The contextual logic

Arnaud Kohler

The contextual logic

Kohler Arnaud
arnaud.kohler@pacariane.com

Abstract— **The propositional logic L_p is the smallest syntax that formalizes Aristotle's three principles. It is commonly accepted that it is insufficient to capture human reasoning. Many formalisms have been proposed to extend the modeling capabilities of formal languages. The most common approach is to extend or to impoverish the syntax of L_p. We propose, with the contextual logic L_c, to take a different path. It consists in automatically integrating into the set of atomic propositions of the language *silent* propositions, which we call thoughts. By identifying the formulae, they bring to the formalism a reflexive reasoning capacity. We use it to define a semantic interpretation function of models, which captures the notions of inconsistency and predicate. The contribution of L_c to the family of non-classical formalisms is that it models fallibilistic reasoning (an intelligent agent has no certainty and believes what seems justifiable to him) and perspectivist reasoning (his beliefs are obtained by summing up the beliefs he has from several disjoint perspectives). We illustrate the behavioral properties of the contextual logic by developing at length an example of application. It allows us to present how to use it in the framework of Symbolic Artificial Intelligence.**

Keywords*— *Formal Language, Nonmonotonic Logic, Belief Revision, Knowledge Representation, Reasoning, Artificial Intelligence, Fallibilism, Perspectivism, Cognitive Science

Résumé— **La logique propositionnelle L_p est la plus petite syntaxe qui formalise les trois principes d'Aristote. Il est communément admis qu'elle est insuffisante pour capturer le raisonnement humain. De nombreux formalismes ont été proposés pour étendre les capacités de modélisation des langages formels. L'approche la plus souvent étudiée consiste à étendre ou à appauvrir la syntaxe de L_p. Nous proposons, avec la logique contextuelle L_c, d'emprunter une voie différente. Elle consiste à intégrer automatiquement dans l'ensemble des propositions atomiques du langage des propositions *silencieuses*, que nous appelons des pensées. En identifiant les formules, elles apportent au formalisme une capacité de raisonnement réflexif. Nous l'utilisons pour définir une fonction d'interprétation sémantique sur les modèles, qui capture les notions d'inconsistance et de prédicat. L'apport de L_c à la famille des formalismes non-classiques est qu'elle modélise le raisonnement faillibiliste (un agent intelligent n'a aucune certitude et croit ce qui lui semble justifiable) et le raisonnement perspectiviste (ses croyances sont obtenues en additionnant les croyances qu'il a de plusieurs perspectives disjointes). Nous illustrons les propriétés comportementales de la logique contextuelle en développant longuement un exemple d'application. Il nous permet de présenter comment l'utiliser dans le cadre de l'Intelligence Artificielle Symbolique.**

Mots clefs*— *Langages Formels, Non Monotonie, Révision des Croyances, Représentation de la Connaissance, Raisonnement, Intelligence Artificielle, Faillibilisme, Perspectivisme, Sciences Cognitives

I. INTRODUCTION

The propositional logic L_p is the smallest syntactic model that formalizes Aristotle's three principles:

- of identity (a proposition is what it is),
- of the excluded third (a proposition is true or false),
- and of non-contradiction (a proposition cannot be both true and false).

Formalisms based on L_p classically focus on modeling knowledge that is deemed to be true. It is insufficient to model the full diversity and complexity of human reasoning, which also exploits inconsistent or uncertain information.

Many propositions have been presented to cover this need. They consist in enriching or impoverishing the syntax of L_p to express levels of necessity (epistemic modal logics), universal quantification (predicate logics), inconsistency (paraconsistent logics), new rules of syntactic production (such as default logic), multiple interpretation strategies (adaptive logics), or multi-valued interpretation (multivalued logics).

They each manage to capture different properties. But they have not succeeded in modelling in a single language the many modes of reasoning empirically observed in humans (D. Andler [2]).

