Logique & Analyse **160** (1997), 335–343

ANNOTATED LOGICS $Q\mathcal{T}$ AND ULTRAPRODUCTS

Jair Minoro ABE and Seiki AKAMA

Abstract
In this paper, we study annotated predicate logic $Q\mathcal{T}$ based on the ultraproduct method. We can generalize some model-theoretic results including Lós' theorem in a paraconsistent setting.

1. *Introduction*

Annotated logics are a class of non-classical logics with significant applications to the foundations of logic programming. They are quite good underlying logics to deal both with incomplete and inconsistent information. Adequate motivations and relevance for the study of these logics are described in [1, 2, 5, 13]. Their foundational aspects were addressed by several authors: N. C. A. da Costa, V. S. Subrahmanian, H. Blair, C. Vago, M. Kifer, E. L. Lozinskii, J. M. Abe, and among others (see [1, 2, 5]).

In [8], an axiomatization of annotated logics at a predicate level was proposed. A detailed investigation was done by Abe in [1, 2], including axiomatization, semantics, and completeness theorem of these logics.

In this work, we study annotated logics based on the well known method of constructing models by the notion of ultraproduct. Throughout this paper, usual conventions and notions of set theory are assumed without extensive comments.

2. *Paraconsistent, Paracomplete and Non-Alethic Logics*

In what follows, we sketch the non-classical logics discussed in this paper, establishing some conventions, definitions and historical aspects.

Let T be a theory whose underlying logic is L. T is called *inconsistent* when it has the theorems of the form A and $\neg A$ (the negation of A). If T is not inconsistent, it is called *consistent*. T is called *trivial* if all formulas of the language of T are also theorems of T. Otherwise, T is said to be *non-trivial*. Note that in trivial theories the extensions of the concepts of formulas and theorems coincide.

A *paraconsistent logic* is a logic that can be used as the basis for inconsistent but non-trivial theories. A theory is called *paraconsistent* if its underlying logic is a paraconsistent logic. Issues such as those described above have been appreciated by many logicians. In 1910, the Russian logician Nikolaj A. Vasil'ev (1880–1940) (Vasil'ev's *imaginary logic*) and the Polish logician Jan Lukasiewicz (1878–1956) (Lukasiewicz's *three-valued logic*) independently glimpsed the possibility of developing such logics. Nevertheless, the Polish logician Stanislaw Jáskowski (1906–1965) developed a paraconsistent logic at a propositional level in 1948. His system is known as *discussive propositional calculus*. Some years later, the Brazilian logician Newton C. A. da Costa (1929–) constructed for the time hierarchies of paraconsistent propositional calculi C_n, $1 \leq n \leq \omega$, of paraconsistent first-order predicate calculi (with and without equality), of paraconsistent description calculi, and paraconsistent higher-order logics (systems NF_n, $1 \leq n \leq \omega$). A survey of paraconsistent logics is to be found in [3].

Another significant class of non-classical logics are *paracomplete logics*. A logical system is called *paracomplete* if it can function as the underlying logic of theories in which there are formulas such that these formulas and their negations are simultaneously false. Intuitionistic logic and several systems of many-valued logics are paracomplete in this sense.

As a consequence, paraconsistent theories do not satisfy the principle of non-contradiction, which can be stated as follows: of two contradictory propositions, i.e. one of which is the negation of the other, one must be false. And paracomplete theories do not satsify the principle of the excluded middle, formulated in the following form: of two contradictory propositions, one must be true. Finally, logics which are simultaneously paraconsistent and paracomplete are called *non-alethic logic*.

Various kinds of real problems need these non-classical logics. For instance, the paradox of set theory, the semantic antinomies, some issues originating in dialetics, and in the theory of constructivity have been discussed in the literature. However, one most amazing applications was obtained in the recent years in Computability Theory and Artificial Intelligence: most applications of these non-classical logics in computer science are related to situations where inconsistencies and overcompleteness arise naturally. Most often, this occurs in deductive database, logic programs, and other formalisms for representing data, knowledge and beliefs (see, for instance, [5, 6, 7, 11, 12, 14]).

