Logique & Analyse **157** (1997), 83-96

SOME RECENT APPLICATIONS OF PARACONSISTENT SYSTEMS TO AI

Jair Minoro ABE

Abstract
In this paper we present some applications of paraconsistent logics that are being established powerfully over the past years. This work does not intend to be complete, nor go to in technical details, restricting and focusing primarily some developments made recently.

1. *Introduction*

A deductive theory is said to be *consistent* if it has no theorem, one of which is the negation of the other; otherwise it is called *inconsistent* (or *contradictory*). A theory is called *trivial* if all formulas (or sentences) of its language are provable; otherwise it is called *non-trivial*.

Analogously, the same definition above applies to propositional systems, sets of information, etc. (taking into account their set of consequences).

If the underlying logic of a theory T is classical logic or most of the extent logics, T is trivial iff it is inconsistent. Therefore, if we want to handle logically inconsistent but non-trivial theories or information systems, we have to use a new kind of logic.

Paraconsistent logic is a logic that can be the basis of inconsistent but non-trivial theories.

This way, paraconsistent logic is of fundamental importance for handling of inconsistent but non-trivial theories or information systems.

Paraconsistent logic has found various applications in Artificial Intelligence (AI), logic programming, etc. and showing itself to be of basic significance for Computer Science in general.

In this paper we summarize of some significative applications obtained recently to Computer Science (ParaLog —a paraconsistent logic programming language), multi-agents systems, knowledge representation (Frames), new framework for Computer Science based on paraconsistent annotated systems, and implementation of paraconsistent electronic circuits.

2. *A Paraconsistent Logic Programming Language — ParaLog*

Inconsistency is a natural phenomenon arising from the description of the real world. This phenomenon may be found in several contexts. Nevertheless, human beings are capable of reasoning adequately. The automation of such reasoning requires the development of formal theories.

The employment of logic systems allowing reasoning about inconsistent information is an area of growing importance in Computer Science, Data Base Theory and AI. For instance, if a knowledge engineer is designing a knowledge base *KB*, related to a domain *D*, he may consult *n* experts in that domain. For each expert e_i, $1 \leq i \leq n$, of domain *D*, he will obtain some information and will present it in some logic such as a set of sentences KB_i, for $1 \leq i \leq n$. A simple way of combining the knowledge amassed from all experts in a single knowledge base *KB* is:

$$KB = \bigcup_{i=1}^{n} KB_i$$

However, certain KB_i and KB_j bases may contain *conflicting* propositions —p and $\neg p$. In such case, p might be a logic consequence of KB_i, while $\neg p$ might be a logic consequence of KB_j. Therefore, *KB* is inconsistent and consequently meaningless, because of the lack of models. However, the knowledge base *KB* is not a useless set of information.

There are some arguments favoring this standpoint, as follows:

- certain subsets of *KB* may be inconsistent and express significant information. Such information cannot be disregarded;

- the disagreement among specialists in a given domain may be significant. For instance, if physician M_1 concludes patient *X* suffers from a fatal cancer, while physician M_2 concludes that same patient suffers from cancer, but a benign one, the patient will probably want to know the causes of such disagreement. This disagreement is significant be cause it may lead patient X to take appropriate decisions —for instance, to get the opinion of a third physician.

The reasoning for the last item is that it is not always advisable to find ways to exclude formulas identified as causing inconsistency(ies) in *KB*, because many times important information may be removed. In such cases, the very existence of inconsistency is important.

Though inconsistency is an increasingly common phenomenon in programming environments —especially in those possessing a certain degree of distribution— it cannot be handled, at least directly, by classical logic, on which most of the current logic programming languages are based. Thus, one has to resort to alternatives to classical logic; it is therefore necessary to search for programming languages based on such new logics.

Paraconsistent Logic, despite having been initially developed from the purely theoretical standpoint, found in recent years extremely fertile applications in Computer Science, thus solving the problem of justifying such logic systems from the practical standpoint.

In [22], [23], [25], and [35] it was proposed a variation of the logic programming language Prolog —ParaLog— which allows inconsistency to be handled directly. This implementation was made independently of results by Subrahmanian and colleagues [28], [55].

