



Jair Minoro Abe  
Leonardo Pujatti

## A META-INTERPRETER BASED ON PARACONSISTENT LEGAL KNOWLEDGE ENGINEERING

**Abstract.** The Legal Knowledge Engineering is a new topic of investigation of Artificial Intelligence. This paper discusses some relevant problems related to this new area in a summarized way. Within the Normative Law Theory, one question that arises naturally is that of contradiction, like for example: articles conflicting with other articles inside the same code, codes conflicting with codes, codes conflicting with jurisprudence, and in general, treatments with conflicting propositions in Normative Law Theory. This paper suggests to treat directly inconsistencies in the Legal Knowledge Engineering; this engineering has as underlying logic a paraconsistent annotated deontic logic. There are three main approximations in Legal Knowledge Engineering based on: cases, rules and logic. In this paper, we consider the approximation based on logic. It is considered a paraconsistent annotated deontic logic. Based on this logic, it is established a new proposal that is called Paraconsistent Legal Knowledge Engineering. For this proposal, it is suggested a meta-interpreter to support the deontic operators as well as inconsistency — entitled on this paper Paralog<sub>D</sub> that can be used as a base to handle with the issues discussed.

*Keywords:* paraconsistent annotated deontic logic, paraconsistent legal knowledge engineering.

### 1. Introduction

The juridical logic, is based, in general, on deontic logics. The use of this logic, applied to the techniques of Artificial Intelligence leads to an area of



study called Legal Knowledge Engineering. The isolated use of deontic logic, however, can be inadequate to treat antinomies and gaps in the Normative Law Theory. In these cases, a paraconsistent deontic logic can be useful, as a underlying logic to the inconsistent and paracomplete normative theories, allowing to deal with inconsistencies and paracompleteness in a non-trivial manner, helping in the resolution of antinomies and in the resolution of gaps, for instance. The non-alethic logic can be applied, as logic subjacent to the incomplete and inconsistent normative theories.

The Legal Knowledge Engineering, in a more general sense, can be seen as a sub-area of the Artificial Intelligence that deals with the functional ontology of Law, of the models, and juridical activities, of the paradigms of approximations in the legal knowledge area, of the epistemological aspects of the legislation, of formalization of the logics of legal knowledge and computer systems applied to the juridical area, such as: specialist systems based on knowledge and programming language modified to accept a juridical logic.

There are several modeling approximations in the Legal Knowledge Engineering. These approximations establish a layout to the solution of problems in real cases in the juridical area, as well as logical-mathematical formulation and computer implementation. [SERGOT 90], one of the pioneers in defining modeling in Legal Knowledge Engineering, divides this area in some approximations, based on: *activities* such as, litigation or layout of contracts; *formalism*, through logical adaptations applied to the juridical area in developed specialist systems, such as in Prolog language, reasoning, with a special attention to the analysis of real cases; *interpretation treatment*, that arranges the representation of law with the legislation normative outlines.

More recently, [VALENTE 95] defines three approximations of modeling in Legal Knowledge Engineering based on rules, cases and logics. The approximations based on *rules* treat the reasoning of legal knowledge as languages of logical programming. The approximations based on *cases* deal with the representation of laws with reasoning techniques based on cases. The approximations based on *logic* schematize the juridical laws through the application of deontic and non-monotonic logics.

The aim of this paper is to work with the approximate modeling based on logic, giving special relevance to the description of a paraconsistent annotated deontic logic applied to the Normative Law Theory and the adaptation of a paraconsistent annotated deontic Prolog, with an intelligent basis capable of doing an automatic verification of the juridical normative structures. Other aims are the discussion, modeling and adaptation of the area of Legal Knowledge Engineering to a paraconsistent annotated deontic logic, together



with computer applications and analysis of real cases, in this work entitled as Paraconsistent Legal Knowledge Engineering.