3. *Annotated Predicate Logics* Qτ

In this section, we define the annotated piredicate logics $Q\tau$. The symbol $\tau = \langle \mid \tau \mid, \leq, \sim \rangle$ indicates some fixed finite lattice called *lattice of truth-values*. We use the symbol $\leq$ to denote the ordering in which τ is a complete lattice, $\perp$ and $\top$ to denote, respectively, the bottom element and top element of τ. Also, $\bigwedge$ and $\bigvee$ indicate, respectively, the greatest lower bound and the least upper bound operators with respect to subsets of $\mid \tau \mid$. We also fix an operator $\sim: \mid \tau \mid \rightarrow \mid \tau \mid$ which will work as the "meaning" of the negation of the system $Q\tau$.

The language L_τ (which is abbreviated by L) of $Q\tau$ is a first-order language (with equality) whose primitive symbols are the following:

1. Individual variables: a denumerably infinite set of variable symbols.
2. Logical connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction) and $\rightarrow$ (implication).
3. For each n, zero or more n-ary function symbols (n is a natural number).
4. For each n, zero or more n-ary predicate symbols (n is a natural number).
5. Quantifiers: $\forall$ (for all) and $\exists$ (there exists).
6. Equalty symbol: =
7. Annotated constants: each member of τ is called an *annotated constant*.
8. Auxiliary symbols: (,).

In the sequel, we suppose that $Q\tau$ posseses at least one predicate symbol. We define the notion of *term* as usual. Given a predicate symbol P of arity n, an annotational constant μ and n terms $t_1, t_2, \ldots, t_n$, an annotated term is an expression of the form $P_\mu(t_1,\ldots,t_n)$. In addition, if t_1 and t_2 are terms, $t_1 = t_2$ is an atomic formula. We introduce the general concept of *formula* in the standard way. An annotated atom $P_\mu(t_1,\ldots,t_n)$ can be read "it is believed that $P(t_1,\ldots,t_n)$'s truth-value is at least μ".

In general, we follow standard terminology and the notation as in Mendelson [13] with the obvious adaptations. We will employ them without extensive comments.

Definition 3.1
Let A and B be formulas of L. We put

$(A \leftrightarrow B) =_{def} (A \rightarrow B) \& (B \rightarrow A)$
$\neg_* A =_{def} A \rightarrow ((A \rightarrow A) \& \neg(A \rightarrow A))$.

The symbol "$\leftrightarrow$" is called the *biconditional* and "$\neg_*$" is called *strong negation.*

Let A be a formula. Then, $\neg^0 A$ is A, $\neg^1 A$ is $\neg A$, and $\neg^k A$ is $\neg(\neg^{k-1} A)$, where $k \geq 0$ is the natural number. The convention is also used for $\sim$.

Definition 3.2
Let $P(t_1,...,t_n)$ be an annotated atom. Then, a formula of the form

$$\neg^k P_\mu(t_1,...,t_n)$$

is called a *hyper-literal.* A formula which is not a hyper-literal is called a *complex formula.*

We now introduce the concept of interpretation for L. An *interpretation* M consists of a non-empty set D, called the *domain* of the interpretation, and an assignment to each predicate symbol P an n-ary function $P^M: D^n \rightarrow \tau$ and to each function symbol f of an n-place operation $f^M: D^n \rightarrow D$.

The concept of satisfiability is introduced as in Mendelson [13]. Let us denote by Σ the set of all denumerable sequence of D. Then, we can give an definition of *satisfaction* as follows:

1. if $P_\mu(t_1,...,t_n)$ is an annotated atom, then the sequence $s = (s_1,...,s_n,...)$ of elements of D satisfies $P_\mu(t_1,...,t_n)$ iff $P^M(s^*(t_1),...,s^*(t_n)) \geq \mu$, that is to say my belief degree of the proposition $P(t_1,...,t_n)$ is less or equal than the given interpretation of that proposition. Here, $s^*(t_i)$ denotes the denotation of a term t_i with respect to s.
2. s satisfies an atomic formula $a = b$ iff $P^M(s^*(a)) = P^M(s^*(b))$.
3. s satisfies a hyper-literal $\neg^k P_\mu(t_1,...,t_n)$ iff s satisfies $\neg^{k-1} P_{\sim\mu}(t_1,...,t_n)$.
4. s satisfies $A \rightarrow B$ iff either s does not satisfies A or s satisfies B.
5. s satisfies $A \& B$ iff s satisfies both A and B.
6. s satisfies $A \vee B$ iff s satisfies A or s satisfies B.
7. s satisfies $\forall x_i B$ iff every sequence s that differs from in at most the i-th component satisfies B.
8. s satisfies $\exists x_i B$ iff there is a sequence s' that differs from in at most the i-th component such that s' satisfies B.

