the context: $Inv$, $St$, $A$, $M$, $L$, $link$, $O$, $AP\_O$, $Unc\_rel$, $Unc\_rel$, $H$, $Dec\_rule$, $lab$ are a shorthand for $Inv(ag)$, $St(ag)$, $A(ag)$, $M(ag)$, $L(ag)$, $Link(ag)$, $O(ag)$, $AP\_O(ag)$, $Unc\_rel(ag)$, $Unc\_rule(ag)$, $H(ag)$, $Dec\_rule(ag)$, respectively. The calculus of granules establishes a scheme for a distributed system of agents that is characterized by the following tuple:

$$Scheme = (U, Inv, St, Ag, L_{rm}, A, M, L, link, O, AP\_O, Unc\_rel, Unc\_rel, H, Dec\_rule, lab)$$

where $U$ denotes a non-empty set of granules (universe) known to agent $ag \in Ag$; $Inv$, denotes an inventory of elementary objects available to $ag$; $St$, a set of standard of objects for $ag$; $Ag$, a set of agents; $L_{rm}$, a rough mereological logic [28]; $A$, an information system of $ag$; $M$, a pre-model of $L_{rm}$ for $ag$; $L$, a set of unary predicates of $ag$; $link$, a string denoting a team of agents communicating objects (input) to an agent for granulation; $O$, a set of operations of an agent; $Unc\_rel$, a set of uncertainty relations; $H$, a strategy for producing uncertainty rules from uncertainty relations; $Dec\_rule$, a set of granule decomposition rules; and $lab$, a set of labels (one for each agent $ag \in Ag$). Calculus of granules provides a computational framework for designing neural networks in the context of a rough set approach to approximate reasoning and knowledge discovery. The original idea of an open world model for inductive learning by agents [13] has been enriched by considering a distributed system of agents that stimulate each other by communicating granules of knowledge gleaned from granules received from other agents.

An approximate rough mereology with its own logic $L_{rm}$ (syntax, grammar for its formulas, axioms, and semantics of its models) provides a formal treatment of being a part in a degree. This paves the way towards a study of granule inclusion degree testing and measures of the closeness of granules implemented by cooperating agents [33]. The calculus of granules is considered adaptive to the extent that the construction of information granules by a distributed system of interacting agents will vary in response to variations in the approximate reasoning by agents about their input signals (input granules). Agents usually live and learn inductively in an open system like the one described by Pawlak [13]. Let $(Inv, Ag)$ denote a distributed system of agents where $Inv$ denotes an inventory of elementary objects and $Ag$ is a set of intelligent computing units (agents). Let $ag \in Ag$ be an agent endowed with tools for reasoning and communicating with other agents about objects within its scope. These tools are defined by components of the agent label (denoted *lab*) [29] such that

$$lab(ag) = (\mathbf{A}(ag), \mathrm{M}(ag), L(ag), Link(ag), St(ag), O(ag),$$
$$AP\_O(ag), Unc\_rel(ag), Unc\_rule(ag), H(ag), Dec\_rule(ag))$$

where

- $\mathbf{A}(ag) = (U(ag), A(ag))$ is an information system relative to agent $ag$, where the elements of universe $U(ag)$ is a finite, non-empty set of granules containing elements of the form $(\alpha, [\alpha])$ such that $\alpha$ is a conjunction of descriptors and $[\alpha]$ denotes its meaning in $\mathbf{A}(ag)$ [13]. It is also possible that the objects of $U(ag)$ are complex granules.

- $M(ag) = (U(ag), [0,1], \mu_0(ag))$ is a pre-model of $L_{rm}$ with a rough inclusion $\mu_0(ag)$ on the universe $U(ag)$. The notation $L_{rm}$ denotes a rough mereological logic.

- $L(ag)$ is a set of unary predicates (properties of objects) in a predicate calculus interpreted in the set $U(ag)$. Further, formulas of $L(ag)$ are constructed as conditional formulas of logics $L_B$ where $B \subset U(ag)$.

- $Link(ag)$ is a collection of strings of the form $ag_1 ag_2 \cdots ag_k ag$ denoting a team of agents such that $ag_1 ag_2 \cdots ag_k$ are the children of agent $ag$ in the sense that $ag$ can assemble complex objects (constructs) from simpler objects sent by agents $ag_1, ag_2, \cdots, ag_k$.

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

- $O(ag) \subseteq \{o | o : U(ag_1) \times U(ag_2) \times \cdots \times U(ag_k) \to U(ag)\ is\ operation\ at\ ag\}$.