One need is to move from a "monotonic" semantics (what is true remains true) to a "non-monotonic" semantics (what is true at one moment may be false the next moment). To achieve this, the contextual logic L_c proposes the following approach: remain strictly within the monotonic syntax of the propositional logic L_p and enrich the set of atomic propositions by *silent* propositions, integrated and consumed automatically.

We call them thoughts. They identify the formulae belonging to the set of knowledge. This gives the formalism a capacity for reflexive reasoning. This property is used to define a non-monotonic semantic interpretation function based on the exploitation of thoughts in the models of the theory.

We first present L_p to lay the theoretical foundations of our work and to share the vocabulary we use and the associated definitions. The supposed limits of the syntax of L_p are recalled.

The principles of L_c and its main properties are then described. It strictly respects the syntax of L_p. Its contribution to the family of formal languages is to propose a formalism that models a non-monotonic, *de dicto*, fallibilistic (an intelligent agent has no certainty and believes what seems justifiable) and perspectivist (its beliefs are obtained by summing up the beliefs it has from several disjoint perspectives) semantic.

The supposed limitations of the propositional logic are then revisited. The properties of L_c make it possible to capture behaviors previously thought to be beyond the reach of the syntax of L_p. We show how they can be used to model the notions of inconsistent knowledge and predicate, to exploit inductive or abductive reasoning methods, and to circumvent the problem posed by the complexity of associated algorithms.

To illustrate this, we conclude our presentation by developing an example of the application of L_c. It calls upon a sufficiently broad knowledge base to demonstrate, through a practical case, the non-monotonic expressiveness of the language and to give meaning to the various technical examples used throughout the text.

Our first aim in this paper is to show that it is possible to deal with the subject of non-monotonicity within the strict framework of the syntax of L_p. However, we agree with D. Batens when he states [13]:

"*The initial goal of the study of logic is to explain human reasoning*"

Our work is rooted in mathematical logics. But our quest for Artificial Intelligence has direct links with computer science, philosophy, and cognitive science. It will not be possible not to address these topics to clarify some of our remarks. However, where necessary, we limit ourselves to introductory presentations. Documentation is readily available if needed.

II. THE PROPOSITIONAL LOGIC

In this article, we refer to several formal languages. We do not detail them in general so as not to make the presentation unnecessarily heavy, inviting the reader to refer to the many documents available on these formalisms.

However, we think it is useful to pause for a moment on the propositional logic L_p. This paragraph does not contain anything new. Its purpose is to share definitions of the vocabulary and symbols we use.

The syntax of L_p

The language of the propositional logic L_p is composed of:

- atomic propositions, forming the set P_{Lp},
- the connector of negation noted $\neg$,
- the connector of implication noted $\rightarrow$,

and parenthesis symbols, which are used according to the classical mathematical rules.

The rules for forming a **well-formed formula** are:

- any atomic proposition is a well-formed formula,
- if f and g are well-formed formulae, then the expressions f, (f), $\neg f$ and $f \rightarrow g$ are well-formed formulae,
- a well-formed formula is obtained only by applying the two precedent rules a finite number of times.

Let f, g and h be some well-formed formulae. The following formulae are some **axioms**:

- $f \rightarrow (g \rightarrow f)$
- $(f \rightarrow (g \rightarrow h)) \rightarrow ((f \rightarrow g) \rightarrow (f \rightarrow h))$
- $(\neg f \rightarrow \neg g) \rightarrow ((\neg f \rightarrow g) \rightarrow f)$

These three axioms are sufficient to cover all the axioms of L_p. For example, $f \rightarrow f$ is another axiom, which can be demonstrated from these three using the **theorem formation rules**:

- every axiom is a theorem,
- let f and g be well-formed formulae. If f and $f \rightarrow g$ are theorems, then g is a theorem (this rule is called the *modus ponens*),
- a theorem can only be obtained by applying the two previous rules a finite number of times.