Theorem 3.3
A sequence s satisfies a hyper-literal $\neg^k P_\mu(t_1,...,t_n)$ iff s satisfies $P_{\sim^k\mu}(t_1,...,t_n)$.

Proof: By induction on k, taking into account 3. of the above definition.

Definition 3.4
A formula A is true for the interpretation M (written $\models_M A$) if every sequence in Σ satisfies A. An interpretation M is said to be a *model* for a set of formulas if every formula in it is true for M. A formula A is said to be *logically valid* if A is true for every interpretaion (that is to say, A is true in all "possible" worlds).

Definition 3.5
1. We say that an interpretation M is *non-trivial* if there is a closed annotated atom such that A is false for M.
2. An interpretation M is said to be *inconsistent* if there is a closed annotated atom A such that $\models_M A$ and $\models_M \neg A$.

So, an interpretation is inconsistent iff there is some closed annotated atom such that it and its negation are both true for M.

Given a closed annotated atom A and a family $(M_j)_{j \in J}$ of interpretations for L, the set $I(A, (M_j)_{j \in J}) = \{j \in J \mid \models_{M_j} A$ and $\models_{M_j} \neg A\}$ is called the set of the *possible inconsistent models* for A with respect to the family $(M_j)_{j \in J}$.

The set $T(A, (M_j)_{j \in J}) = \{j \in J \mid \not\models_{M_j} A\}$ is called the set of *possible worlds not trivializable* by A.

Definition 3.6
An interpretation M is called *paracomplete* if there exists a closed annotated atom A such that $\not\models_M A$ and $\not\models_M \neg A$, that is to say A and $\neg A$ are both false for M.

Given a closed annotated atom A and a family $(M_j)_{j \in J}$ of interpretations for L, the set $P(A, (M_j)_{j \in J}) = \{j \in J \mid \not\models_{M_j} A$ and $\not\models_{M_j} \neg A\}$ is called the set of the *possible paracomplete worlds* for A (with respect to the family $(M_j)_{j \in J}$).

Theorem 3.7
$Q\tau$ is paraconsistent iff $\#\tau \geq 2$. Here, $\#\tau$ denotes the cardinarity of τ

Proof: Suppose that $\#\tau \geq 2$ and that there is at least one predicate symbol p. Let $|M|$ be a non-empty set satisfying $\#|M| \geq 2$. We then define $p_M : |M|^n \to |M|$ setting $p_M(a_1,..., a_n) = \bot$ and $p_M(b_1,...,b_n) = \top$, where $(a_1,...,a_n) \neq (b_1,...,b_n)$. Then, $\models_M p_\bot(i_1,...,i_n)$, where i_j is the name of b_j, $j = 1,..., n$, and $\models_M \neg p_\bot(i_1,...,i_n)$. Likewise, $\not\models p_\top(j_1,...,j_n)$, where j_i is the name of a_i, $i = 1,..., n$. So, $Q\tau$ is paraconsistent. The converse is immediate.

Theorem 3.8
For all τ there are systems $Q\tau$ that are paracomplete and also systems that are not paracomplete. If $Q\tau$ is paracomplete, then $\#\tau \geq 2$.
Proof: Similar to the proof of the preceding theorem.

Definition 3.9
An interpretation M is called *non-alethic* if M is both paraconsistent and paracomplete. The system $Q\tau$ is said to be *non-alethic* if there is an interpretation M for $Q\tau$ such that M is non-alethic.

Given an interpretation M, we can define the theory $Th(M)$ associated with M to be the set $Th(M) = Cn(\Gamma)$ where Γ is the set of all annotated atoms which are true for M; $Cn(\Gamma)$ indicates the set of all semantic consequences of elements of Γ. Then, we obtain the following theorem by the definition of $Th(M)$.