- $AP\_O(ag)$ is a collection of pairs of the form

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

  where $o(ag,t) \in O(ag)$, $n$ is the arity of $o(ag,t)$, $t = ag_1, ag_2, \cdots, ag_k \in Link(ag)$, $AS_i(o(ag,t), in)$ is a parameterized approximation space corresponding to the $i^{th}$ argument of

$o(ag,t)$ and $AS(o(ag,t), out)$ is a parameterized approximation space for the output of $o(ag,t)$. The meaning of $o(ag,t)$ is that an agent performs an operation enabling the agent to assemble from objects $x_1 \in U(ag_1)$, $x_2 \in U(ag_2)$, $\cdots$, $x_k \in U(ag_k)$ the object $z \in U(ag)$ that is an approximation defined by $AS(o(ag,t), out)$ to $o(ag,t)(y_1, y_2, \cdots, y_k) \in U(ag)$ where $y_i$ is the approximation of $x_i$ defined by $AS_i(o(ag,t), in)$. One may choose here either a lower or upper approximation.

- $Unc\_rel(ag)$ is a set of uncertainty relations $unc\_rel_i$ of type

$$\begin{array}{l}(o_i(ag,t), \rho_i(ag), ag_1, ..., ag_k, ag,\\ \mu_o(ag_1), ..., \mu_o(ag_k), \mu_o(ag),\\ st(ag_1), ..., st(ag_k), st(ag))\end{array}$$

of agent $ag$ where $ag_1$, $ag_2$,$\cdots$,$ag_k \in Link(ag)$, $o_i(ag,t) \in O(ag)$ and $\rho_i$ is such that

$$\rho_i((x_1, \varepsilon_1), \cdots, (x_i, \varepsilon_k), (x, \varepsilon))$$

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

$$o_i(ag,t)(st(ag_1)_i, \cdots, st(ag_k)_i) = st(ag)_i.$$

Values of the operation $o$ are computed in three stages. First, approximations to input objects are constructed. Next, an operation is performed. Finally, the approximation to the result is constructed. A relation $unc\_rel_i$ provides a global description of this process. In practice, $unc\_rel_i$ is composed of analogous relations corresponding to the three stages. The relation $unc\_rel_i$ depends on parameters of approximation spaces. Hence, to obtain satisfactory decomposition (similarly, uncertainty and so on) rules, it is necessary to search for satisfactory parameters of approximation spaces. This search is analogous to weight-tuning in traditional neural computations.

- $Unc\_rule(ag)$ is a set of uncertainty rules $unc\_rule_i$ of type

**if** $o_i(ag,t)(st(ag_1)_i, \cdots, st(ag_k)_i) = st(ag)_i$ and $x_1 \in U(ag_1), \cdots, x_k \in U(ag_k)$
satisfy the conditions $\mu_o(x_j, st(ag_j)_i) \geqslant \varepsilon(ag_i)$ for $i = 1, ..., k$

**then** $\mu_o(o_i(ag,t)(x_1, ..., x_k), st(ag)_i) \geqslant f_i(\varepsilon(ag_1), ..., \varepsilon(ag_k))$

where $ag_1$, $ag_2, \cdots, ag_k \in Link(ag)$ and $f_i : [0,1]^k \to [0,1]$ is so called *rough mereological connective*. Uncertainty rules provide functional operators (approximate mereological connectives) for propagating uncertainty measure values from the children of an agent to the agent. The application of uncertainty rules is in negotiation processes where they inform agents about plausible uncertainty bounds.

- $H(ag)$ is a strategy that produces uncertainty rules from uncertainty relations.

- $Dec\_rule(ag)$ is a set of decomposition rules

$$(\Phi\,(ag_1)\,,...,\Phi\,(ag_k)\,,\Phi\,(ag))$$

of type $(o_i\,(ag,t)\,,ag_1,...,ag_{k,}ag)$ of agent ag where

$$\Phi\,(ag_1) \in L\,(ag_1)\,,...,\Phi\,(ag_k) \in L\,(ag_k)\,,\Phi\,(ag) \in L(ag,)$$

$ag_1, ag_2, \cdots, ag_k \in Link(ag)$, and there exists a collection of standards

$$st(ag_1)_i, \cdots, st(ag_k)_i, st(ag)_i$$

such that $o_j(ag,t)(st(ag_1)_i, \cdots, st(ag_k)_i) = st(ag)_i$ and these standards are satisfying

$$\Phi(ag_1), \cdots, \Phi(ag_k), \Phi(ag),$$

respectively. Decomposition rules are decomposition schemes. That is, such rules describe the standard $st(ag)_i$ and standards $st(ag_1)_i, \cdots, st(ag_k)_i$ from which the standard $st(ag)$ is assembled under $o_i$ relative to predicates that these standards satisfy.

## 3. Parameterized Approximation Spaces

In this section, the fulfillment of an ontology of approximate reason stems from the consideration of granular computing in the context of parameterized approximation spaces as a realization of an adaptive granule calculus. This realization is possible due to the introduction of a parameterized approximation space in the design of a reasoning system for an agent. A brief introduction to parameterized approximation spaces is given in this section. It has been pointed out that there is an analogy between calculi of granules in distributed systems and rough neural computing [31, 33, 34], namely:

1. An agent with input and output ports providing communication links with other agents provides a model for a neuron $\eta$ (analogously, agent $ag$) with inputs supplied by neurons $\eta_1, \ldots, \eta_k$ (analogously agents $ag_1, \ldots, ag_k$), responds with output by $\eta$. The output $\eta$ is designed together with a parameterized family of activation functions represented as rough connectives. In effect, a neuron resembles the model of an agent proposed by Milner [7].

