Learning, from a Logical Point of View

Pablo Noriega B.V.

Instituto Nacional de Estadística Geografía e Informática and Laboratorio Nacional de Informática Avanzada

México

Abstract

Learning is a pervasive topic in Artificial Intelligence (AI). It was already a well expressed concern in the very first works in the field and has since continued to be present in AI activity, sometimes as a subsidiary part of an area of research in AI, other times as a subfield on its own.

Although one can distinguish several schools of thought, most of these would agree that (artificial) learning is at least "any process by which a system improves its performance". This conception allows for many approaches; differences stemming from methodological, technical and teleological commitments.

In this paper, I will concentrate on the approach taken by logically inspired works. In this approach, the "improvement" that would account for learning is interpreted, either by extending or refining the class of sentences deducible in a formal theory, or by achieving better efficiency in the deduction process. I will present some characteristic devices used by logicians to obtain these improvements and will illustrate how purely formal devices are complemented by metalogical constructs to represent learning-theoretical tenants.

0 Introduction

Since the purpose of this symposium is to bring together researchers from distant fields to set the grounds for possible collaboration, I will present a candid view of how we, logicians who work in learning, do our work.

This logical approach is not uncommon in AI. Logic has been a rather popular testing ground for AI theories, mainly because of its formal simplicity and its purported connection with thought, but also because it tends to be a useful tool. Thus, AI textbooks abound with comments, topics, examples and notation that have a logical origin (Nilsson, 1980; Walker et al. 1987). Moreover, if one reviews the specific topic of learning in AI literature, one would find that a significant portion of this literature has been logically inspired[1].

But I have another reason for choosing a logical perspective, it is that formalistic approaches *seem* to be rigid and arbitrary when in fact they are not. When modeling reality, the logician is confronted with choices where intuition and pragmatic issues interplay with philosophical concerns and formal tools in a distinctive way. This interplay is not always explicit and tends to be further obscured by terse language, however, as I will attempt to show, it is not alien to that of other disciplines.

By choosing such a specific perspective and further focusing on a specific example, I expect to give a clearer picture of the kind of concerns and tools that are common to many works in this field in order to facilitate their assessment by scientists from a different background.

1 A Glimpse at Logic

Logicians are optimists who act as if reality, or at least any given part of it, were a theory.

Logical theories are, in essence, just a collection of sentences in a formal language, with a dynamic component usually known as "deduction". The simplicity and universality of these concepts are firmly grounded on a mathematical basis, endowing logicians with a powerful set of tools and a clear conscience.

The main task of a logician, when building a theory, is to devise (or select) an appropriate representation of reality and work on that representation which for all practical purposes has become his or her reality.

In this process, the logician is guided by principles that are mainly theory dependent. *Formal* principles, such as completeness or soundness; *aesthetic*

---

[1] See (Newell and Simon, 1956) or (Simon, 1991) for very early logical motivations and concerns about automated learning; (Cohen, et al., 1982), Chpt. XIV, for a classical AI overview; (Michalski et al., 1983) for seminal papers, and (Carbonell, 1990) and (Kodatroff and Michalski, 1990) for recent works and overviews.

principles of elegance, economy and simplicity; and, *pragmatic* principles like the quality and scope of the consequences of the theory, its intelligibility or -- when using a computer -- its efficiency. Usually, it is not until the *theory-dependent* criteria have been fully applied that *reality-dependent* concerns are brought into consideration, and although these may prove to be quite important for the logician, it is the formal and aesthetic criteria that tend to influence work in the strongest manner.

As with any other discipline, in Logic, the process of theory construction is essentially a process of designing metaphors that are suggestive and accurate. But logical metaphors are analytical, as other mathematical metaphors are, in the sense that familiarity makes them suggestive, and once a superficial accuracy is established, deeper accuracy is discovered through a systematic inspection of the metaphor. The difference between logical metaphors -- and the models these metaphors span -- with respect to other theory construction methodologies, is that objective formal properties can be tested and used to decide upon the quality of the model, and these take precedence over the empirical testing of the theory. This does not mean that subjective considerations are absent in the work of logicians, or that logicians are oblivious to the real world applicability of their models. It only means that logicians learn to trust their tools and instruments more than they trust their intuitions, and that their artificial formal world is neater than the real one.