Theorem 3.10
Given an interpretation M for L, we have:

1. $Th(M)$ is a paraconsistent theory iff M is a paraconsistent interpretation.
2. $Th(M)$ is a paracomplete theory iff M is a paracomplete interpretation.
3. $Th(M)$ is a non-alethic theory iff M is a non-alethic interpretation.

In view of the preceding theorem, $Q\tau$ is, in general, a paraconsistent, paracomplete and non-alethic logic.

4. *Reduced Direct Products, Ultraproducts and Ultrapowers*

In this section, we show a well known way of constructing models based on the *ultraproduct method* (cf. Bell and Sloman [4]). A *filter* on a non-empty set J is a set D of subsets of J satisfying the following conditions: (1) $\varnothing \notin D$, $J \in D$; (2) If $X, Y \in D$, then $X \cap Y \in D$; (2) If $X \in D$ and $X \subset Y \subset J$, then $Y \in D$. Let $(M_j)_{j \in J}$ be a family of (normal) models for L and F be a filter on J. For each $j \in J$, let D_j denotes the domain of the model M_j. $\Pi_{j \in J} D_j$ indicates the Cartesian product of the family $(D_j)_{j \in J}$. We construct the model $\Pi_{j \in J} M_j / F$, known as *reduced direct product*. When F is an ultrafilter, $\Pi_{j \in J} M_j / F$ is called an *ultraproduct*. When F is an ultrafilter and all the M_j's are the same model N, then $\Pi_{j \in J} M_j / F$ is denoted by N^J / F and is called a *ultrapower*.

Theorem 4.1 (Lós' Theorem)
Let F be an ultrafilter on a set J and let $M = \Pi_{j \in J} M_j / F$ be an ultraproduct. Then, we have:

1. Let $s = ((g_1)_F,...,(g_n)_F,...)$ be a denumerable sequence of elements of $\Pi_{j \in J} D_j / F$. For each $j \in J$, let s_j be the denumerable sequence $(g_1(j),...,g_n(j),...) \in D_j$. Then, for any formula A, s satisfies A in M iff $\{j \in J \mid s_j$ satisfies A in $M_j\} \in F$.
2. For any sentence A of L. A is true in $\Pi_{j \in J} M_j / F$ iff $\{j \in J \models_{M_j} A\} \in F$.

Proof: As in the classical case, by induction on the number m of connectives and quantifiers in A: the only new addition to the proof is to take into account theorem 3.3.

Corollary 4.2
Given a family $(M_j)_{j \in J}$ of interpretations for L, and an ultrafilter F on J, $M = \Pi_{j \in J} M_j / F$ is paraconsistent iff there exists annotated atoms A and B such that

1. $T(B, (M_j)_{j \in J}) \in F$,
2. $I(A, (M_j)_{j \in J}) \in F$.

Proof: Suppose that $\Pi_{j \in J} M_j / F$ is paraconsistent. Then, there are annotated atoms A and B such that

1. $\models_M A$ and $\models_M \neg A$, and
2. $\not\models_M B$.

But (1) iff $\{j \in J \models_{M_j} A\} \in F$ and $\{j \in J \models_{M_j} \neg A\} \in F$. So, $\{j \in J \models_{M_j} A$ and $\models_{M_j} \neg A\} \in F$, and $I(A, (M_j)_{j \in J}) \in F$.

Also, $\{j \in J \models_{M_j} B\} \notin F$, so $\{j \in J \not\models_{M_j} B\} \in F$, namely, $T(B, (M_j)_{j \in J}) \in F$.

Conversely, assume that $I(A, (M_j)_{j \in J}) \in F$ and $T(B, (M_j)_{j \in J}) \in F$. Then, $I(A, (M_j)_{j \in J}) = \{j \in J \models_{M_j} A$ and $\models_{M_j} \neg A\}$, so $\{j \in J \models_{M_j} A\}$, $\{j \in J \models_{M_j} \neg A\} \in F$. By Lós' theorem, $\models_M A$ and $\models_M \neg A$.

Also, if $\models_M B$, then $\{j \in J \models_{M_j} B\} \in F$. So,

$$T(B, (M_j)_{j \in J})[= \{j \in J \not\models_{M_j} B\}] \cap \{j \in J \models_{M_j} B\} = \varnothing$$

which is a contradiction. So, $\not\models_M B$. Therefore, $\Pi_{j \in J} M_j / F$ is paraconsistent.