3. *A Paraconsistent Multi-Modal System*

Multi-agents systems are an important topic in AI. The use of modal systems for modeling knowledge and belief has been largely considered in Artificial Intelligence. For instance, it seems that the first one to consider knowledge and belief to machines was McCarthy [49]. Subsequently, Rosenschein [53], Parikh and Ramamujam [50], Rosenschein and Kaelbling [54], Fischer and Immerman [43], Halpern and Fagin [44], Halpern and Moses [45], among others, have considered knowledge in multi-agent systems, besides other approaches.

The essential ideas underlying the systems proposed by Halpern and Fagin [44], Halpern and Moses [45], and collaborators can be summarized as follows: $\Box_i A$ can be read *agent i knows A*, $i = 1, \ldots, n$. *Common knowledge* and *Distributed knowledge* are also defined in terms of additional modal operators: $\Box_G$ ("everyone in the group G knows"), $\Box_G^C$ ("it is common knowledge among agents in G"), and $\Box_G^D$ ("it is distributed knowledge among agents in G") for every nonempty subset G of $\{1, \ldots, n\}$.

Nevertheless, the most of those proposals use extensions of classical logic or at least part of it, keeping as much as fundamental characteristics of classical logic. When it is taken questions of logical omniscience, one relevant concept that appears is that of contradiction. Some authors have taken into account this problem, for instance, Cresswell [29], [30], [31]. Other authors have showed how different properties of knowledge can be captured by imposing certain conditions on semantics, which permit such contradictions (see Wansing [58], Lipman [46], [47], and [48]).

The attractiveness of admitting paraconsistency and paracompleteness in the system becomes evident if we observe that some agents can actually lie or be ignorant about certain propositions: an agent may state both A and $\neg A$ (the negation of A) hold (or that none of A and $\neg A$ hold).

In [12], [16] it was presented a class of paraconsistent and, in general, paracomplete and non-alethic multimodal systems $J\tau$ which may constitute, for instance, a framework for modeling paraconsistent knowledge (also see [13]).

4. *A Multi-Agent Paraconsistent Framework*

In [51], [52] it was described a specification and prototype of an annotated paraconsistent logic-based architecture, which integrates various computing systems —planners, databases, vision systems, etc.— of a manufacture cell. Throughout this paragraph, such systems will be referred to as agents.

In application domains such as robot control and flexible manufacture cells, the complexity of the control task grows proportionally with the increase and variety of stimuli coming from the external world to the system.

To deal with such complexity and to adequate to those stimuli within the time constrains imposed by the application domain, the control task should not be centralized. However, control decentralization is not easy to implement: paradoxically, it can lead to an increase in the time required to solve the problem, since it can interfere with the coherence of the resolution process. To avoid this phenomenon the architecture specifies:

a) How each agent is going to use its knowledge, plans, goals and skills in the resolution process.
b) How each agent is going to behave when faced with imprecise and inconsistent information.

c) How, and when, each agent is going to pass on to the other agents its plans, goals, skills and beliefs.
d) How each agent is going to internally represent the information received from the other agents and its belief in this information.

Finally, the proposed architecture is able to "encapsulate" the existing computing systems, as well as hide from these systems the mechanisms of co-operation, co-ordination and inconsistency handling. This reduces the effort necessary to integrate the systems.

Gathering concepts and techniques from Distributed Artificial Intelligence and Annotated Paraconsistent Logic, the architecture has also enabled the agents to work in a co-operative fashion, even in the presence of inconsistent data and results, in order to achieve common or distinct interactive goals.

In Distributed Artificial Intelligence Systems the agents are members of a network, and each of them only possesses its own local perception of the problem to be solved. In a traditional Distributed Processing approach an intense message exchange among the nodes of the network is necessary, so as to supply the nodes with the information necessary to the processing and local control of each node. The result of this intense communication is a drop in the performance of the entire system, and a high level of synchronism in the agents' processing.