The usual approach taken by logicians when they model reality consists of five basic steps:

First, they choose a problem; then they look for a logical convention to cast the problem into. Traditionally, the chosen problem tends to be a mathematical one -- such as the invariant properties of Vector Spaces and similar structures, or the kind of assertions uttered by topologists -- yet, on occasions, they work with a softer kind of phenomena such as learning or natural language.

These two steps are not always taken in the stated order. Logicians are "tool-makers" -- as well as tool users -- thus many develop or master a formal convention that they are eager to try out in whatever task comes in hand, or worse, they search for a task that fits the tool. But even when a problem precedes a convention, there is a Procrustean struggle to fit the problem into the confines of the convention: the problem is ultimately reduced to those aspects amenable to formal treatment. These choices, though, involve philosophical, aesthetic and practical considerations that may be more or less explicit.

Once a problem and a formal convention have been decided upon, the logician fiddles with the formal convention and constructs a system. This system, as I said before, consists basically of a collections of sentences and a deductive component, but some metalogical elements --and their implications -- are also considered and decided upon at this stage. For instance, the logician will have to decide how "syntactic" the system should be, or whether or not a "classical semantics" is appropriate. On many occasions, the problem is reduced even more at this stage and many times the formalism is refined in order to better accommodate the specific nuances of the problem. However, the kind of criteria that come into play here are mainly formal and aesthetic ones.

The fourth step consists in exploring the consequences of the system. This exploration is made through proving that certain formal properties do or do not hold in the system. Again formal, aesthetic and (now, at last) pragmatic criteria are employed to gauge the quality and interest of the system thus constructed.

Lastly, when the system is finally settled upon, the logician would normally subject it to a deeper analysis, aspiring to derive insights about the original problem from the structural components and the metalogical properties of the system.

This ideal route is followed more or less closely depending mainly on the kinds of metaphors and tools a logician likes best and, naturally, on the subjective intuitions the logician may hold about the problem.

The essential set of familiar metaphors the logician uses is small. There are some very basic concepts -- such as "truth" and "inference" -- that have been laboriously elaborated by logicians, and notions such as "proof", "consistency" and "completeness" that take peculiar forms and meanings in Logic. There is widespread agreement on their interpretation and applicability among logicians, but two markedly different groups of "logics", though, should be distinguished: classical and non-classical. Their choice will normally impose different *methodological*, *teleological* and even *ontological* commitments to the practicing logician, thus biasing the logician's attention and activity.[2]

Classical Logic holds two basic tenants and a few subsidiary ones. The first tenant holds that the only meaningful component of a statement is its truth value, and that any statement has a value which has to be either true or false. Second, Classical Logic assumes that the meaning of a complex statement can be derived from the meaning of its components. From these tenants a number of constructs are concocted and some major results become paradigmatic. Notably, the distinction between *syntax* and *semantics*, stratification of *language* and *metalanguage* and the elusive goal of *completeness* of systems. Classical Logic has proved to be quite successful for modeling mathematical concepts, but it tends to be inappropriate for less artificial domains of application.

Non-classical Logics vulnerate some or many of those tenants, constructs and results in order to accommodate atypical assumptions, or validate alternative intuitions. Activity in non-classical logics has been expanding, and logicians are becoming more and more dextrous in their use. This is due, on the one hand, to the fact that non-mathematical applications of Logic are becoming more frequent and, on the other, to the use of computers as modeling grounds for logicians.

But classical Logic has a well established reputation among logicians, and the amount and quality of its established results far surpass those of non-classical logics. Thus, when classical logic can be applied to a problem, the stock of techniques and results that are available is generally rather large. Furthermore, our understanding of its resources and limitations is such that it is either straight-forwardly applied, or dismissed in favor of a non-classical logic without much deliberation.

---

[2]See (Haack, 1978) for a systematic and readable discussion on the philosophical issues associated with basic logical notions, as well as classical and non-classical distinctions and commitments. (Martin, 1983) addresses complementary issues.

2 Logical Learning. An Example

From a logical point of view, the crude notion of learning as "any process through which a system improves its performance" (Simon, 1986) is adequate[3]. Since logicians model reality through theories with a deductive component, logically inspired works in learning can be conveniently classified according to the way they deal with the notion of "improvement" by either focusing on the theory, or on the deductive component. Thus, one can divide logically inspired works in two large groups: those which improve the efficiency of the deduction process, and those which -- by dealing with the class of sentences that belong to a specific theory -- refine or enrich the original set of sentences or their consequences.

