On the Generalized Deduction, Induction and Abduction as the Elementary Reasoning Operators within Computational Semiotics

Ricardo R. Gudwin
DCA-FEEC-UNICAMP
Caixa Postal 6101, 13.083-970 Campinas – SP, Brasil
e-mail: gudwin@dca.fee.unicamp.br

ABSTRACT: We first elaborate on the definition of an elementary unit of knowledge, called here knowledge unit. Then we define three knowledge operators, knowledge extraction, knowledge generation and knowledge selection, aimed at being the elementary ways of reasoning. These are compared to the traditional deduction, induction and abduction reasoning operators, resulting in the proposal that our knowledge operators could be viewed as generalized interpretations of the standard deduction, induction and abduction reasoning procedures. We finalize by proposing them as the building blocks for universal intelligent systems.

KEYWORDS: Computational Semiotics, Deduction, Induction, Abduction, Knowledge Units

1. INTRODUCTION

Computational Semiotics refers to the attempt of emulating the semiosis cycle within a digital computer. Among other things, this is done aiming for the construction of autonomous intelligent systems able to perform intelligent behavior, what includes perception, world modeling, value judgement and behavior generation. There is a claim that most part of intelligent behavior should be due to semiotic processing within autonomous systems, in the sense that an intelligent system should be comparable to a semiotic system. Mathematically modeling such semiotic systems is being currently the target for a group of researchers studying the interactions encountered between semiotics and intelligent systems.

The key issue on this study is the discovery of elementary, or minimum units of intelligence, and their relation to semiotics. Some attempts have been made aiming for the determination of such elementary units of intelligence, i.e., a minimum set of operators that would be responsible for building intelligent behavior within intelligent systems. These attempts include Albus' outline for a theory of intelligence [1] and Meystel's GFACS algorithm [2]. In this paper, we discuss on an alternative set of operators, namely knowledge extraction, knowledge generation and knowledge selection, that should be understood as generalizations for the deduction, induction and abduction reasoning methods respectively. In this sense, knowledge extraction is viewed as an abstraction for deduction, knowledge generation is an abstraction for induction and knowledge selection is an abstraction for abduction, leading us to universal operators of generalized deduction, generalized induction and generalized abduction. We discuss on how these concepts are connected, and how these abstractions fit the current understanding of what is deduction, induction and abduction. We also show how we can use such operators to build Albus-like architectures and the GFACS algorithms. Associated to Gudwin's hierarchy of types of knowledge, we speculate on how they would serve as the building blocks of universal intelligent systems.

2. KNOWLEDGE UNITS

There are many attempts to define what would be the exact semantics for the term "knowledge", what would be the difference between "knowledge" and "information", and what would be the elementary pieces of knowledge, sometimes called knowledge units. We provide our own definition: "A Knowledge Unit is a granule of information encoded into a structure". The exact understanding of such definition needs some philosophical background, to be provided in the following paragraphs.

First of all, we consider the existence of an **environment,** or **real world**, which is defined as a set of dynamic continuous phenomena running in parallel. We assume we are not able to know this environment in its whole. The part of environment we are able to know, in a process that goes through our sensors, is called our Umwelt [3]. The Umwelt, also called our **sensible environment**, is our best possible comprehension of reality. It is very important to stress, though, that Umwelt is not reality. It comprises only our best understanding of reality. In this sense, our sensors are the primary source of information that flows into our mind. These sensors do provide a continuous and partial information about phenomena occurring in Umwelt. From this continuous source of information, we extract what we call singularities, i.e., clusters of information that can be aggregated under a single concept. These singularities are discrete entities that model, in a specific level of resolution, the phenomena

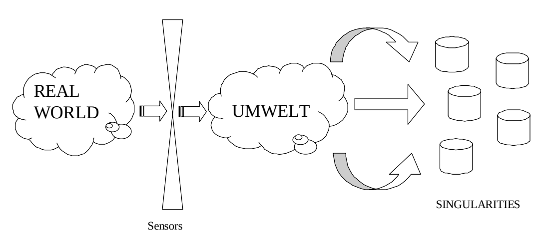

Figure 1- Singularities Extraction

occurring in the world. We can also view these singularities as an intensional definition for what we are calling here knowledge units (figure 1).