One possible manner to reduce the communication and synchronization rates among agents is to let them produce partial (candidate), incomplete, or incorrect results. Or, even, inconsistent and/or paracomplete results in comparison with the partial results produced by other agents.

This kind of processing requires a resolution problem architecture, which allows the co-operation among agents, in such a way that the partial results of each agent can be revised and enhanced from the information obtained during the interaction with the other agents.

For the past two decades, some Distributed Artificial Intelligence architectures have been proposed in the most varied fields, ranging from signal integration to industrial applications. However, such frameworks do not deal with the inconsistency phenomenon. In the majority of them, only the most recent data are considered during the resolution process. The earlier data (regardless their origin), which may lead to inconsistency, are not taken into account. Despite its importance, the inconsistency phenomenon is a research field in Distributed Artificial Intelligence, which has not received enough attention.

One possible reason for the current situation is that the inconsistency phenomenon and/or paracompleteness can not be directly dealt with through the Classic Logic. Therefore, in order to tackle inconsistencies and paracompleteness directly, one should employ a logic other than the classic one. In this work, we have employed the Annotated Paraconsistent logic to deal with the systems' inconsistencies.

In order to make possible the use of such logic in complex application domains (intense information input and critical agent response time), like the manufacture cells, it has been necessary to extend and refine the techniques and concepts of the Paraconsistent Logic Programming, Evidential Logic and the Amalgama's Knowledge-base.

5. *Paraconsistent Frame System*

In Computer Science, a good solution for a given problem many times depends on a good representation. For most Artificial Intelligence applications, the choice of a knowledge representation is even more difficult, since the criteria for such choice are less clear.

Though no general consensus exists of what is knowledge representation, many schemes were proposed to represent and store knowledge. Many of such schemes have been successfully used as a foundation for the implementation of some existing systems. There are, however, several characteristics of knowledge that are not yet well understood, such as defaults and inconsistencies. Until a better comprehension of such characteristics is achieved, the representation of knowledge will remain as an active field of study.

There are several schemes to represent knowledge. The two schemes that better capture the knowledge concerning objects and their properties are semantic networks and frames.

The first of these schemes to represent knowledge, semantic networks, were originally proposed by psychology researchers, as modeling systems for the human associative memory. Later, several Computer Science researchers extended the original concept of semantic networks to facilitate the handling of more complex objects and relationships. Basically, a semantic network is a graph in which the nodes show objects, or a class, and the links show a relationship, generally binary, between objects or classes connected by the link. The nodes may be of two types: individual and generic. The first represents descriptions or affirmations concerning an individual instance of an object, while the second is related to a class or category of objects. The classes are pre-ordered in a taxonomy, and there are links representing special binary relationships such as *isa* —is a— and *ako* —a kind of. The first link type connects an individual node to a generic node and identifies an individual as belonging to a certain class. The second links two generic nodes between them and shows that a given class is a subclass of another.

The second of these knowledge representation schemes —frames— became popular in the 70s due to the appearance of the frame theory. The frame theory appeared initially as a result of an paper written by Marvin Minsky. A frame system as proposed by Minsky consists in a collection of frames articulated in a semantic network. At the time, the use of frames was recommended as basic to understand visual perception, dialogues in natural language and other complex behavior. The development of languages for frame handling was partly intended for the implementation of frame-based Artificial Intelligence systems.

Semantic network-based systems and frame-based systems may be considered similar as to their structure, but they differ in the knowledge they represent. That is, while a semantic-network-based system represents simple objects, a frame-based system may represent complex objects.

Nevertheless, there is, a gap between the knowledge represented by the frame-based system and the knowledge of the real world. Among the few frame systems are concerning with bridging that knowledge gap stands the system proposed by Sandri & Bittencourt, using possibilistic frames. This system, however, similarly to the others, does not handle adequately issues such as exceptions and the inconsistency phenomenon.

A frame is a representation of a complex object. It is identified by a *name* and consists of a set of *slots*. Each frame possesses at least one hierarchically superior frame, thus providing the basis of the inheritance mechanism. A special frame is the root of this inheritance hierarchy.