The first group of works, those that improve the deduction process, is generally done with classical tools. Some are centered around "pruning techniques" that shorten search paths of a deduction tree, while others attempt to model the kind of knowledge a person uses to search through a tree, by incorporating "hints" or similar heuristics, and reinforce successful strategies by preferring certain deduction paths or choice criteria. "Learning" is achieved when the deduction tree or its associated searching algorithms are automatically modified, and more efficient or more "natural" deductions replace the original ones. (Mitchell et al., 1983; Mostow, 1990; Wang and Bledsoe, 1987),

In the second group there is one shared convention: One starts with a given set of sentences, T, and after some process one ends up with a different set, T'. T and T' may be different for a variety of reasons, but there are at least three well distinguishable strategies to achieve or acknowledge these changes.

First there are "inductive" strategies, where the essential difference between T and T' is in the generality of their respective sentences. In its typical form, induction starts with a collection of examples and, through algorithmic manipulation, these examples are condensed into a (generally) much reduced set of sentences from which all the examples can be deduced (for example, the "general diagnostic rules"

---

[3]Other, more elaborated, definitions of "machine learning" or "automated learning" have been proposed by the AI community, see for example (Cohen et al., 1982) pp. 324 & ss.; (Carbonell, 1990), pp 1-9; (Anderson, 1990) or (Michalski and Kodatroff, 1990).

are induced from a database containing simple clinical reports) (Quinlan, 1983; Michalski and Chilausky, 1980; Goldberg, 1989).

Second, there are works that relax the truth-conditions of the theories, by either qualifying the degree to which any given sentence is thought to belong to the theory (e.g. through "credibility factors"), or by ontologically distinguishing between the commitments to the sentences (e.g. by distinguishing among "beliefs" and "truths"). As new evidence is acquired (through experience or any other interaction) the truth-conditions of the sentences are changed (thus, T and T' may have the same sentences but with different credibilities associated to them in T and T', or sentences that used to be beliefs in T, become truths in T' and vice-versa) (Nilsson, 1986; Reiter, 1980; Wilks and Bien, 1984).

Finally, there are those where the emphasis lies in the metalogical mechanisms that account for deduction, so that the class of sentences that belong to an improved theory T' are those that may be obtained from the original theory T through those (non-classical) metalogical processes (e.g. "circumscription" or "dialogues") (Lenat, 1983; McCarthy, 1980; Emde et al., 1983; Bowen and Kowalski, 1982; Hayes-Roth, 1983; Perlis, 1985).

As my own wording of these classifications should have suggested, these strategies are not absolutely independent and many specific projects take advantage of more than one strategy.

In the following paragraphs I will present an example that belongs to the last kind of strategy. In order to facilitate its presentation I will stick to the five step outline of the previous section.[4]

A. Selection of the problem

When one builds an "Expert System" (or more properly, a "knowledge-based"

[4] This example was developed by Santiago Negrete, Edgar González and myself at IBM Mexico Scientific Center. It is fully documented in (Negrete, 1990).

system), one is essentially constructing a logical Theory, i.e. a set of sentences in a formal language, and a deductive mechanism.

There are various formal conventions for expressing these sentences, a very common one is through "production rules", that is, conditional sentences of the form:

IF C1, C2, ..., Cn, *THEN* P

where, the Ci's are "conditions" that have to be simultaneously satisfied in order to trigger a process P. These conditions and processes are usually of the form "object, attribute, value" (i.e., atomic formulae in a first order predicate language). A collection of these sentences constitute a (rule-based) knowledge base. A knowledge base plus an automated deduction system (or "inference mechanism") is called a "knowledge-based system", (KBS for short). These inference mechanisms can also vary but most are simple computational incarnations of classical rules of inference.

These concepts have become so standard that people have constructed computational packages that contain a formal representational convention and an inference mechanism, together with other devices that facilitate the processes of management and exploitation of the knowledge base. Most of these packages claim to be general enough to be used on all sorts of empirical knowledge, they are called Expert-system "shells". Thus, when a specialist of a given field of expertise wants to build a KBS, he or she may use one of these shells and simply load it with the rules that translate his or her expertise.