Once those granules of information (singularities) are identified in the Umwelt, they need to be encoded to become a knowledge unit (as given by our original definition). This codification needs a **representation space** and an **embodiment vehicle** (structure) that is placed within the representation space. These structures may be abstracted to mathematical structures (figure 2), i.e., (a) numbers, (b) lists, (c) trees and (d) graphs.

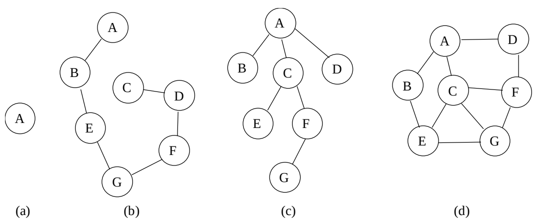

Figure 2 – Mathematical Structures

Each structure has a place at the representation space (figure 3).

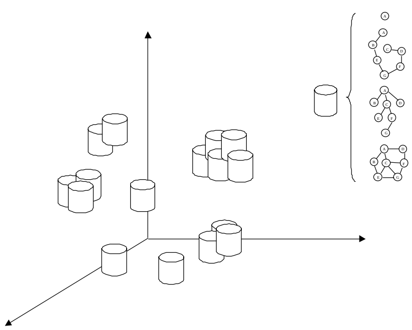

Figure 3 – Representation Space

The view shown in figure 3 is though, our view of representation space **after** an interpretation. Before interpretation, the representation space is more like in figure 4: a set of values occupying a place in space. To build a knowledge unit, then, we need what we call a "focus of attention" mechanism, which selects a closed region of representation space, that is our primary field of interpretation.

Then comes what we call the first interpretation problem, (illustrated in figure 5). How a set of values embraced by the focus of attention is going to be interpreted ? This is called the structural identification problem.

A second interpretation problem, that comes once we identified the structure within our focus of attention, is related with the semantic identification of information within the structure. If the data represented by the structure respects to a direct modeling of an environment phenomenon, this knowledge unit is called an **icon**. If it gives the localization within the representation space of another structure, it is called an **index**. And, if it is a key in a conversion table, it is a **symbol**. In this case, we will need to use the conversion table (that should be another structure in the representation space),

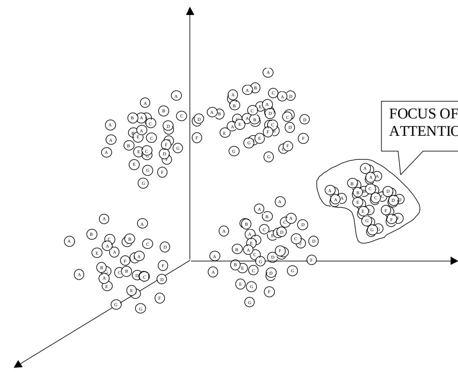

Figure 4 – Focus of Attention and Structures Identification

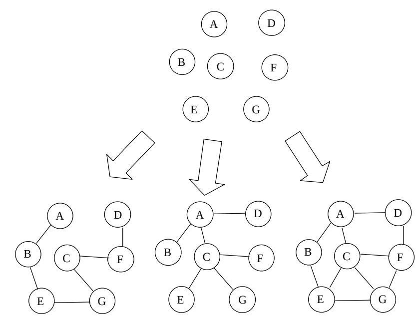

Figure 5 – Interpretation Problem

in order to locate the icon representing the phenomenon we want to refer to.

Elementary knowledge units are formed due to these singularity extraction mechanisms. More elaborate knowledge units, though, are formed by the application of knowledge processing operators, illustrated in figure 6.

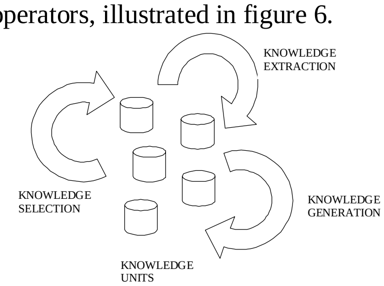

Figure 6 – Knowledge Processing Operators

These knowledge processing operators are from 3 basic types, that we are going to call here generalized deduction, generalized induction and generalized abduction. We are going to address them in the following sections.