The statement f is a theorem is denoted $\vdash_{Lp} f$.

A **theory** E_{Lp} is a set of well-formed formulae. The formulae $f \in E_{Lp}$ represent the hypotheses of E_{Lp}.

A formula f is said to be provable in E_{Lp} if, and only if, it can be produced from E_{Lp} by applying the theorem formation rules, for all hypotheses of E_{Lp} behaving as theorems. In this case, f is a **theorem** of E_{Lp}, and it is denoted $E_{Lp} \vdash_{Lp} f$:

A theory is said to be **consistent** if it does not produce the negation of a theorem. Otherwise, it is said to be **inconsistent**.

To simplify the expression of formulae, the language is commonly extended to disjunction (denoted $\vee$), conjunction (denoted $\wedge$) and equivalence (denoted $\leftrightarrow$) connectors. For f and g well-formed formulae, they are defined by:

- $f \vee g$ is equivalent to $(\neg f \rightarrow g)$,
- $f \wedge g$ is equivalent to $\neg (f \rightarrow \neg g)$,
- $f \leftrightarrow g$ is equivalent to $(f \rightarrow g) \wedge (g \rightarrow f)$

A **literal** is an atomic proposition or the negation of an atomic proposition. A **clause** is a disjunction of literals.

A formula is said to be in **normal form** if it is a conjunction of clauses. A. Thayse [23] indicates that any well-formed formula admits a logically equivalent rewriting in normal form. For example, the normal form of the formula $((\neg f \rightarrow g) \rightarrow h)$ is $((\neg f \vee h) \wedge (\neg g \vee h))$.

Let E_{Lp} be a theory, f be a clause and a be a literal such that $E_{Lp} \vdash_{Lp} f \vee a$. $f \vee a$ is in its **minimal clausal form** if f is not a theorem of E_{Lp}. And the normal form of a formula is said to be in its **minimal normal form** if each clause in it is unique and is in its minimal clausal form.

P. Siegel [22] proposes a linear complexity process that rewrites any well-formed formula into its minimal normal form.

The semantic of L_p

Classically, the logician's attitude is to see in formal language only mathematical symbols. What is relevant is the study of the mechanisms and laws of reasoning, modeled by syntactic rules. Any reference to semantic content is discarded. However, it is possible to attribute a meaning to connectors if it is strictly symbolic and univocal.

Let E_{Lp} be a theory of L_p, and f and g two well-formed formulae. The syntactic interpretation function of L_p is defined by a function I_{Lp} such that:

- $I_{Lp}(E_{Lp}, f) = true$ or exclusively $I_{Lp}(E_{Lp}, f) = false$ or exclusively $I_{Lp}(E_{Lp}, f) = unknown$
- If f is a hypothesis of E_{Lp}, then $I_{Lp}(E_{Lp}, f) = true$

The meaning of the connectors is then defined by:

- $I_{Lp}(E_{Lp}, \neg f) = true$ if, and only if, $I_{Lp}(E_{Lp}, f) = false$
- $I_{Lp}(E_{Lp}, f \rightarrow g) = true$ if, and only if, $I_{Lp}(E_{Lp}, f) = false$ or $I_{Lp}(E_{Lp}, g) = true$

The syntactic interpretation $\vDash_{Lp}$ is defined by:

$$E_{Lp} \vDash_{Lp} f \text{ if, and only if, } I_{Lp}(E_{Lp}, f) = true$$

The *unknown* value should not be understood as a third truth value. It is used to indicate that the constraints of the theory do not allow to compute the truth value *true* or *false*. For example, the value of $I_{Lp}(\{a \rightarrow b\}, a)$ is not computable.