But the process of loading a knowledge base can be cumbersome and prone to error. Two common errors gravitate around this process: one is that the knowledge is "incomplete" (i.e., that the theory doesn't take care of all possible cases), the other is that it is "inconsistent". A knowledge base is said to be inconsistent when one can deduce contradictory statements from it (for example, an inconsistent knowledge base that deals with credit card applications, would accept *and* reject one given card application).

Both kinds of error are undesirable from a practical point of view. The problem of incompleteness is to a certain degree unavoidable, but some pragmatics have been developed to avoid its worse manifestations. The problem of inconsistency has deserved a more formal treatment (Bowen and Kowalski, 1982). Some shells provide tools for checking if the knowledge base is inconsistent (Nguyen et al., 1987). Most of these check for simple inconsistencies, but checking for absolute consistency can be an undecidable problem. Even for common knowledge bases it is computationally very expensive and therefore rarely provided (Walker et al, 1987). Furthermore, when an inconsistency is detected it seldom is automatically corrected.

There are two stages at which these problems appear. On the one hand, the expert rarely has a complete and consistent theory to start with. He or she usually has a fairly approximate idea of the theory and some paradigmatic cases but, while creating the knowledge base, many intuitions become explicit and new insights on the problem are gained. It is far from unusual that, in the process of translating these insights and intuitions into rules, many subtleties are overlooked, thus errors creep into the knowledge base. On the other hand, even when an expert has built a robust knowledge base, reality tends to prove it incorrect. As the knowledge base is used, experience brings to attention new cases that show that either the knowledge base was not as good as it was thought to be or that the underlying problem changed. Consequently the knowledge base has to be updated, but in the process of manually updating it, errors are easily introduced.

That was the problem we chose to tackle: the evolution of knowledge bases. In particular, the evolution that is forced upon the knowledge base when inconsistent knowledge is introduced. We decided to construct a set of programs that validated the consistency of a knowledge base -- as it was being built or used -- and, when an inconsistency was detected, the system would revise the knowledge base and automatically propose changes that would resolve the inconsistency. Thus, our system would "learn" by changing the set of sentences that belonged to the original theory.

In order to get a handle on a problem that was practical and concrete enough, we decided to restrict the domain of knowledge bases to those that could be expressed through production rules, in the classical object-attribute-value convention, and where deduction is formalized as classical inference. We decided to use a real world working system, so we chose IBM's Shell, ESE ("Expert System Environment").

B. Choice of Logical Convention

The choice, in reality, went from tool to problem. We had been working for some time with formal dialogues; the evolution of knowledge bases was a natural testing ground for some of the ideas, constructs and programs we had explored and developed so far (González and Noriega, 1985; Noriega and González, 1985; Noriega and González, 1987; Negrete et al., 1988).

Formal dialogues are descendants of Greek and Medieval "dialectics"; in consequence, their basic intuitions have undergone a long process of depuration. However, it wasn't until relatively recent times that they were studied under the light of formal logic (Lorenzen, 1967; Stegmüller, 1974; Hamblin, 1977; Rescher, 1977).

A (formal) dialogue -- or dialogical system -- is a sort of game that involves two participants who exchange (formal) locutions according to a strategy (or set of rules) that depends on their role in the game. Each participant holds a set of "beliefs" that may change during the dialogue. One participant, the "proponent", believes at least one statement -- the "thesis" -- to be true; the other participant, the "opponent" tries to prove it wrong. The game usually consists of defending the thesis against the arguments of the opponent, but once a thesis has been upheld or refuted (and even during the process of attacking and defending it) the beliefs of the participants may change. Why and how these beliefs change is what made dialogical systems relevant to the problem of automatic learning.

Our main interest was to develop a formal theory of dialogues (González and Noriega, 1985) and after having constructed computational models of formal dialogues (Negrete et al., 1988) we wanted to explore more interesting applications in AI.

We had shown (Noriega and González, 1985) that formal dialogues were reducible to classical (monological) logic if and only if the set of beliefs is fixed and known. So, if in a given situation beliefs change, and that is the case of knowledge bases, dialogues could be used to explore that situation, presumably with better results than plain monological logics.

The problem of evolution of knowledge bases seemed like a good candidate to elaborate and test our intuitions for three more reasons:

The *dialogical structure* is immediate -- the knowledge base could be thought of as the *proponent* and "reality" (or the expert that was updating the KB) as the *opponent*.

The evolutionary process of knowledge bases includes the two stages of "full" dialogues. A stage where the thesis confronts an *antithesis* (the test for contradictions), and the stage where a *synthesis* is created (the correction of the knowledge base)

The *language* in which the dialogue would take place was propositional logic, as simple as it gets.