3. A TAXONOMY FOR KNOWLEDGE UNITS

Knowledge units can be classified according to a taxonomy of types of knowledge [4,5,6,7]. This taxonomy is inspired on the classification of different types of signs, given by Peirce, and the different dimensions for an interpretant, by Morris. Basically, each type of knowledge is associated with a different kind of concept (or idea), that is, the semantic that is intrinsic to a given knowledge type. We may have static and dynamic knowledge types. The types referred as rhematic and dicent [4,5,6,7] are static, in the sense that they only exist as data. The knowledge types known as arguments are dynamic. They are dynamic in the sense that they do not only exist as data, but also performs transformations in the system. A direct analogy of static and dynamic types is the classification of information within a computer memory as data and code. Static types are just like data in a computer memory. Dynamic types are like code in a computer memory. They can be seen as data or code, depending on the context being analyzed. Dynamic knowledge units are the primary source of activity in a semiotic system. They are responsible for the extraction of singularities and also for the further discovery and manipulation of new knowledge units within the semiotic system.

4. ARGUMENTATIVE KNOWLEDGE

Dynamic knowledge types are classified into the knowledge hierarchy within a particular branch, involving the family of argumentative knowledge units. Basically, an argumentative knowledge unit is a piece of knowledge whose semantic is the understanding of knowledge manipulation. In other words, it indicates how to produce new pieces of knowledge, taking as input a set of knowledge units.

5. GENERALIZATION AND SPECIALIZATION

Knowledge units from some types of knowledge can be compared to each other by means of an "abstraction" partial order relation ($\prec$). In this sense, if some knowledge units a and b are related by $a \prec b$, then we say that b is an abstraction of a. Or, in other words, that b is a generalization of a, and a is an specialization of b. These concepts are fundamental for our definitions of knowledge extraction, knowledge generation and knowledge selection. The key issue for understanding the abstraction relation is to remember the two possible ways of defining a set. We may define a set using an *extensional definition*, where we explicitly list all elements within the set. This way of definition is fine for finite sets only. There is also the intensional definition of a set, where we define a set as the collection of all possible points satisfying a condition. Using an *intensional definition*, we may represent a whole infinite set with only a finite number of parameters. This implies in an encoding able to convert from the intensional representation for elements in the extensional representation. For example, let a set S be defined as $S = \{(x,y) \in R^2 \mid y = 2x^3+7x+1 \}$. This is an intensional definition for set S. We may represent set S, as the tuple $(2,0,7,1)$, that encodes all the information necessary to reconstruct the points (x,y) belonging to S. Suppose now a knowledge unit $a = (1,10)$ and a knowledge unit $b = (2,0,7,1)$. If we interpret a as being a pair in R^2, and b as being the parameters representing the infinite set S, we may say that $a \prec b$, because knowledge unit b comprises not only a, but a whole set of pairs obeying the same relationship. Notice that we may have also a knowledge unit $c = (0,1,1,10,2,31)$, that should be decoded as the set $T = \{(0,1),(1,10),(2,31)\}$, and we would also have $c \prec b$, and $a \prec c \prec b$. This is only a clue for understanding the nature of abstraction operator. The way in which we decided to encode set S in tuple $(2,0,7,1)$ is not trivial. We may view this operation as a kind of data compression operator. Each knowledge unit b that can be expanded to other knowledge units a_i through some particular interpretation is said to be a generalization of them. And the a_i's are said to be specializations of b.

6. THE ELEMENTARY KNOWLEDGE OPERATORS

We propose a minimum set of operators as a conceptual basis for the construction of intelligent systems. These are the "knowledge extraction" operator, "knowledge generation" operator and "knowledge selection" operator. In fact, they are not exactly operators, but classes of operators.

6.1 Knowledge Extraction

Suppose a knowledge unit b and a knowledge unit a, such that $a \prec b$. Then, a function *fke* that maps b (remembering that b is a structure, e.g. a structured number) onto a: $a = fke(b)$, is called a knowledge extraction operator.