2. Values of rough inclusions are analogous to weights in traditional neural networks.

3. Learning in a system governed by an adaptive calculus of granules is in the form of a back propagation where incoming signals are assigned a proper scheme (granule construction) and a proper set of weights in negotiation and cooperation with other neurons.

In this section, a step towards the realization of an adaptive granule calculus in a rough neurocomputing scheme is described along the lines of [31]. In the scheme for information granule construction in a distributed system of cooperating agents, weights are defined by approximation spaces. In effect, each agent (neuron) in such a scheme controls a local parameterized approximation space.

**Definition 3.1.** A parameterized approximation space is a system $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$ where

- $\#, \$$ denote vectors of parameters,
- $U$ is a non-empty set of objects,
- $I_{\#} : U \to \wp(U)$ is an uncertainty function, where $\wp(U)$ denotes the powerset of $U$, and
- $\nu_{\$} : \wp(U) \times \wp(U) \to [0, 1]$ denotes rough inclusion

The uncertainty function defines for every object $x$ in $U$, a set of similarly described objects. A constructive definition of an uncertainty function can, for example, be based on the assumption that some metrics (distances) are given on attribute values. A set $X \subseteq U$ is definable in $AS_{\#,\$}$ if it is a union of some values of the uncertainty function. The rough inclusion function $\nu_{\$}$ defines the value of inclusion between two subsets of $U$. Using rough inclusion, the neighborhood $I_{\#}(x)$ can usually be defined as a collection of close objects. It should also be noted that for some problems it is convenient to define an uncertainty set function of the form $I_{\#} : \wp(U) \to \wp(U)$. This form of uncertainty function works well in signal analysis, where we want to consider a domain over sets of sample signal values.

## 3.1. A Threshold–Based Approximation Space

A threshold–based approximation space is presented below.

**Example 3.1.** This example is derived from [38]. Consider an information system $IS = (U, A)$. Let $A = \{a\}$, where the attribute $a$ is real-valued, and let $U \subseteq \Re$ be a non-empty set of reals. Consider two elementary granules $a(x) \in [v_1, v_2]$ and $a(x) \in [v'_1, v'_2]$ for intervals of real numbers where $v_1 < v_2$ and $v'_1 < v'_2$. We want to measure the degree of inclusion of the granule $a(x) \in [v_1, v_2]$ in the granule $a(x) \in [v'_1, v'_2]$ (i.e., we assume elements of $R$ and neighborhoods of objects are such intervals). First, we introduce an overlapping range function $r_a$ to measure the overlap between a pair of subintervals.

$$r_a([v_1, v_2], [v'_1, v'_2]) = \max(\{\min(\{v_2, v'_2\}) - \max(\{v_1, v'_1\}), 0\}).$$

Let $a \in A$, and $\delta$ be a real number. The notation $\lfloor a(x)/\delta \rfloor$ denotes the greatest integer less than or equal to $a(x)/\delta$. The following uncertainty function $I_{a,\delta} : U \to \wp(U)$ is defined as follows.

$$\mathrm{I}_{a,\delta}(x) = \begin{cases} [v_1, v_2], \text{ if } \lfloor a(x)/\delta \rfloor \in [v_1, v_2] \\ [v'_1, v'_2], \text{ if } \lfloor a(x)/\delta \rfloor \in [v'_1, v'_2] \\ [0, 0], \text{ otherwise} \end{cases}$$

A rough inclusion function $\nu : \wp(U) \times \wp(U) \to [0, 1]$ is then defined as follows.

$$\nu\left(I_{a,\delta}(x), I_{a,\delta}(x')\right) = \frac{r_a\left(I_{a,\delta}(x), I_{a,\delta}(x')\right)}{v_2 - v_1}$$

Elementary granule $a(x) \in [v_1, v_2]$ is included in an elementary granule $a(x) \in [v'_1, v'_2]$ to a degree at least $t_a$ if, and only if

$$\nu\left(I_{a,\delta}(x), I_{a,\delta}(x')\right) \geq t_a.$$

This version of a parameterized approximation space depends on a threshold parameter $t_a$ used as a standard by which rough inclusion values can be judged. That is, the degree of inclusion of one interval in another interval is judged relative to the threshold $t_a$ such that $\nu(I_{a,\delta}(x), I_{a,\delta}(x')) \geq t_a$. Changes in the parameter $\delta$ in the uncertainty function $I_{a,\delta}$ and in the threshold parameter $t_a$ in the inclusion model $\nu$ in this sample approximation space change the result. Values of these parameters can be learned during training (i.e., during training, adjustments in the parameters are made to achieve an improvement in the classification of elementary input granules).