The inheritance hierarchy is a consequence of the classic notion of taxonomic hierarchy as a way to organize knowledge. The taxonomic hierarchy is just the beginning of inheritance reasoning. Researchers in Artificial

Intelligence have added tools to represent class properties, exceptions to inherited properties, multiple superclasses and *structured concepts* with specific relations over the structural elements. Furthermore, the reasoning by inheritance naturally leads to simple default reasoning and nonmonotonic reasoning, and may be used to reason about prototypes and typical instances of inheritance system classes.

The two main types of existing inheritance systems are: those that do not admit exceptions to inherited properties and those that admit exceptions to inherited properties. It is easy to describe the semantics of the first type of inheritance in first order logic, in which frames may be interpreted as unary predicates and slots may be interpreted and binary predicates. The description of the semantics of the second type of inheritance system in first order logic is much more difficult, since exceptions introduce nonmonotonicity.

Since the late 70s several nonmonotonic formalisms have been proposed. Among the most widely published are: Clark's *predicate completion*, Reiter's *default logic*, McDermott and Doyle's *nonmonotonic logic I*, McCarty's *circumscription*, McDermott's *nonmonotonic logic II*, and Moore's *autoepistemic logic*. However, none of these formalisms deal adequately with issues like the inconsistency phenomenon.

Despite this phenomenon being increasingly common in programming environments —mainly in those possessing a certain degree of distribution— it cannot be treated, at least directly, by the Classic Logic, in which most of the current logic programming languages are based.

Thus, so as to be able to study these inconsistencies directly, one has to resort to alternative logic, it being therefore necessary to look for programming languages based on such logic.

We have described a variation of the Prolog logic programming language based on the Annotated Logic $Q\tau$ allowing to deal directly with the inconsistency. The proposed logic programming language is called Paraconsistent Prolog —*ParaLog*.

To implement frame systems dealing with the inconsistency, the difficulty caused by the lack of a formal semantics both for paraconsistent frame systems and for inheritance reasoners dealing with inconsistencies and multiple inheritance frame systems must be taken into account.

In [20], [24], [26], [27] it is presented the main features of a paraconsistent inheritance reasoner allowing to deal properly with exceptions and inconsistencies in multiple inheritance frame systems. The paraconsistent inheritance reasoner represents knowledge by means of paraconsistent frames and infers based on the inconsistency/under-determinedness degree. This reasoner, being a wide-encompassing one, also allows less complex inheritances to take place.

Furthermore, its main feature is not to eliminate contractions, *ab initio*.

6. *Paraconsistent Logics and Nonmonotonic Reasoning*

There are various intelligent systems including nonmonotonic reasoning in the field of Artificial Intelligence. Each system has different semantics. More than two nonmonotonic reasoning maybe required in complex intelligent systems. It is more desirable to have a common semantics for such nonmonotonic reasoning. We propose the common semantics for the nonmonotonic reasoning by annotated logics and annotated logic programs.

7. *Paraconsistent Electronic Circuits*

In [36], [40] it was proposed digital circuits (logical gates COMPLEMENT, AND, OR) inspired in a class of paraconsistent annotated logics $P\tau$. These circuits allow "inconsistent" signals in a nontrivial manner in their structure.

Such circuits consist of six states; due the existence of literal operators to each of them, the underlying logic is functionally complete; it is a many-valued and paraconsistent (at least "semantically") logic.

The simulations were made at 50 MHz, 1.2 μm, by using the software *AIM-SPICE*, version 1.5a. Also, it was presented a paraconsistent analyzer module (PAM) [38], [41] combining several paraconsistent circuits, as well as a circuit that allows to detect inconsistent signals and gives a non-trivial treatment.

As far as we know, these results seem to be pioneering in using the concept of paraconsistency in the theory of electronic circuits. The applications appear to be large in horizon: it expands the scope of applications where conflicting signals are common, such as in sensor circuits in robotics, industry automation circuits, race signal control in electronic circuits, and many other fields.