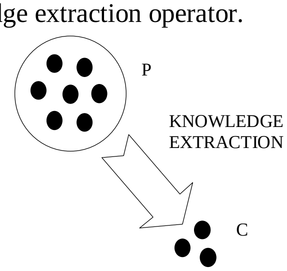

Figure 7 – Knowledge Extraction

In figure 7 we have an example of knowledge extraction. From a set P of knowledge units, called the “premise”, the operator extracts a set C of knowledge units, called the “conclusion”. We call this operation knowledge extraction, because the extensional definition of the knowledge units in C is a subset of the extensional definition of knowledge units in P. So, it “extracts” from P only part of its semantic content.

6.2 Knowledge Generation

Suppose now the same a and b above, and also a function *fkg* that maps a onto b, i.e.,$b = fkg(a)$. Then *fkg* is called a knowledge generation operator. Usually, this kind of operator is not single input/single output, but comprises a set of input knowledge units and a corresponding set of output knowledge units. Then, we have e.g. that $(b_1,...,b_m) = fkg(a_1,a_2,...,a_n)$.

In figure 8, the premise P is the collection of knowledge units a_i and the conclusion C is the collection of knowledge units b_i. One of the particularities of this operator is that the extensional definition of the knowledge units in C necessarily contains elements that are not originally in the extensional definition of knowledge units in P. They have been added during the process of knowledge generation. This is what characterizes the knowledge generation operation. This process can be done in many different ways, including

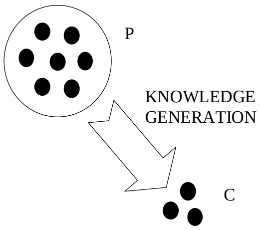

Figure 8 – Knowledge Generation

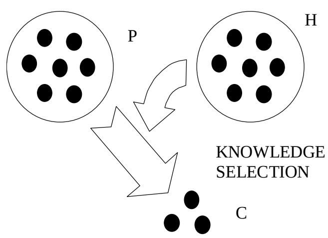

Figure 9 – Knowledge Selection

combination of knowledge units, **fusion** of knowledge units, **transformation** of knowledge units (including insertion of noise), **interpolation, fitting and topologic expansion** of knowledge units or any hybridism of these techniques. A lot of examples may be set here. For example, the interpolation of functions adds all the points surrounding the original samples. The fitting of functions neither requires the inclusion of sample points. Topologic expansion from a number to a fuzzy number, adds all the points in the vicinity of it. The learning algorithm of a neural network transforms a set of weights describing a nonlinear classifying function into another, by adjusting it in order to include new sample points.

6.3 Knowledge Selection

Suppose now, that we have a set of input knowledge units, $\{a_1,a_2,...,a_n\}$, and a set of candidates $\{c_1,c_2,...,c_m\}$ for being the output. Suppose also a function *fks*, that performs the selection among the candidates: $b = fks(a_1,a_2,...,a_n,c_1,c_2,...,c_m)$, in the sense that b is one of the c_i's, and the a_i's are used to evaluate and choose among the c_i's. Then, *fks* is called a knowledge selection operator

In figure 9, the knowledge units a_i are within set P (Premise) and the c_i's are within set H, also called set of hypothesis. The knowledge units b_i's (more than one, in this case), are selected among the c_i's, and are indicated by C (Conclusion).

Notice that there is a special case when there is only one c_i in H, and then the selection becomes a "validation". In this case, the knowledge units in P are used to validate the new knowledge unit being output at C. In this sense, if they are not able to validate c_i, it produces nothing as output.

7. THE ELEMENTARY KNOWLEDGE OPERATORS AND THE CLASSICAL REASONING OPERATORS

There is a close connection between our elementary knowledge operators and the classical reasoning operators, namely deduction, induction and abduction. Despite not being the usual definition, our knowledge extraction operator could be associated with a generalized deduction, in the sense that it comprises the behavior of traditional deduction operators, but extends it a little bit, allowing any kind of knowledge unit as input. The same is valid to generalized induction. Traditional understanding of what is induction is within our definition for knowledge generation. It is important to note, although, that our knowledge generation operator also includes some operators that are not traditionally accepted as performing "induction". In this sense, our definition generalizes the idea of induction, to include any kind of procedure that generates knowledge units that are more abstract than its inputs. And finally, maybe the more controversial claim, our knowledge selection can be compared to standard abduction operators. Abduction is often viewed as inference to the "best" explanation. It usually comprises the generation of a set of hypothesis, their evaluation and the selection of the best one. Our claim is that this view of abduction has interconnections with induction. The part related to the generation of a set of hypotheses is clearly an induction (or, better saying, a generalized induction). The real abductive work is effectively the evaluation and selection of the best hypothesis. This view conflicts with some approaches found in the literature.