In addition, we had speculated on the relationships between dialogues and learning for the case of data bases (Noriega and González, 1986). The case of knowledge bases was a better example to explore this concern.

Finally, it was advantageous to our research programme to exhibit a real-world application.

C. Construction of the System

At this stage we could state the problem in almost precise terms:

There is an asymmetrical dialogical system D, where the proponent is a consistent knowledge base B, in the language of ESE, and the opponent, O, holds a sentence S, not in B. O wins iff B u {S} is shown to be inconsistent by an effective procedure P. There is an effective procedure Q such that If O wins, then Q produces a consistent knowledge base B', in ESE, such that S is in B' and most of the consequences of B are also consequences of B'.

In order to make the statement precise, we have to build the effective procedures *P* and *Q*.

Since we wanted to have a true working system, we had to build two main modules on top of IBM's shell, one to detect contradictions, *P*, another one to correct them, *Q*. The first one was rather straight-forward; for the second module we speculated on different approaches and chose the simplest effective one.

In order to build the contradiction-detection module, we needed a theorem prover. We had already constructed an automatic theorem prover, based on dialogues, that worked for finite theories in Classical and Intuitionistic Propositional Logic (Negrete et al., 1988), so we extended our theorem prover to the language of ESE, and then built a simple recursive process that would obtain minimal inconsistent subsets of the knowledge base, using the improved theorem prover to find a contradiction (Negrete, 1989). This involved some work: Like most production rule languages, ESE's language is reducible to a propositional language except for the fact that atomic formulae have variables. If we wanted to use the propositional theorem prover we had, we would then have to turn these variables into constants. There are two ways of doing this, one is to instantiate the variables so they become constants for demonstration purposes; the other way is to wait for the variables to be instantiated by a user during a consultation. The first one is convenient when the number and kinds of variables is somewhat restricted (e.g. when no functions are present and relations range over finite or finitizable sets). That turned out to be quite frequent and easy to detect, but when these restrictions are not present, combinatorial explosion and undecidability make the problem unmanageable. Waiting for consultations to instantiate variables, is not elegant, but it proved to be a convenient procedure to detect contradictions in those uncomfortable real applications where automatic instantiation is impractical.

To deal with theory correction, we based our speculations on philosophical debates on how scientific theories evolve (Lakatos and Musgrave, 1970). From this very general approach, we distilled three computational strategies. One, following Popper's ideas, was to revise the old theory globally in order to produce a new one that preserved most of the previous conclusions -- and the added statement that produced the inconsistency -- but avoided the contradiction. A second one, based on Lakatos' ideas, would attempt to solve the contradiction by locally changing sentences of the original theory. The third one, based on Suppes' proposals, would doubt the validity of new information. From a computational point of view, the most convenient strategy is the second one, because it carries a stronger syntactic content. That is the one we decided to implement.

Our system would then take a knowledge base and check if it was inconsistent. If so, it would find a minimal inconsistent set and then identify and propose changes that would correct the inconsistency. These changes are always the addition of an atomic formula, or a negated atomic formula, to a sentence in the inconsistent set. For a given inconsistent set of sentences, there could be more than one possible correction, all of them were automatically generated and any one of them would correct the inconsistency. The choice among equally effective corrections is semantic, so we left it to the human expert.

D. Exploring the System

We first established some *formal* properties of the system we had constructed. For example, we found that for a large class of knowledge bases -- which we were able to characterize -- the "inconsistency" module was sound and correct, in the sense that it found inconsistencies if and when they were present, but its computational complexity was exponential. We also proved that the "correction" procedure was effective, in the sense that for any given minimally inconsistent set it produced a new consistent set.

These formal properties have immediate *aesthetic* implications for logicians. For instance, the fact that the inconsistency module is exponential is not nice, nor is the fact that it is not absolutely general, although the characterization of its range of applicability is an acceptable consolation. The effectiveness of the correction module is very nice.

Besides these formal-dependent assessments, other more subjective aesthetic judgements were also made. The problem proved to be "fruitful", the basic dialogical metaphor was "suggestive" and "appropriate", the instantiation of variables through consultation was "not elegant" but "clever", and so on.