The syntactic interpretation of an axiom is always true, and L_p is adequate and complete: everything that is produced (using $\vdash_{Lp}$) is true, and everything that is true (according to $\vDash_{Lp}$) is produced.

$$E_{Lp} \vDash_{Lp} f \text{ if, and only if, } E_{Lp} \vdash_{Lp} f$$

We conclude this presentation of propositional logic by giving a definition that we will use a lot. Let E_{Lp} be a theory of L_p. A **model** of E_{Lp} consists in associating to each atomic proposition only one truth value (*true* or exclusively *false*) such that the result verifies the logical constraints expressed by E_{Lp}. E_{Lp} is consistent if it has at least one model. It is inconsistent otherwise.

For example, the theory $E_{Lp} = \{a, b \rightarrow c\}$ is verified by three models:

$$\{(a, true), (b, true), (c, true)\}$$
$$\{(a, true), (b, false), (c, true)\}$$
$$\{(a, true), (b, false), (c, false)\}$$

So, it is consistent. As a counter example, $\{a, \neg a\}$ does not accept a model: if a is assumed to be *true*, $\neg a$ is not verified - and *vice versa*. So, it is inconsistent.

III. THE LIMITS OF L_P

Modeling human reasoning with L_p has encountered four major difficulties.

1) Human reasoning sometimes seems incoherent. But syntactic inconsistency leads to the production of everything and its opposite: whatever f and g two well-formed formulae of L_p, $\{f, \neg f\} \vdash_{Lp} g$. This is the explosion principle. It forbids the appearance of a syntactic inconsistency in a theory. This topic is covered in paragraph V.

2) Human reasoning uses semantic links between propositions. But the symmetrical behavior of connectors prohibits this type of modeling in L_p. Put more explicitly with an example, $f \rightarrow (g \rightarrow h)$ is syntactically equivalent to $g \rightarrow (f \rightarrow h)$: f and g have the same behavior in the formula, and neither of them has a privileged relationship with h. This topic is covered in paragraph VI.

3) Theory distinguishes three modes of reasoning found in humans: deduction (establishing a particular law from general facts), induction (establishing a general law from particular facts) and abduction (identification of the most likely cause of an observed event). Deduction is formalized by the *modus ponens* rule, but the other two modes escape the formalism of L_p. This topic is covered in paragraph VII.

4) The computational algorithms associated with L_p are of exponential complexity. In practice, it takes several seconds to deduce knowledge using a base of a few dozen formulae. This is obviously not acceptable in the context of human reasoning, and more generally in the context of Artificial Intelligence. It requires a good level of responsiveness. This topic is covered in paragraph VIII.

The successive failures of logic researchers to solve these problems have led many to conclude that the modeling of human reasoning probably escapes L_p, and logic formalisms more generally (D. Andler [2]).

IV. THE CONTEXTUAL LOGIC

Let us consider a thought. We perceive it in the sense defined by R. Descartes [4]:

> "*By the name of thought, I understand all that is so much in us that we are immediately aware of it*" (translation)

and we describe it with a set of sentences. However, even if this description were ideally complete and perfect, we are immediately aware that it is not the thought it describes.

We model this observation by distinguishing, for a given thought, two notions in the syntax of the language: a unitary sign c, which symbolizes it, and a combination of signs f, which reproduces the sentences that describe it. This leads to the need to define a relationship between c and f. To this end, we consider the following postulate [11]:

Contextual postulate *Let L be a formal language with the functions of syntactic production $\vdash_L$ and the syntactic interpretation $\vDash_L$. A well-formed formula f of L is a set of signs that has no meaning. Its meaning is carried by a thought, which is an atomic proposition of L "which is not pronounced". For c symbolizing this thought, the relation between c and f is $c \vDash_L f$.*

The expression $c \vDash_L f$ asserts neither the thought c nor the sentence f. It models that the sentence f expresses the thought c. c is an atomic proposition which respects the syntactic properties of L. So, $c \vDash_L f$ is equivalent to $\vDash_L c \rightarrow f$ if L is the propositional logic L_p.