Corollary 4.3
Given a family $(M_j)_{j \in J}$ of interpretations for L, and an ultrafilter F on J, $\Pi_{j \in J} M_j / F$ is paracomplete iff there exists an annotated atom A such that

$$P(A, (M_j)_{j \in J}) \in F.$$

Corollary 4.4
Given a family of interpretations for L, and an ultrafilter on J, $\Pi_{j \in J} M_j / F$ is non-alethic iff there exists annotated atom A, B, C such that

$$I(A, (M_j)_{j \in J}) \in F,$$
$$T(B, (M_j)_{j \in J}) \in F,$$
$$P(C, (M_j)_{j \in J}) \in F.$$

The present paper reveals that the ultraproduct method can also be applied to annotated logics to obtain some model-theoretic results. The ultraproduct method is an interesting tool for studying models in classical logic. It is also promising for annotated logics. We believe that the results in this paper motivate the use of "paraconsistent logics" in the context of model theory. There is a sense in which paraconsistent annotated models can generalize classical models. Future work will thus address further developments of a paraconsistent annotated model theory started from Abe [2].

ACKNOWLEDGMENTS

The first author was supported by FAPESP Grant 97/02328-9. We are grateful to a referee for his suggestions on the presentation of the paper.

J.M. Abe:
Department of Informatics, ICET — Paulista University,
R. Dr. Bacelar, 1212, 04026-002, São Paulo, SP, Brazil.
e-mail: jmabe@lsi.usp.br
and
Institute for Advanced Studies,
University of São Paulo, São Paulo, Brazil.

S. Akama:
Computational Logic Laboratory, Department of Information Systems,
Teikyo Heisei University, 2289 Uruido, Ichihara-shi, Chiba, 290-0193,
Japan.
e-mail: akama@thu.ac.jp

REFERENCES

[1] Abe, J. M. (1992): *On the Foundations of Annotated Logics* (Portuguese), Ph.D. Thesis, University of São Paulo, Brazil, 135 p.

[2] Abe, J. M. (1993): On annotated model theory, *Coleção Documentos, Série Lógica e Teoria da Ciência*, IEA-USP, no. 11.

[3] Arruda, A. I. (1980): A survey of paraconsistent logic, *Mathematical Logic in Latin America*, North-Holland, Amsterdam.

[4] Bell, J. L. and Slomson, A. B. (1971): *Models and Ultraproducts*, North-Holland, Amsterdam.

[5] Blair, H. and Subrahmanian, V. S. (1989): Paraconsistent logic programming, *Theoretical Computer Science 68*, 135–154.

[6] Blair, H. and Subrahmanian, V. S. (1988): Paraconsistent foundations for logic programming, *Journal of Non-Classical Logic 5*, 44–73.

[7] da Costa, N. C. A., Henschen, L. J., Lu, J. J. and Subrahmanian, V. S. (1990): Automatic theorem proving in paraconsistent logics: foundations and implementation, *Proc. of the 10th International Conference on Automated Deduction*, 72–86, Springer, Berlin.

[8] da Costa, N., Subrahmanian, V. and Vago, C. (1991): The paraconsistent logic $\mathbf{P}_\mathcal{T}$, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik 37*, 139–148.

[9] da Costa, N., Abe, J. and Subrahmanian, V. (1991): Remarks on annotated logic, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik 37*, 561–570.

[10] Fitting, M. (1988): Logic programming on a topological billatice, *Fundamenta Mathematicae 11*, 209–218.

[11] Kifer, M. and Lozinskii, E. L. (1989): RI: a logic for reasoning with inconsistency, *Proc. of LICS4*, 253–262.

[12] Kifer, M. and Subrahmanian, V. S. (1989): On the expressive power of annotated logic programs, *Proc. of the 1989 North American Conference on Logic Programming*, 1069–1089.

[13] Mendelson, E. (1987): *Introduction to Mathematical Logic*, 3rd Edition, Wadsworth and Brooks, Monterey.

[14] Subrahmanian, V. (1987): On the semantics of quantitative logic programs, *Proc. of the 4th IEEE Symposium on Logic Programming*, 173–182.