Finally, we did some evaluation from a *pragmatic* point of view. The system was found to be "slow" in general, "practical" for small knowledge bases, "very useful" for real world applications, etc.

While performing this evaluation, we realized that there were improvements that we could introduce to make the system more useful. For instance, on many occasions the corrections that were automatically generated to resolve an inconsistency were not unique and that the choice, being semantic, was left to the user. In practice, many knowledge-based systems do a simple two-way classification (e.g. "grant" or "deny" credit). When an inconsistency is to be resolved in this case, an important semantic distinction is whether the corrected knowledge base should be more or less "restrictive" (i.e. prefer denial over acceptance in the credit example). So, we introduced a question that allowed the user to indicate his or her preference with respect to the alternative atomic formulae (grant or deny), then the system would only propose those corrections that were consistent with the stated preference. The trick, trivial as it is, proved to be, rhetorically, very impressive.

E. Further Insights

We succeeded in our pragmatic task to build a self-improving KBS. We succeeded, also, in showing that formal dialogues were powerful modeling tools, and we developed and proved new formal notation and properties. In this process we also learned about "learning".

From a practical point of view we showed that it is convenient and feasible to extended the basic architecture of KBS's to include an automatic learning module: Real world knowledge bases are seldom static objects, they evolve as new experience is gained or when their authors realize that the knowledge they have expressed is not sufficient. This dynamic process tends to introduce contradictory knowledge into the base and this, for most applications, is not desirable. Thus, to have a truth-maintenance system linked to a shell is a desirable and convenient tool. This is, nowadays, a frequent feature in sophisticated shells. What is not so frequent, however, is to have a learning mechanism attached to the truth-maintenance system. We showed that simple learning mechanisms can easily be constructed and constitute a practical and usable feature.

We were more pleased by our theoretical cogitations. Any dialogue has two essential "moments", one where a contradiction is established, another where the contradiction is resolved. This is an obvious property of dialogues which we were able to exploit in constructing our system. What may not be so obvious is that these two moments can be thought of as two interdependent cognitive operations that are present, in various guises, in many rational processes. For some processes, such as a legal debate, the assessment of a contradiction takes prevalence. In others, negotiation for example, subsequent corrective actions are the more interesting. Yet in some others, such as some forms of learning, both components seem to be equally important. The processes and cognitive strategies involved in finding a contradiction are different from the ones used to correct it. And different are also the specific strategies for dealing with contradictions and their resolution in these various dialogical settings. What we were able to get a glimpse of in our project was that no matter how different the specifics may be, these two processes have characteristic structural and dynamic components that may be formally elucidated. Therefore it is worth paying deeper attention to them.

As far as learning is concerned, we would not claim that our system went beyond the crudest forms of learning. In essence, our system was able, only, to assess its beliefs and change them when they were contradictory. But we realized that what had been a formal requirement -- the separation of the two moments of detection and resolution of contradictions -- evoked, in a rather natural and immediate way, two frequent concerns in the field: "anagnorisis" (or a reflective process that models conscience) and "creativity" (or a dynamic process that is triggered by the reflective process). The fact that we used very simple formal devices to deal with them, does not betray their significance, I think that on the contrary, it made explicit their radical difference and also their connection. Furthermore, the success of these simple devices suggests a likely course for additional analysis and more sophisticated formal constructions.

3 Closing Remarks

Formalistic approaches in AI have been criticized by some AI practitioners and also by external observers. They have been caricatured as looking for a lost coin under the lamppost that is lit, not where the coin was dropped. I, naturally, do not hold such a pessimistic view. While recognizing the intrinsic limitations of formalisms to depict reality, I prefer to see these tools, not as a static lamppost but as an illuminating "flash light".

When we deal with a problem we may trust our tools for guidance but we also hold intuitions and, eventually, confront reality. The process is far from being rigid, as I have tried to show, yet it is peculiar. Is it *sound*?

That question is difficult to answer. I think it makes sense only if we answer it from a *pragmatic* axiology, and in that perspective the fact that a model works is perhaps as far as I would go. I can not claim that a model *truly* corresponds to reality. But I like the fact that a model allows me to interpret reality in a way that was not available before it was built, and I like the fact that a model allows me to interact with reality in a better way than I did before having it. And in that sense I claim that the method is, at least *fruitful*.

Some philosophers of Logic go further. Platonists claim that when we construct a beautiful system we *discover* a reality that was waiting for a lucky or virtuous logician to stumble upon it. Formalists would claim that it is only a *game* we play with symbols, and if one is clever and tenacious, one eventually builds a nice system.