8. *Conclusions*

As it can be seen by the previous exposition, the applications of paraconsistent systems have been very fruitful in many aspects. In fact, nowadays it has converted in one most interesting research area in Informatics and much rich applications are to be done.

Acknowledgment: FAPESP Grant 97/02328-9

Department of Informatics, ICET – Paulista University
Rua Dr. Bacelar, 1212
04026–002 São Paulo – SP – Brazil

Institute For Advanced Studies, University of São Paulo
Av. Prof. Luciano Gualberto, Travessa J, 374, Térreo, Cidade Universitária,
05508--900 São Paulo – SP – Brazil
E-mail: jmabe@lsi.usp.br

REFERENCES

[1] ABE, J.M., "Fundamentos da Lógica Anotada" ("Foundations of Annotated Logics") (in Portuguese), Ph.D. Thesis, FFLCH/USP, São Paulo, Brazil, 1992.

[2] ABE, J.M., "Annotated Logics $Q\tau$ and Ultraproducts", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 16, 20 p., 1994.

[3] ABE, J.M., "On Annotated Modal Logics", *Mathematica Japonica*, 40(3), 1994, 553–560.

[4] ABE, J.M., "Annotated Logics, Reduced Direct Products, Ultraproducts and Ultrapowers", presented in Workshop on Logic, Language, Information and Computation, Wollic 94, Recife, Brazil, 1994.

[5] ABE, J.M., "Curry Algebras N_1", *Atti Acc. Lincei Rend. Fis.*, s.9, vol. 7, 1996, 125–128.

[6] ABE, J.M., "A logical system for reasoning with inconsistency", 5th Annual Meeting of SBPN'97, Ciência e Cultura na Globalização – Novos Paradigmas, Aug. 08–10, 1997, Águas de Lindóia, SP, Brazil, 196–201, 1997.

[7] ABE, J.M., "Lógica Paraconsistente e Inteligência Artificial", Coleção Cadernos de Estudos e Pesquisas – UNIP, Série: Estudos e Pesquisas, n° 1–004/97, Universidade Paulista, 28p, 1997.

[8] ABE, J.M., "A note on Curry Algebras $P\tau$", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 25, 7 p., 1997.

[9] ABE, J.M., "Filters and ideals of a Pτ-Algebra", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 26, 7 p., 1997.

[10] ABE, J.M., "Curry Algebras Pτ", *The First World Congress On Paraconsistency*, Universiteit Gent, Belgium, 1997, submitted for publication.

[11] ABE, J.M. & S. AKAMA, "Incorporating time in paraconsistent reasoning", to appear, 1997.

[12] ABE, J.M., B.C. ÁVILA e J.P.A. PRADO, "Multi-agents and inconsistency", to appear in ICCIMA'98, *The International Conference on Computational Intelligence and Multimedia Applications*, Monash University, Feb. 9–11, Churchill, Victoria, Austrália, 1998.

[13] ABE, J.M., B.C. ÁVILA & J.P.A. PRADO, "On a class of paraconsistent multimodal systems", *The First World Congress On Paraconsistency*, Universiteit Gent, Belgium, 1997, submitted for publication.

[14] ABE, J.M. & J.I.S. FILHO, "Implementação de Circuitos eletrônicos de funções lógicas paraconsistentes Radix N", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 22, 37 p., 1996.

[15] ABE, J.M. & J.I.S. FILHO, "Inconsistency and Electronic Circuits", to appear in *International Symposium on Engineering of Intelligent Systems*, University of Laguna, Tenerife, Spain, 1998.

[16] ABE, J.M., J.P.A. PRADO & B.C. ÁVILA, "On a class of paraconsistent multimodal systems for reasoning", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 24, 12 p., 1997.

[17] ABE, J.M. & L. PUJATTI, "Paraconsistent Legal Knowledge Engineering", *The First World Congress On Paraconsistency*, Universiteit Gent, Belgium, 1997, submitted for publication.

[18] AKAMA, S. & J.M. ABE, "Many-valued and annotated modal logics", submitted in *International Symposium on Multiple-Valued Logic*, Japan, 1998.