This work tries to establish a paraconsistent deontic logic, as an underlying logic to the Paraconsistent Legal Knowledge Engineering, as well as the conception and modeling of this area; encourage juridical discussions about the juridical ontology and epistemologies; implement a paraconsistent annotated deontic Prolog — entitled, in this work, `Paralog_D` — for the application in real cases of the juridical area, performing an analysis with the usual theories.

This work also intends to discuss the possible solutions of the paradoxes (antinomies) in the area of Paraconsistent Legal Knowledge Engineering, providing a valuable helping tool for the professionals of the area (Computing and Law).

## 2. Cases analysis and the `Paralog_D`

It was developed a Paraconsistent Annotated Deontic meta-interpreter to give support to some real cases of application of the Paraconsistent Legal Knowledge Engineering. This meta- interpreter is based on an implementation of a modified Prolog — `Paralog_D` — to accept the formalism and modeling of the underlying logic to Normative Law Theory, which is a paraconsistent annotated deontic logic. The functional ontology of law helps in the understanding of the analysis of real cases.

Based on the Paraconsistent Prolog — `Paralog` — developed by [ÁVILA, ABE & PRADO 97] and others and, on the Molog — developed by [CERRO 86, 95] — the meta-interpreters were modified to accept the operators of deontic logic, entitled `Paralog_D`. The `Paralog_D` manipulates concepts of inconsistency, impossible to treat in the standard Prolog; this language also establishes weights.

`Paralog_D` is a machine of general inference to the construction of a large class of meta- interpreters; this is adjusted to receive the deontic modal operators: Obligatory (entitled © in this terminology), Forbidden (®), Permitted (Ψ) and Indifferent (£).

The following example is extracted from a real case and shows the existence of antinomies among codes and judicial decisions related to an adoption. This case was extracted from the Brazilian jurisprudence.<sup>1</sup>

---

<sup>1</sup> The articles and all other juridic concepts involved in the example should be think as an adaptation.



A couple intends to adopt a baby, son of a single mother, retired in public house belonging to the State.

The magistrate, denied the request claiming that the destitution of the paternal-power, essential to the adoption, couldn't be promoted, once it would be necessary to make the quotation through an edict because the mother was in an unknown and doubtful place, what it was impossible, due to the nature of the process which requires justice secret.

The couple made a petition in a superior instance and then obtained the possibility of adoption of the kid.

The first judge based his decision on the following articles:

Article 5<sup>th</sup>, LV of the Federal Constitution:

To the litigants in judicial or administrative lawsuit, and to the accused in general are assured the contradictory and large defense [...]

Article 213 of the Civil Process Code:

The quotation is the act through which it is called to court the accused in order to defend himself

Article 231 of the Civil Process Code:

It will be done the quotation through an edict: ... II – when ignored, uncertain or inaccessible the place that he can be found,

The second judge based his decision on the following articles:

Article 5<sup>th</sup>, LX of the Federal Constitution:

The law will only be able to restrain the publication of the procedural acts for the defense of privacy, or in case of social interest requirement;

Article 155 of the Civil Process Code:

The procedural acts are public. However, the following lawsuits are done throughout justice secret:

I – lawsuits which are required by the public interest;

II – lawsuits related to marriage, filiation, separation, divorce, feeding and custody of underage.

The Judge, therefore, interpreted that in this case the procedural acts should be secret and allowed the adoption of the kid, not depending on the quotation of the natural mother.

In the deontic normative sentences we have:

$O \text{ Secret}(X) \wedge O \text{ unknown\_mother}(Y) \rightarrow V \text{ adoption}(X)$ ; and,

$O \text{ Secret}(X) \wedge O \text{ unknown\_mother}(Y) \rightarrow O \text{ adoption}(X)$ .



In the classic case, we have the inconsistency between the two sentences, that is to say,  $\text{V adoption}(X)$  and  $\text{O adoption}(X)$ .