7.1 Deduction and Knowledge Extraction

Usually deduction is considered within the scope of mathematical logic, generally propositional or first-order logic. The most common form of deduction is the application of "modus ponens". An example of deduction is given below.

$$\frac{\begin{array}{l}\text{man(Socrates)}\\ \text{man(x)} \rightarrow \text{mortal(x)}\end{array}}{\text{mortal(Socrates)}}$$

In this case, if we assume that it is true that *Socrates* is a *man*, and it is also true that if *something* is a *man* then this *something* is *mortal*, we may deduce that *Socrates* is *mortal*.

Notice how this definition is within our definition of knowledge extraction. The knowledge that Socrates is mortal is also within the premise that Socrates is a man and that all man are mortal. So, $mortal(Socrates) \prec \{man(Socrates) \cup man(x) \rightarrow mortal(x)\}$.

Propositions are dicent knowledge units, and usually deduction is applied only to dicent knowledge. Our definition allows for the inclusion of other types of knowledge, as it generalizes for any operation that explicits knowledge that is already present at the premises. In this sense, we call our knowledge extraction operation as a generalized deduction. Notice how the calculation of a function is somewhat alike modus ponens. We know $f(x)$, and also x_0 .Then we deduce $y_0 = f(x_0)$. This is more or less the same that is happening at modus ponens, but in another context. In this sense, our generalized deduction does exactly what its name says, it generalizes the idea of deduction to other types of knowledge units.

7.2 Induction and Knowledge Generation

Induction is known as a process of producing a general proposition on the ground of a limited number of particular propositions such that these become special instances of the former. It concerns a process of going from particular to universal, from concrete to abstract. This process is also called generalization [8,9].

The main process of induction starts with a set of samples (of propositions, of concepts, of points, etc), that is used as a seed to the general concept to be generated, by different possible techniques. This is similar to the procedure we called here knowledge generation, with one difference. In ours knowledge generation, not necessarily the generated knowledge units have to be a generalization of the samples. The samples are used only as a seed, to any procedure that uses it to generate new knowledge units. This is the difference between induction, in its classic understanding, and our proposition known as knowledge generation.

What comprises pragmatically this difference? Procedures that are not actually recognized as induction may fit the more general definition. For example, the crossover and mutation procedures used in genetic algorithms can be included in knowledge generation, despite they are not usually referred as being induction. In this sense, our knowledge generation would not be properly called induction, but a form of *generalized* induction, where transformations of knowledge generating new knowledge are addressed in a general sense, not only when this new knowledge is a generalization of the input knowledge. Of course, this includes induction in its classical form.

7.3 Abduction and Knowledge Selection

Abduction is one of the least studied ways of reasoning, and sometimes one of the most misunderstood. Even Peirce has not put it clearly, sometimes calling it *abduction*, sometimes calling it *retroduction*. Different authors focus on different aspects of abduction. For example, Ram & Leake [10] focus on the whole process of explanatory reasoning, which includes anomaly detection, explanatory hypothesis construction, hypothesis verification and selection of best hypothesis. Van der Lubbe [11] understands abduction as the inverse of deduction, and treats it by what he calls pseudo-abduction, inverting knowledge trees and performing deduction. We would like to consider abduction not only applied to dicent knowledge (logical calculus), but also to other types of knowledge, specifically rhematic knowledge [4]. This motivated us in trying to discover new abstractions for what exactly happens during abduction, which leaded us to the proposal of understanding the generalized abduction as a step of selection/validation. The example of Ram & Leake is interesting, because they treat abduction as the whole process of explanatory reasoning. The question is: are all the sub-steps parts of abduction, or are they only a context in which abduction is performed ? A deeper analysis shows that most steps indicated by Ram & Leake as parts of the abduction process could be classified within other modalities. For example, anomaly detection, is clearly a step of deduction. Explanatory hypothesis construction could be classified as a generalized induction. The step that concentrates the essence of abduction would be then only hypothesis verification and selection. This set out our position. We propose understanding abduction not as the process of anomaly detection (which would be a deduction), neither as the process that generates hypothesis (which would be a induction), but as the test and selection of those hypothesis. This is not the current understanding, but we believe that it is a bettern suited insight, when we analyze the contribution of the three generalized operators in building intelligent systems.