I think that even the staunchest Platonist, or the most raging of Formalists among logicians is guided, in the process of building a theory, by some sort of conditioning that is biological in a very fundamental way. It may be that we only "discover" an immanent reality or, in contrast, simply "build" a purely artificial one. But in its everyday manifestations the two processes are indistinguishable. And they are indistinguishable because, I believe, logical constructs are fruitful or are deemed beautiful for no arbitrary reasons: they obey the same fundamental principles of life. The paradigmatic results in Logic that tend to guide our work are paradigmatic precisely because they express the *natural* qualities of regularity, symmetry, simplicity,... Our flashlight illuminates with its stark -- and narrow -- light these qualities.

How, then, can neuroscientists profit from the logicians work and vice versa? I find it unlikely that any superficial connections can be established between the work of neuroscientists and logicians. We may both claim to be interested in a topic, say, learning, however the accepted meaning, the approaches and the foci will probably be very different. But in the process of exploring the topic, in the process of expressing the results, we may both reach a fortunate metaphor, an insight, a construct that can be shared.

4 References

Anderson, J.R. (1982): A theory of the origin of human knowledge. In *Machine Learning, Paradigms and Methods*, Carbonell, J. (ed). pp 313-351

Bowen, K. A. and Kowalski, R.A. (1982): Amalgamating language and metalanguage in logic programming. In *Logic Programming*, Clark, K.L. and Tarnlund, S.A. (eds.) Academic Press, NY, pp. 153-172

Carbonell, J. (1990): *Machine Learning, Paradigms and Methods*. MIT Press, Cambridge, MA.

Clark, K. L. and Tarnlund, S. A. (1982): *Logic Programming*. Academic Press, NY.

Cohen, P. and Feigenbaum, E. (1982): *Handbook of Artificial Intelligence*, Vol. 3. HeurisTech Press & Morgan Kaufmann, Palo Alto, CA.

Elithorn, A. and Banerji, R. (1984): *Artificial and human Intelligence*. Elsevier, Amsterdam.

Emde, W., Habel, C. U. and Rollinger, C.R. (1983): The discovery of the equator, or concept driven learning. *Proceedings of the International Joint-Conference on Artificial Intelligence, Karlsruhe, W. Germany*. William Kaufmann, Los Altos, CA. pp. 455-458

Goldberg, D. (1989): *Genetic Algorithms*. Addison Wesley, Reading, MA.

González, E. and Noriega, P. (1985): Los sistemas dialógicos y algunas de sus aplicaciones. In *Memorias de la II Reunión de Trabajo en Inteligencia Artificial*, pp. 55-57. Fundación Arturo Rosenblueth, MX.

González, E., Negrete, S. and Noriega, P. (1989): Un diseño para bases de conocimientos evolutivas. In *Memorias de la V Reunión Nacional de Inteligencia Artificial*, pp. 221-231. LIMUSA, MX.

Haack, S. (1978): *Philosophy of Logics*. Cambridge University Press, Cambridge.

Hamblin, C.L. (1970): *Fallacies*, Methuen, London.

Hayes-Roth, F. (1983): Using proofs and refutations to learn from experience. In *Machine Learning, an Artificial Intelligence Approach*, Michalski, R.S. et al. (eds.) pp. 221-240.

Kodratoff, Y. and Michalski, R. (1990): *Machine Learning*, Volume III, Morgan Kaufmann, San Mateo, CA.

Lakatos, I. and Musgrave, A. (1970): *Criticism and the growth of knowledge*. Cambridge University Press, Cambridge.

Lenat, D. B. (1983): The role of heuristics in learning by discovery: three case studies. In *Machine Learning: An Artificial Intelligence Approach*, Michalski, R.S., Mitchell, T.M. and Carbonell, J. (eds.) pp. 243-306.

Lorenzen, P. (1967): *Normative Logic and Ethics*, Bibliographisches Institut, Zurich.

Martin, R.M. (1983): *Mind, Modality, Meaning, and Method.* SUNY, Albany, NY.

McCarthy, J. (1980): Circumscription --a form of non-monotonic reasoning. *Artificial Intelligence*, Vol 13, pp. 27-39.

Michalski, R.S. and Chilausky, R.L. (1980): Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybeans disease diagnosis. *Policy Analysis and Information Systems*, Vol. 4, No. 2.