When elaborating the new theory, putting weights  $[1.0, 0.0]$ , we have:

$$\text{O secret}(X)[1.0, 0.0] \wedge \text{O unknown\_mother}(Y)[1.0, 0.0] \rightarrow \text{V adoption}(X)[1.0, 0.0];$$

$$\text{O secret}(X)[1.0, 0.0] \wedge \text{O unknown\_mother}(Y)[1.0, 0.0] \rightarrow \text{O adoption}(X)[1.0, 0.0].$$

This new theory is still causing inconsistencies. Considering different points of view, we have:

In the first sentence: In the first paragraph of this sentence putting the  $\text{O secret}(X)[0.3, 0.8]$ , the Judge determines that the act of adoption should be secret with 30% of certainty and 80% of uncertainty, considering that the quotation of the mother should be through an edict. In the second paragraph of this sentence putting the  $\text{O unknown\_mother}(Y)[0.9, 0.2]$ , the Judge has the ingratitude of the mother with a degree of 90% of certainty and 20% of uncertainty, once the mother didn't show up until the day of the judgement. On the third paragraph of this sentence  $\text{V adoption}(X)[0.7, 0.3]$ , the judge forbids the adoption with 70% of certainty and 30% of uncertainty once he didn't take into consideration the quotation of the natural mother. Therefore, the first sentence would result:  $\text{O secret}(X)[0.3, 0.8] \wedge \text{O unknown\_mother}(Y)[0.9, 0.2] \rightarrow \text{V adoption}(X)[0.7, 0.3]$ .

In the second sentence: In the first paragraph of this sentence considering  $\text{O secret}(X)[0.8, 0.2]$ , the Judge determines that the act of adoption should be secret with 80% of certainty and 20% of uncertainty, once the quotation of the mother was unnecessary and the procedural acts would occur in secret. In the second paragraph of this sentence considering  $\text{O unknown\_mother}(Y)[0.9, 0.3]$ , the Judge has the gratitude of the mother with 90% of certainty and 30% of uncertainty, once the mother didn't show up until the day of the judgement. In the third paragraph of this sentence  $\text{O adoption}(X)[0.9, 0.2]$ , the Judge determines the adoption with 90% of certainty and 20% of uncertainty once the quotation of the natural mother through edict was unnecessary and the procedural acts continued normally. Therefore, the second sentence would result:  $\text{O secret}(X)[0.8, 0.2] \wedge \text{O unknown\_mother}(Y)[0.9, 0.3] \rightarrow \text{O adoption}(X)[0.9, 0.2]$ .



In `Paralog_D`, the example is implemented in the following way:

$$\textcircled{R}(\text{adoption}(X)) : [0.7, 0.3] \longleftarrow \textcircled{C} \text{Secret}(X) : [0.3, 0.8] \ \& \\ \textcircled{C} \text{unknown\_mother}(Y) : [0.9, 0.2];$$
$$\textcircled{C}(\text{adoption}(X)) : [0.9, 0.2] \longleftarrow \textcircled{C} \text{Secret}(X) : [0.8, 0.2] \ \& \\ \textcircled{C} \text{unknown\_mother}(Y) : [0.9, 0.3].$$

### 3. Conclusions

The use of the paraconsistent non-procedural programming language — `Paralog_D` — enlarges the perspectives of new researches in the area of the Legal Knowledge Engineering such as construction of juridical specialist systems, a more friendly relation between men and the machines for the inexperienced users and modification of other programming languages (C, C++, LISP, for example). These tools do not have the intention of substituting the professional of the juridical area, but to work as a great helper in the hard task of judging a lawsuit or giving a sentence.

However, in the level that the tool `Paralog_D` is now, it is necessary the presence of Knowledge Engineer to interact permanently with a professional of the juridical area, once the computing manipulation requires the constant interaction with the professional of the area of Artificial Intelligence. In conclusion, it has been studied the development of a specialist system that will be able to decrease the dependence of the professional of the juridical area, in relation to this expert of Knowledge Engineering.