8. BUILDING INTELLIGENT SYSTEMS

Through the use of object networks [6,7], we are able to create systems where knowledge units are explicited by means of objects, and especially argumentative knowledge units are active objects. In this section, we illustrate how to build intelligent systems using such ideas, using object networks to model an architecture similar to the one given by Albus, and also the GFACS algorithm given by Meystel. Meystel has proposed GFACS as the elementary unit of intelligence. Here, we show how to build intelligent systems using more elementary knowledge units.

8.1 Knowledge Processing within an Object Network

Knowledge units are represented within an object network by means of objects, i.e., tokens that are at the same time data and code, put in places along the network. Figure 10 presents a small section of an object network.

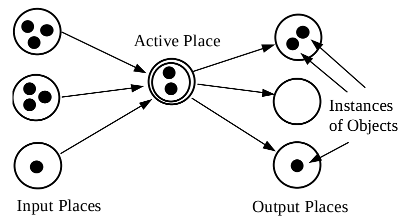

Figure 10 – Section of an Object Network

The objects within the active place in figure 10 are objects able to perform computations. They select among the objects at input places, use the information within them to build other objects and to modify their own structure, and releases the new objects created (or transport an input object) to output places. The used objects may be released back to input places, may be destroyed or may be transported to output places. There are two main functions involved in this process. The first one is the selection functions which assign the objects to be used as source of information, and the other is the internal function, which takes this information, plus the information inside the structure of the own active object and uses them to generate a new object. Conceptually, both selection function and internal function may be any function. This makes the active object a perfect representation of argumentative knowledge, able to perform each or all of the three generalized ways of reasoning: deduction, induction and abduction. The selection function would perform the role of abduction, and the internal function would perform the role of deduction or induction, depending on the output being a knowledge unit more specific or more generic than the input. This makes object networks a very good tool for representing and modeling intelligent systems.

One interesting property of an object network is that due to its conceptualization, an active object may generate another active object, in the same or in another place in the network. In this case, the active part of the network is able to be changed, increased or decreased dynamically, with the network behavior. This property allows the modeling of systems that are able to change its own structure, like learning and adaptive systems. Examples of object networks implementing fuzzy systems, neural networks and genetic algorithms are showed at [12].

8.2 Building Intelligent Systems

An illustration of an intelligent system, in Albus' sense [1], implemented through object networks is shown in figure 11.

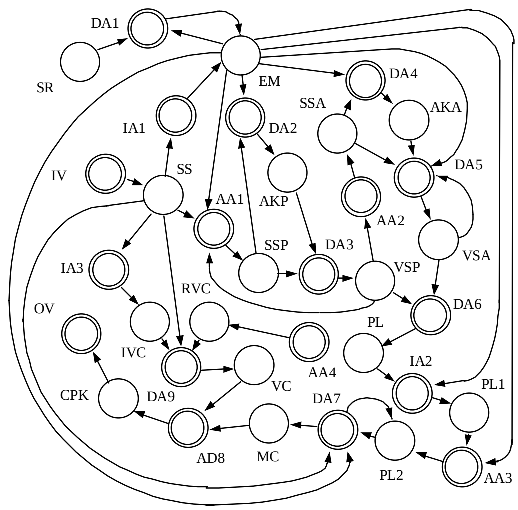

Figure 11 – An Intelligent System for the Control of an Autonomous Guided Vehicle

The detailed description of such system may be found in [7], and also in [5]. Inside it, the basic functions detected by Albus, i.e., Sensorial Perception (SP), World Modeling (WM), Value Judgement (VJ) and Behavior Generation (BG) are distributed among the generalized deductive, inductive and abductive arguments along the object network. In this sense, those argumentative knowledge units are the building blocks of intelligent systems. Based on its fundamental behavior, intelligent behavior is able to emerge, depending on the different types of knowledge being processed.