### 3.2. Indistinguishability Relation

To begin, let $IS = (U, A)$ be an infinite information system where $U$ is a non-empty subset of the reals and $A$ is a non-empty, finite set of real valued attributes, where $a : U \to V_a$ for every $a \in A$. Let $\delta > 0$ be a positive real number. In addition, let $a(x) \geq 0$. The parameter $\delta$ serves as a *neighborhood* size on real-valued intervals. Reals within the same subinterval bounded by $k\delta$ and $(k+1)\delta$ are considered indistinguishable. For each $B \subseteq A$, there is associated an equivalence relation $Ing_{A,\delta}(B)$ defined by

$$Ing_{A,\delta}(B) = \{(x, x') \in U^2 | \forall a \in B \lfloor a(x)/\delta \rfloor = \lfloor a(x')/\delta \rfloor\}$$

If $(x, x') \in Ing_{A,\delta}(B)$, we say that objects $x$ and $x'$ are indistinguishable from each other relative to attributes from $B$. Let $Id = \{id\}$ where $a(x) = x$ for $x \in U$. Such sensor $id$ is introduced to avoid the situation, where there is more then one stimulus for which a sensor takes the same value (see the example in the next section). We can write

$$Ing_{A,\delta}(B \cup Id) = \left\{ (x, x') \in \Re^2 | \lfloor x/\delta \rfloor = \lfloor x'/\delta \rfloor \ \wedge \ \forall a \in B \ \lfloor a(x)/\delta \rfloor = \lfloor a(x'/\delta \rfloor \ \right\}$$

The notation $[x]_B^\delta$ denotes equivalence classes of $Ing_{A,\delta}\ (B)$. Further, partition $U/Ing_{A,\delta}(B)$ denotes the family of all equivalence classes of relation $Ing_{A,\delta}(B)$ on $U$. For $X \subseteq U$, the set $X$ can be approximated only from information contained in $B$ by constructing a $B$-lower and a $B$-upper approximation denoted by $\underline{B}X$ and $\overline{B}X$ respectively, where $\underline{B}X = \left\{x \mid [x]_{B \cup Id}^\delta \subseteq X\right\}$ and $\overline{B}X = \left\{x \mid [x]_{B \cup Id}^\delta \cap X \neq \emptyset\right\}$.

In some cases, we find it necessary to use a sensor reading $y$ (an ordinate or *vertical* value) instead of stimulus $x$. In such cases, we create an equivalence class consisting of all points (ordinate values) for which sensor readings are *close* to $y$ and define

$$[y]_B^\delta = \left\{ x \in \Re | \forall a \in B \ \left\lfloor \frac{a(x)}{2\delta} \right\rfloor = \left\lfloor \frac{y}{2\delta} \right\rfloor \right\}$$

This is quite important in cases where we want to extract information granules relative to sensor signals (sensor measurements rather than sensor stimuli then hold our attention). Certainly, the relation $Ing_{A,\delta}(B)$ is an equivalence relation.

### 3.3. More Sample Approximation Spaces

In this section, we consider a parameterized approximation space defined relative to the indistinguishability relation and uncertainty set functions where the domain of such a function is the power-set (set of subsets) of $U$.

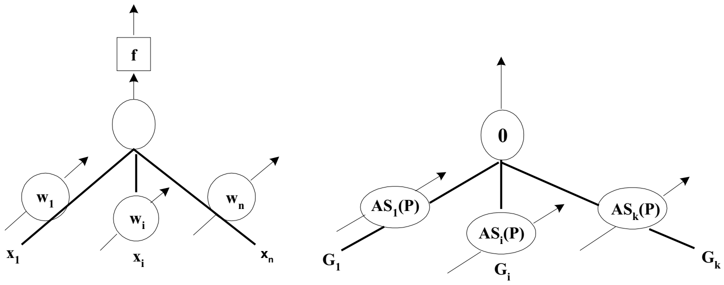


Figure 1. Comparison of Classical and Granular Network Architectures

**Example 3.2.** *Indistinguishability-Based Approximation Space*. Consider a parameterized approximation space with an uncertainty set function $I_{B,\delta}$ that constructs a granule (namely, an upper approximation) based on a knowledge of $B$ and indistinguishability relation with parameter $\delta$ and a rough inclusion function $\nu$ with threshold parameter $t_0$. We write $\nu_{t_0}(X, Y)$ to denote $\nu(X, Y) > t_0$. For simplicity, the traditional $\overline{B}X$ is constructed by a method named *constructUPP* as part of the definition of $I_{B,\delta}$, where $I_{B,\delta} : \wp(U) \to \wp(U)$ for $I_{B,\delta}(X) = constructUPP(X) = \overline{B}X$, and let $Y = [y]_B^\delta$. In this example, rough inclusion $\nu : \wp(U) \times \wp(U) \to [0, 1]$ is defined by

$$\nu(I_{B,\delta}(X), Y) = \frac{\rho\,(I_{B,\delta}(X) \cap Y)}{\rho\,(Y)}.$$

where $\wp$ denotes a measure on $\wp(U)$.