Michalski, R.S. and Kodatroff, Y. (1990): Research in Machine Learning: recent progress, classification of methods and future directions. In *Machine Learning*, *Volume III*, Kodratoff, Y. and Michalski, R. (eds.) pp. 3-30.

Michalski, R.S., Mitchell, T.M. and Carbonell, J (1983): *Machine Learning: An Artificial Intelligence Approach*. Tioga, Palo Alto, CA.

Mitchell, T.M., Utgoff, P. and Banerji, R. (1983): Learning by experimentation: acquiring and refining problem-solving heuristics. In *Machine Learning: An Artificial Intelligence Approach*, Michalski, R.S., Mitchell, T.M. and Carbonell, J. (eds.). pp. 163-190.

Mostow, J. (1990): Design by derivational analogy: issues in the automatic replay of design plans. In *Machine Learning, Paradigms and Methods*, Carbonell, J. (ed.), pp. 119-184

Newell, A. (1990): *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.

Newell, A. and Simon, H. (1956): The Logic Theory Machine. *IRE Transactions on Information Theory* 2, pp. 61-79.

Negrete, S. (1989): INTERESE: Un detector de inconsistencias para el entorno ESE (Expert System Environment). In *Memorias de la VI Reunión Nacional de Inteligencia Artificial*, pp. 453-461, LIMUSA, MX.

Negrete, S., (1990): *Sistemas Dialógicos y sus Aplicaciones a Bases de Conocimiento*. UNAM, Facultad de Ciencias, MX.

Negrete, S., Noriega, P. and González, E. (1988): DIALOREN. Un simulador de diálogos formales. In *IBERAMIA-88*, *Memorias del Primer Congreso Iberoamericano de Inteligencia Artificial*, pp. 59-67. Barcelona.

Negrete, S., González, E. and Noriega, P. (1989): Sistemas Dialógicos y Demostración Automática de Teoremas. In *Memorias de la V Reunión Nacional de Inteligencia Artificial*, pp. 277-285, LIMUSA, MX.

Nilsson, N.J. (1980): *Principles of Artificial Intelligence*, Tioga, Palo Alto, CA.

Nilsson, N.J. (1986): Probabilistic Logic. *Artificial Intelligence* 28, pp. 71-87.

Nguyen, T.A., Perkins, W., Laffey, T., Pecora, D. (1987): Knowledge Base verification. *AI Magazine* (Summer), pp. 69-75

Noriega, P. and González, E. (1986): En torno de un modelo dialógico de base de datos. *Comunidad Informática* 26, pp. 34-40.

Noriega, P. and González, E. (1985): Retórica , Dialéctica y Cambio de Creencias. Un nuevo punto de vista sobre la teoría del debate. *Estudios* 5, pp 49-59.

Perlis, D. (1985): Languages with self-reference I: Foundations (Or: We can have everything in First-order logic) *Artificial Intelligence* 25, pp. 301-322

Quinlan, J.R. (1983): Learning efficient classification procedures and their application to chess and games. In *Machine Learning*, Michalski, R.S., Mitchell, T.M. and Carbonell, J. (eds). pp. 463-482.

Reiter, R. (1980): A Logic for default reasoning. *Artificial Intelligence* 13, pp. 81-132

Rescher, N. (1977): *Dialectics*, SUNY, Albany, NY.

Simon, H. (1991): *Models of my Life*. Basic Books, NY.

Simon, H. (1986): Why should machines learn. In *Machine Learning*, Michalski, R.S., Mitchell, T.M. and Carbonell, J. (eds). pp. 25-37.

Stegmüller, W. (1974): Remarks on the completeness of logical systems relative to the validity-concepts of P. Lorenzen and K. Lorenz. *Notre Dame Jrnal of Formal Logic*, Vol. V, No. 2.

Walker, A., McCord, M., Sowa, J. and Wilson, W. (1987): *Knowledge Systems and Prolog*. Addison-Wesley, Reading, MA.

Wang, T. and Bledsoe, W. (1987): Hierarchical Deduction. *Journal of Automated Reasoning* 3, pp. 35-77.

Wilks, Y. and Bien, J. (1984): Beliefs, points of view and multiple environments. In *Artificial and Human Intelligence*, Elithorn, A. and Banerji, R. (eds.), pp. 147-171