This example illustrates how our generalized reasoning operators are useful in building intelligent systems. It concerns the control of an autonomous guided vehicle, that have been implemented and tested within a simulation environment ([5], also in [7])

8.3 GFACS and argumentative knowledge

GFACS stands for "Grouping, Focusing Attention and Combinatorial Search". It consists by a set of procedures that acting together in a multi-resolutional cycle, was pointed out as the elementary unit of intelligence, by Meystel [2]. We would like to point out, in virtue of the previously exposed, that this procedure is able to be broken in more elementary units, i.e., the generalized deduction, induction and abduction procedures. For example, grouping is a kind of generalized induction, focusing attention is a generalized deduction, and combinatorial search is a mixture of generalized induction and abduction.

9. CONCLUSION

We presented in this paper, some concepts that we believe are critical for the borning field of computational semiotics. We touched on how semiotics concepts can be explored within the context of intelligent systems, proposing some changes in the current view of some known concepts

Despite putting this matter in a somewhat speculative form, the aim of this work is to open discussion about those issues, in order to prepare the terrain for the further construction of a future intelligent systems theory, enhanced and sustained by computational semiotics.

10. REFERENCES

[1] J.Albus, "Outline for a Theory of Intelligence", IEEE Transactions on SMC, vol. 21, n.3,May/June 1991.

[2] A.Meystel, "Semiotic Modeling and Situation Analysis : An Introduction", AdRem Inc., 1995.

[3] T.A. Sebeok, "The Evolution of Semiosis", Tutorials 1 – Semiotics - ISAS Conference, Gaithersburg, 1997.

[4] R.R.Gudwin and F.A.C.Gomide,"An Approach to Computational Semiotics"- Proceedings of the 1997 ISAS Conference, Gaithersburg, 1997, pp.467-470.

[5] R.R.Gudwin - "Contributions to the Mathematical Study of Intelligent Systems" - Ph.D. Thesis - DCA-FEEC-UNICAMP - May, 1996. (in portuguese)

[6] R.R.Gudwin and F.A.C. Gomide, "Computational Semiotics : An Approach for the Study of Intelligent Systems - Part I : Foundations" – Technical Report RT-DCA09 - DCA-FEEC-UNICAMP, 1997.

[7] R.R.Gudwin and F.A.C. Gomide, "Computational Semiotics : An Approach for the Study of Intelligent Systems - Part II : Theory and Applications" - Technical Report RT-DCA09 - DCA-FEEC-UNICAMP, 1997.

[8] L.Goldfarb and. S.Nigam, "The Unified Learning Paradigm : A Foundation for AI", in Artificial Intelligence and Neural Networks: Steps toward Principled Integration, eds. V.Honavar and L.Uhr, Academic Press, Boston, 1994.

[9] L.Goldfarb, "Inductive Theory of Semiotic Systems", Proc. of the ISAS'97 – International Conference on Intelligent Systems and Semiotics, Gaithersburg, MD, USA, Sept.1997, pp. 111-116.

[10] A.Ram and D. Leake, "Evaluation of Explanatory Hypotheses", Proc. of the 13th Annual Conference of the Cognitive Science Society, Chicago, IL, August, 1991, pp. 867-871.

[11] J.C.A. van der Lubbe and E.Egbers, "An Integrated Approach to Fuzzy Learning and Reasoning in Hierarchical Knowledge Structures", Proc. of the ISAS'97 – International Conference on Intelligent Systems and Semiotics, Gaithersburg, MD, USA, Sept. 1997, pp.37-43

[12] R.Gudwin and F.Gomide, "A Computational Semiotics Approach for Soft Computing", Proc. of the SMC'97, IEEE International Conference on Systems, Man and Cybernetics, Orlando, FL,USA, Oct. 1997, pp. 3981-3986