The rough inclusion of granule $I_{B,\delta}(X)$ is acceptable in granule $Y$ provided that the following constraint is satisfied.

$$\nu_{t_0}(X, Y),\ \text{i.e., } \nu(X, Y) > t_0.$$

The essential thing to notice about this variant of $I_{B,\delta}$ in this example is that it constructs the granule $\bar{B}X$ from its domain $X$. The rough inclusion function then measures the degree of overlap between $\bar{B}X$ and a set represented by $Y$. The composition of the set $Y$ is not treated in this example. The parameters for $I_{B,\delta}$ are $\delta$ (tolerance) and set of attributes (features) $B$. The parameter for $\nu$ is the threshold $t_a$.

## 4. Approximation Neuron Models

Parameters in a parameterized approximation space may be treated as counterparts of weights in a traditional neural network, and each instance of such a granule-producing agent with a parameterized approximation space design parallels the architecture of a neuron in a conventional neural network. In Figure 1, $w_1, \ldots, w_n, \sum, f$ denote weights, aggregation operator, and activation function of a classical neuron, respectively, while $AS_1(P), \ldots, AS_k(P)$ denote parameterized (by $P$) approximations spaces where agents process input granules $G_1, ..., G_k$ and $O$ denotes on operations that produce the output of a granular network. To carry this analogy a step further, parameters of an approximation space should be learned to induce the relevant information granules.

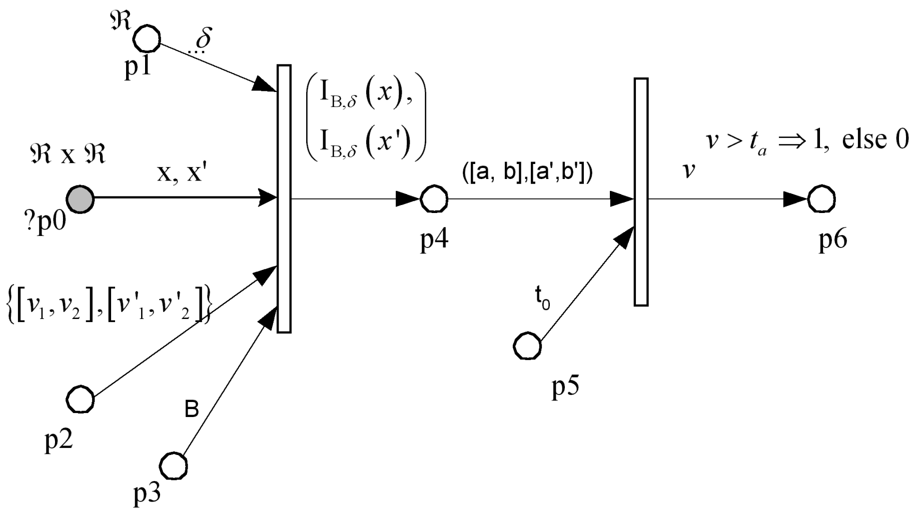


Figure 2. Threshold–Based Neuron Design

A neuron designed relative to a parameterized approximation space is called an approximation neuron. In its simplest form, such a neuron constructs an elementary granule as a result of approximating the information received from its inputs. In more elaborate forms of an approximation neuron, for example, the output of the neuron may take the form of a rule derived from a condition vector of inputs, or a reduct derived from a received decision table, or a set of rules derived from a received reduct and a received decision table. The particular configuration of such a neuron depends on the instantiation of the approximation space and particular activation functions used in the design of the neuron. The design of an approximation neuron changes each time we modify the definition of the uncertainty function $I_{\#}$ and the rough inclusion function $\nu_{\$}$ as well as the parameters in $\#$ and \$ chosen for these functions. In this section, we consider two fairly basic models of elementary approximation neurons (EA-neurons). An EA-neuron constructs an elementary granule as its output. An elementary information granule is an information granule that contains a single *piece* of information (e.g., an attribute (sensor) value, a condition for a rule, a measurement of rough inclusion). The output of such an approximation neuron is a rough inclusion value.

## 4.1. Threshold–Based Approximation Neuron

It is possible to design a simple prototype neural network where changes in the parameters rather than changes in weights provide a basis for training in the context of a parameterized approximation space. That is, we want to consider a threshold–based approximation neuron with an elementary granule as its output, namely, $\nu(I_{B,\delta}(x), I_{B,\delta}(x'))$ (see Figure 2).

In Figure 2, a Petri net is given to model an approximation neuron. This is an example of a rough Petri net. The label $?p0$ in Figure 2 denotes a receptor process (always input ready) *connected* to the environment of an agent. For more details about this form of a Petri net, see [23].