[19] AKAMA, S. & J.M. ABE, "Sequent calculus for annotated logics", submitted in Tableaux'98, Netherlands, 1998.

[20] ÁVILA, B.C., "Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de Frames com Múltipla Herança", Ph.D. thesis, University of São Paulo, São Paulo, 1996.

[21] ÁVILA, B.C., J.M. ABE e J.P.A. PRADO, "Reasoning in Paraconsistent Frame Systems", to appear in *CSCDWID'97, The Second International Workshop on CSCW in Design*, Nov. 26–28, BangKok, Thailand, 1997.

[22] ÁVILA, B.C., J.M. ABE & J.P.A PRADO, "Uma extensão da linguagem Prolog para suportar programação lógica evidencial",

Coleção Documentos, Série Lógica e Teoria da Ciência, IEA–USP, n° 29, 43p., 1997.

[23] ÁVILA, B.C., J.M. ABE e J.P.A. PRADO, "ParaLog-e: A Paraconsistent Evidential Logic Programming Language", to appear in XVII *International Conference of the Chilean Computer Science Society*, IEEE Computer Society Press, Valparaíso, Chile, Nov., 1997.

[24] ÁVILA, B.C., J.M. ABE & J.P.A. PRADO, "A Paraconsistent Logic Programming Language", *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS'98)*, Vol.3, AI, Editor:E. Alpaydin, ICSC Academic Press International Computer Science Conventions Canada/Switzerland, ISBN 3-906454-12-6, 281-287, 1998.

[25] ÁVILA, B.C., J.M. ABE e J.P.A. PRADO, "ParaLog-e: A Paraconsistent Logic Programming Language (extended version)", to appear, 1997.

[26] ÁVILA, B.C., J.M. ABE & J.P.A PRADO, "Um sistema de frames utilizando programação lógica evidencial", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 27, 43p., 1997.

[27] ÁVILA, B.C., J.M. ABE & J.P.A PRADO, "A utilização de redes de herança em representação de conhecimento", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 28, 18p., 1997.

[28] BLAIR, H.A. & V.S. SUBRAHMANIAN, "Paraconsistent Foundations for Logic Programming", *Journal of Non-Classical Logic*, 5, 2, 45–73, 1988.

[29] CRESSWELL, M.J., "Classical intensional logics", *Theoria*, 36, 347–372, 1970.

[30] CRESSWELL, M.J., "Intensional logics and logical truth", *Journal of Philosophical Logic*, 1, 2–15, 1972.

[31] CRESSWELL, M.J., "Logics and Languages", London, Methuen and Co., 1973.

[32] DA COSTA, N.C.A., "Belief and Contradiction", *Crítica, Revista hispanoamericana de Filosofía*, Vol. XX, No. 60 (diciembre, 1988), 3–11.

[33] DA COSTA, N.C.A., V.S. SUBRAHMANIAN & C. VAGO, "The paraconsistent logics Pτ", *Zeitschr. f. Math. Logik und Grundlagen d. Math.*, 37: 139–148, 1991.

[34] DA COSTA, N.C.A., J.M. ABE, V.S. SUBRAHMANIAN, "Remarks on Annotated Logic", *Zeitschrift f. Math. Logik und Grundlagen d. Math.*, 37, pag. 561–570, 1991.

[35] DA COSTA, N.C.A., J.P.A. PRADO, J.M. ABE, B.C. ÁVILA & M. RILLO, "Paralog: Um Prolog Paraconsistente baseado em Lógica Anotada", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 18, 21 p., 1995.

[36] FILHO, J.I.S. & J.M. ABE, "Lógica paraconsistente anotada e circuitos de portas lógicas", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 23, 41p., 1997.

[37] FILHO, J.I.S., J.M. ABE & P.L. SANCHES, "Circuitos de portas lógicas primitivas fundamentados em lógica paraconsistente anotada", III WORKSHOP DE IBERSHIP, Feb. 19–21, 1997, Departamento de Ingeniería Elétrica, CINVESTAV – IPN, México D.F., México, 227–237, 1997.