### References

- [ABE 92] Abe, J. M.: *Fundamentos da lógica anotada*, (Foundations of annotated logics), in Portuguese, Ph.D. thesis, University of Sao Paulo, Sao Paulo, 135 p, 1992.
- [ABE 94] Abe, J. M. *On annotated modal logic*. Math. Japon. 40 (1994), no. 3, 553–560.
- [ALCHOURRÓN & BULYGIN 81] Alchourrón, C., Bulygin, E.: *The expressive conception of norms*. In R. Hilpinen, editor, *New Studies in Deontic Logic*, pages 123–148. D. Reidel. 1981.



- [ÁVILA, ABE & PRADO 97] Ávila, B. C., J. M. Abe & J. P. A. Prado, ParaLog-e: A Paraconsistent Evidential Logic Programming Language, *XVII International Conference of the Chilean Computer Science Society, IEEE Computer Society Press*, pp. 2–8, Valparaíso, Chile, Novembro, 1997.
- [CERRO 86] Cerro, L., F. Molog: *A system that extends Prolog with modal logic*. The New Generation Computer Journal 4. pp. 35–30, 1986.
- [CERRO 95] Cerro, L., F. Molog: *A tool for non-classical logic programming*. Internet. IRIT. 1995.
- [DA COSTA, ABE & SUBRAHMANIAN 91b] Da Costa, N. C. A., J. M. Abe, and V. S. Subrahmanian, 1991, Remarks on annotated logic, *Zeitschr. f. Math. Logik und Grundlagen d. Math.* 37: 561–570.
- [KELSEN 91] Kelsen, H: *General Theory of Norms*. Clarendon Press, Oxford.
- [LEMMON 62] Lemmon, E. J.: *Moral dilemmas*. Philosophical Review 71: 139–158, 1962.
- [LEMMON 65] Lemmon, E. J.: *Deontic logic and the logic of imperatives*. Logique et Analyse 6: 39–61, 1965.
- [MACCARTHY 89] Maccarthy, T.: *A language for legal discourse I. Basic structures*. In Proc. of the second international conference on AI and Law, pages 180–189, Vancouver, ACM, 1989.
- [RAZ 72] Raz, J.: *Legal principles and the limits of law*. The Yale Law Journal 81: 823–854.
- [RESCHER 68] Rescher, N.: *Topics in philosophical logic*. D. Reidel Pub. Co., Dordrecht, Holland, 1968.
- [SCHEREIBER 93] Schreiber, G., et al.: *A principal approach to knowledge-based system development*. Academic Press, The Netherlands, 1993.
- [SERGOT 90] Sergot, M.: *The representation of law in computer programs: a survey and comparison of past and current projects*. In T. Bench-Capon, editor, Knowledge Based Systems and Legal Applications. Academic Press, 1990.
- [SUBRAHMANIAN 87] Subrahmanian, V. S.: *On the semantics of quantitative logic programs*. Proc. 4th IEEE Symposium on Logic Programming, Computer Society Press, Washington DC, pp. 173–182, 1987.
- [VALENTE 95] Valente, A.: *Legal Knowledge Engineering. A Modeling Approach*. IOS PRESS. The Netherlands, 1995.
- [VON WRIGHT 56] Von Wright, G. H.: *A note on deontic logic and derived obligation*. Mind, 65: 507–509, 1956.



JAIR MINORO ABE  
Dept. of Informatics  
ICET – Paulista University  
R. Dr. Bacelar, 1212  
04026-002 Sao Paulo – SP, Brazil  
and  
Istitute for Advanced Studies  
University of Sao Paulo  
Av. Prof. Luciano Gualberto  
Travessa J, 374, térreo  
Cidade Universitária  
05508-900 Sao Paulo, Brazil  
[jairabe@uol.com.br](mailto:jairabe@uol.com.br)

LEONARDO PUJATTI  
Universidade Paulista  
ICET – UNIP  
R. Parnamirim, 50 – apto 172  
Sao Paulo – SP, Brazil  
[datat@virtual-net.com.br](mailto:datat@virtual-net.com.br)