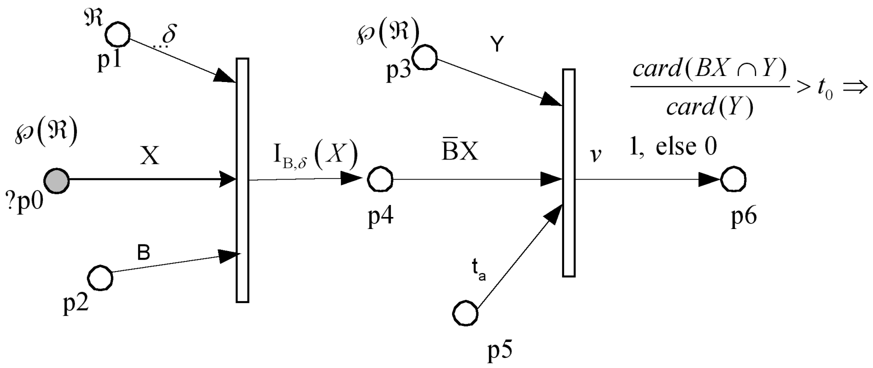


Figure 3. Indistinguishability-Based Neuron Design

## 4.2. Indistinguishability–Based Approximation Neuron

It is also possible to design a prototype neural network based on the indistinguishability relation. In the form of neural computation described in this section, training entails changing $\delta$ (interval width) until the rough inclusion function value exceeds the threshold $t_0$. That is, an indistinguishability-based approximation neuron can be designed with an elementary granule as its output, namely, $\nu\,(I_{B,\delta}(X), Y)$, where $I_{B,\delta}(X)$ computes $\overline{B}X$ and on the output we have 1 if $card(\overline{B}X \cap Y)/card(Y) > t_0$ and 0, otherwise (see Figure 3).

## 4.3. Approximation Neuron Training Model

In this section, a model for training is limited to a single approximation neuron. Up to this point, no guarded transitions have been used in the Petri net models of approximation neurons. In this section, we consider a Petri net approximation neuron training model with three transitions, namely, *neuron*, *train* and a third transition named *approx* (see Figure 4).

Except for the guards and one extra communication transition, the Petri net in Figure 2 is represented by the neuron transition together with its inputs and output in Figure 4. The firing of neuron results in the computation of the rough inclusion of the input $X$ relative some set $Y$, and either an initial value of $\delta$ or a changed value of $\delta$ *propagated back* from the transition train to place $p1$. Changes in $\delta$ occur during training and are the result of executing a procedure named $BP$ (see Figure 4). The term back propagation (BP) is typically used to describe training of a multi-layer perceptron using gradient descent applied to a sum-of-squares error function. Training in the basic neuron in Figure 4 is much simpler, since we only need to modify one parameter, namely, $\delta$. If the transition *train* in Figure 4 had more than one rough inclusion computation as input and more than one $\delta$ to adjust, then it would be appropriate to consider some form of traditional back propagation method in adjusting the $\delta$ values. Transition *train* is enabled if $\nu(I_{B,\delta}(X), Y) < t_0$ (i.e., the rough inclusion falls below the threshold $t_0$). Each time transition *train* fires, a new $\delta$ value is computed by the error function $BP_\nu(\delta)$. The output place labeled $p1$ for the transition *train* in Figure 4 is an alias for the input place $p1$ for the transition neuron. In the simple neural training model in Figure 4, the modeling of what happens to a neuron output in the case

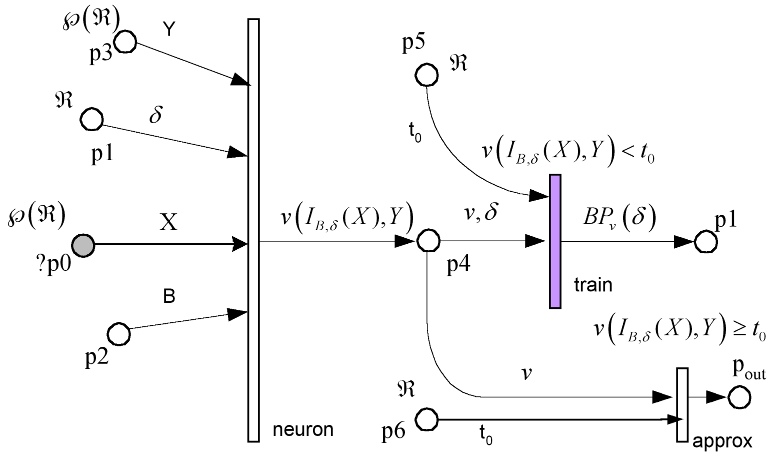


Figure 4. Basic Neuron Training Model

where the rough inclusion value falls in the interval $[t_0, 1]$ is represented by the transition *approx*, which is enabled in those instances where the neuron output is at or above the required threshold.