[38] FILHO, J.I.S. & J.M. ABE, "Módulo analisador paraconsistente: uma proposta de circuito lógico paraconsistente", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 30, 18p., 1997.

[39] FILHO, J.I.S. & J.M. ABE, "PARASENSOR: um sensor paraconsistente projetado para uso em sistema eletrônico embasado em lógica paraconsistente anotada", *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA–USP, n° 31, 18p., 1997.

[40] FILHO, J.I.S. & J.M. ABE, "Electronic Circuits and Inconsistency", *The First World Congress On Paraconsistency*, Universiteit Gent, Belgium, 1997, submitted for publication.

[41] FILHO, J.I.S. & J.M. ABE, "Paraconsistent Analyzer Module", *The First World Congress On Paraconsistency*, Universiteit Gent, Belgium, 1997, submitted for publication.

[42] FILHO, J.I.S., J.M. ABE & P.L. SANCHES, "Circuitos de portas lógicas primitivas implementados a partir de uma classe de lógicas paraconsistentes anotadas", to appear in Boletim da Escola Politécnica da USP, 1997.

[43] FISHER, M.J. & N. INMERMAN, "Foundation of knowledge for distributed systems". In J.Y. Halpern (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. Fifth Conference*, pp. 171–186. San Francisco, Cal.: Morgan Kaufmann, 1986.

[44] HALPERN, J.Y. & R. FAGIN, "Modeling knowledge and action in distributed systems", *Distributed Computing* 3(4), 159–179, 1989.

[45] HALPERN, J.Y. & Y. MOSES, "Knowledge and common knowledge in a distributed environment", *Journal of the ACM* 37(3), 549–587, 1990.

[46] LIPMAN, B.L., "Decision theory with impossible possible worlds", Technical report Working paper, Queen's University, 1992.

[47] LIPMAN, B.L., "Logics for non-omniscience agents: An axiomatic approach". Technical report Working paper 874, Queen's University, 1992.

[48] LIPMAN, B.L., "An axiomatic approach to the logical omniscience problem". In R. Fagin (Ed.), *Theoretical Aspects of Reasoning about*

Knowledge: Proc. Fifth Conference, pp. 182–196. San Francisco, Cal.: Morgan Kaufmann, 1994.

[49] McCARTHY, J., "Ascribing mental qualities to machines". Technical report CRL 90/10, DEC–CRL, 1979.

[50] PARISH, R. & R. RAMAMUJAN, "Distributed processing and the logic of knowledge". In R. Parikh (Ed.), *Proc. Workshop on Logics of Programs*, pp. 256–268, 1985.

[51] PRADO, J.P.A., J.M. ABE & B.C. ÁVILA, "PARANET: a paraconsistent multi-agent system", *The First World Congress On Paraconsistency*, Universiteit Gent, Belgium, 1997, submitted for publication.

[52] PRADO, J.P.A. & J.M. ABE, "Um Planejador Baseado em Lógica Paraconsistente". *2o Simpósio Brasileiro de Automação Inteligente*, Curitiba, pp. 177–178, 1995.

[53] ROSENSCHEIN, S.J., "Formal theories of AI in knowledge and robotics". *New Generation Computing* 3, 345–357, 1985.

[54] ROSENSCHEIN, S.J. & L.P. KAELBLING, "The synthesis of digital machines with provable epistemic properties". In J.Y. Halpern (Ed.), *Theoretical Aspects of Reasoning about Knowledge: Proc. Fifth Conference*, pp. 83–97. San Francisco, Cal.: Morgan Kaufmann, 1986.

[55] SUBRAHMANIAN, V.S., "On the semantics of quantitative logic programs", Proc. 4th IEEE Symposium on Logic Programming, Computer Society Press, Washington D.C., 173–182, 1987.

[56] SYLVAN, R. & J.M. ABE, "On general annotated logics" (abstract), 1995.

[57] SYLVAN, R. & J.M. ABE, "On general annotated logics", with an introduction to full accounting logic, draft, 1996.

[58] WANSING, H., "A general possible words framework for reasoning about knowledge and belief", *Studia Logica* 49(4), 523–539, 1990.