## Conclusion

The underpinnings of an ontology of approximate reason based on rough set theory were considered in this paper. We focus upon the structural features of approximate reason. This form of reasoning in a distributed system of agents is considered in the context of a calculus of granules and parameterized approximation spaces. Communicating agents interact with one another by receiving granules that require classification. Agents classify received information granules in the context of parameterized approximation spaces and a calculus of granules. We also discussed indistinguishability of points of uncountable sets and a proposed model for approximation neurons. We briefly mention other features of approximate reason (typing, disposition, capacity, inherent character). These features are largely outside the scope of this paper. Reflective judgment by agents is another important facet of approximate reason that merits a detailed study at some point in the future.

## References


[1] Bocheński, I.M., *A history of formal logic*, Chelse Publishing Co., NY, 1956.

[2] Bonabeau, E., Dorigo, M., Theraulaz, G., *Swarm intelligence: From natural to artificial systems*, Oxford University Press, Oxford, 2000.

[3] Gellert, W., Kustner, H., Hellwich, M., Kastner, H., *The VNR concise encyclopedia of mathematics*, Van Nostrand Reinhold Co., London, 1975.

[4] Halmos, P.R., *Measure theory*, D. Van Nostrand Co., Inc., London, 1950.

[5] Kant, E., *Critique of judgment*, (trans. by J.H. Bernard), Hafner Press, Toronto, 1951.

[6] Leibniz, G.W., *Opuscles et fragments inedits de Leibniz*, edited by L. Couturat. Paris, 1903, 256.

[7] Milner, R., *Communication and concurrency*, Prentice-Hall, Upper Saddle River, NJ, 1989.

[8] Mitra, S., Mitra, P., Pal, S.K., Evolutionary modular design of rough knowledge-based network with fuzzy attributes, *Neurocomputing: An International Journal*, **36**, 2001, 45–66.

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

[10] Nguyen, H.S., Skowron, A., Szczuka, M.S., Situation identification by unmanned aerial vehicle, in: [56], 49–56.

[11] Pal, S.K., Polkowski, L., Skowron, A. (Eds.), *Rough-Neuro Computing: Techniques for Computing with Words*, Springer–Verlag, Berlin, 2002 (to appear).

[12] Pal, S.K., Peters, J.F., Polkowski, L., Skowron, A., Rough-Neuro Computing: An Introduction, in: [11], 16–43.

[13] Pawlak, Z., *Rough sets: Theoretical aspects of reasoning about data*, Kluwer Academic Publishers, Boston, MA, 1991.

[14] Pawlak, Z., Skowron, A., Rough membership functions. in: R. Yager, M. Fedrizzi, J. Kacprzyk (Eds.), *Advances in the Dempster-Shafer Theory of Evidence*, John Wiley & Sons, NY, 1994, 251–271.

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

[16] Pawlak, Z., *On rough derivatives, rough integrals, and rough differential equations*, ICS Research Report **41/95**, Institute of Computer Science, Nowowiejska 15/19, 00-665 Warsaw, Poland, 1995.

[17] Pawlak, Z., Elementary rough set granules: Towards a rough set processor, in: [11], 5–15.

[18] Peters, J.F., Skowron, A., Suraj, Z., An application of rough set methods to automatic concurrent control design, *Fundamenta Informaticae* **43**(1–4), 2000, 269–290.

[19] Peters, J.F., Ramanna, S., Skowron, A., Borkowski, M., Wireless agent guidance of remote mobile robots: Rough integral approach to sensor signal analysis, in: [55], 413–422.

[20] Peters, J.F., Ramanna, S., Borkowski, M., Skowron, A., Approximate sensor fusion in a navigation agent, in: [54], 500–504.

[21] Peters, J.F., Ramanna, S., Suraj, Z., Borkowski, M., Rough neurons: Petri net models and applications, in: [11], 474–493.

[22] Peters, J.F., Skowron, A., Stepaniuk, J., Rough granules in spatial reasoning, in: *Proc. 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, 1355–1361.

[23] Peters, J.F., Ramanna, S., A rough set approach to assessing software quality: Concepts and rough Petri net model, in: S.K. Pal, A. Skowron (Eds.), *Rough Fuzzy Hybridization: A New Trend in Decision-Making*, Springer-Verlag, Berlin, 1999, 349–380.

[24] Peters, J.F., Skowron, A., Suraj, Z., Han, L., Ramanna, S., Design of rough neurons: Rough set foundation and Petri net model, in: Z.W. Ras, S. Ohsuga (Eds.), Foundations of Intelligent Systems, Lecture Notes in Artificial Intelligence **1932**, Springer Verlag, Berlin, 2000, 283–291.

[25] Peters, J.F., Skowron, A., L. Han, S. Ramanna, Towards rough neural computing based on rough membership functions: Theory and Application, in: [56], 572-579.

[26] Peters, J.F., Han, L., Ramanna, S., Rough neural computing in signal analysis, *Computational Intelligence* **1**3, 2001, 493–513.

[27] Polkowski, L., Skowron, A., Approximate reasoning about complex objects in distributed systems: Rough mereological formalization, in: W. Pedrycz, J.F. Peters (Eds.), *Computational Intelligence in Software Engineering, Advances in Fuzzy Systems-Applications and Theory* **16**, World Scientific, Singapore, 1998, 237–267.

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

[29] Polkowski, L., Skowron, A., Towards adaptive calculus of granules, in: *Proc. of the Sixth Int. Conf. on Fuzzy Systems*, Anchorage, Alaska, 4-9 May, 1998, 111–116.

[30] Polkowski, L., Skowron, A., Calculi of granules based on rough set theory: Approximate distributed synthesis and granular semantics for computing with words, in: [53], 20–28.

[31] Polkowski, L., Skowron, A., Rough-neuro computing, in: [56], 57–64.

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

[33] Skowron, A., 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.

[34] Skowron, A., Approximate reasoning by agents in distributed environments, in: [54], 28-39.

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

[36] Skowron, A., Stepaniuk, J., Information granules: Towards foundations of granular computing, *International Journal of Intelligent Systems* **16**(1), Jan. 2001, 57–104.

[37] Skowron, A., Stepaniuk, J., Tolerance approximation spaces, *Fundamenta Informaticae* **27**, 1996, 245–253.

[38] Skowron, A., Stepaniuk, J., Information granules and approximation spaces, in: *Proc. of the 7th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Paris, France, 6-10 July 1998, 1354–1361.

[39] Skowron, A., Stepaniuk, J., Peters, J.F., Hierarchy of information granules, in: H.D. Burkhard, L. Czaja, H.S. Nguyen, P. Starke (Eds.), *Proc. of the Workshop on Concurrency, Specification and Programming*, Oct. 2001, Warsaw, Poland, 254–268.

[40] Skowron, A., Stepaniuk, J., Peters, J.F., Extracting patterns using information granules, *Bulletin of the International Rough Set Society* **5**(1–2), 2001, 135–142.

[41] Skowron, A., Swiniarski, R.W., Information granulation and pattern recognition, in: [11], 600–632.

[42] Swiniarski, R.W., *RoughNeuralLab, software package developed at San Diego State University*, San Diego, California, U.S.A., 1995.

[43] Swiniarski, R.W., Rough sets and neural networks application to handwritten character recognition by complex Zernike moments, in [56], 617–624.

[44] Swiniarski, R.W., Hargis, L., Rough sets as a front end of neural networks texture classifiers, *Neurocomputing* **36**, 2001, 85–103.

[45] Szczuka, M. S., Rough sets and artificial neural networks. in: L. Polkowski, A. Skowron (Eds.), *Rough Sets in Knowledge Discovery 2: Applications, Cases Studies and Software Systems*, Physica Verlag, Berlin, 1998, 449–470.

[46] Szczuka, M. S., Refining classifiers with neural networks, *International Journal of Intelligent Systems*˜**16**(1), Jan. 2001, 39–56.

[47] Szczuka, M., Wojdyłło, P., Neuro-wavelet classifiers for EEG signals based on rough set methods, *Neurocomputing* **36**, 2001, 103–122.

[48] Szczuka, M.S., Rough set methods for constructing artificial neural networks, in: B.D. Czejdo, I.I. Est, B. Shirazi, B. Trousse (Eds.), *Proc. of the 3rd Biennial Joint Conf. on Engineering Systems Design and Analysis* **7**, Montpellier, France, 1-4 July 1996, 9–14.

[49] Wojdyłło, Wavelets, rough sets and artificial neural networks in EEG analysis. Lecture Notes in Artificial Intelligence **1424**, Berlin, Springer-Verlag, 1998, 444-449.

[50] Tarski, A., *Introduction to logic and to the methodology of deductive sciences*, Oxford University Press, NY, 1965, IV, 68–78.

[51] Terano, T., Nishida, T,. Namatame, A., Tsumoto, S., Ohsawa, Y., Washio, T,. *New frontiers in artificial intelligence*, Lecture Notes in Artificial Intelligence **2253**, Springer-Verlag, Bedrlin, 2001.

[52] Zadeh, L.A., Fuzzy logic = computing with words, *IEEE Trans. on Fuzzy Systems* **4**, 1996, 103–111.

[53] Zhong, N., Skowron, A., Ohsuga, S. (Eds.), *New Directions in Rough Sets, Data Mining, and Granular Soft Computing*, Lecture Notes in Artificial Intelligence **1711**, Springer-Berlag, Berlin, 1999.

[54] Zhong, N., Liu, J., Ohsuga, S., Bradshaw, J. (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.

[55] Zhong, N., Yao, Y.Y., Liu, J., Ohsuga, S. (Eds.), *Web Intelligence*, Lecture Notes in Artificial Intelligence **2198**, Springer-Verlag, Berlin 2001,

[56] Ziarko, W., Yao, Y.Y. (Eds.), *Proc. Int. Conf. on Rough Sets and Current Trends in Computing*, Lecture Notes in Artificial Intelligence **2005**. Berlin: Springer-Verlag, 2001.