For example, if we consider two sentences f and g that express the thoughts c_1 and c_2 respectively, we should not formalize this knowledge by the set $\{f, g\}$ (because f and g are some sets of signs that has no meaning) but by $\{c_1 \rightarrow f, c_2 \rightarrow g\}$. It models: "*$c_1$ is expressed by f and c_2 is expressed by g*". We thus agree with L. Wittgenstein when he states [25]:

> "*We should not say: The complex sign aRb says that a is in the relation R with b, but: That a is in a certain relation R with b says that aRb*" (translation)

The application of the postulate to a formalism L produces the contextualized logic L. By language convention, we call contextual logic, denoted L_c, the contextualized propositional logic.

Each contextual formula takes a form $c \rightarrow f$, for c a thought (and an atomic proposition) and f a well-formed formula in the sense of L_p. Expressions in L_c accept a natural order:

- atomics propositions of L_p are at rank 0,
- a thought is at rank 1 or higher. We will see later that we propose to automatically handle the assignment of a rank to a thought,
- the rank of a well-formed formula in the sense of L_p is equal to the maximum rank of the atomic propositions (including thoughts) that compose it.

We define a well-formed formula in the sense of L_c to be a formula $c \rightarrow f$, for c a thought of rank n and f a well-formed formula in the sense of L_p of rank m, such that $m < n$.

No well-formed expressions are acceptable. They allow for $\vdash_{Lp}$ production and $\vDash_{Lp}$ syntactic interpretation. However, we only lend them meaningless technical behavior, which makes them useless in the context of the language.

Given the syntax $c \rightarrow f$ of the contextual formulae, the set $\{(c_i, false), c_i$ are the thoughts$\}$ characterizes some models that verify any contextual theory. For example, $\{(c_1, false), (c_2, false)\}$ characterizes some models that verify $\{c_1 \rightarrow f, c_2 \rightarrow g\}$.

The first consequence is that a contextual theory admits at least one model. So, it is always consistent.

The second consequence is that contextual logic is unable to produce certainty. Put differently, in L_c, any thought is possibly false, and, as a direct effect of the application of the form $c \rightarrow f$, any formula f (except for axioms), and so any atomic proposition of L_p, can be true or false: there is no certainty in L_c. Uncertainty is intrinsically embedded in the syntax.

To remedy this problem, L_c adopts the following principles:

- because any formula can be true or false, there is no need to interrogate a contextual theory with a question such as "Is f true or false?". The solution adopted is to ask the question in the form: "What can I conclude if I suppose that f (or $\neg f$) is true?".

  Notation *f is called the stimulus. It is possibly empty, and it is denoted S_{Lc}.*

- because every thought is possibly false, we propose to relativize the semantic interpretation to the subsets of thoughts identified as *the most relevant*. We cannot conclude that f is true or false, but we can say "f is true (or false) with respect to the most relevant sets of thoughts".

For example, let a and b be two atomic propositions of L_p, and c_1, c_2 and c_3 be three thoughts. Consider the following set:

$$E_{Lc} = \{c_1 \rightarrow a, c_2 \rightarrow \neg a, c_3 \rightarrow b\}$$

We cannot product that a or b is true or false. But we can say that a is true considering $\{c_1\}$, or that b is true considering $\{c_2, c_3\}$, etc. There are many possible combinations, so we should define a method for selecting "the most relevant sets of thoughts".

For this purpose, we need some definitions.

Definitions *Let E_{Lc} be a theory of L_c and i and j be 2 integers such that $0 < i \leq j$.*

*A set of thoughts is called a **context**.*

*A context is said to be **of rank i to j** if all the thoughts in it are of rank i to j. A context of rank i to i is said of rank i.*

*A context that is verified by at least one model of E_{Lc} is called a **possible** (or a consistent) **context**.*

*A context that does not check any model of E_{Lc} is called an **impossible** (or an inconsistent) **context**.*

*An impossible context is called a **strict impossible context** if each of its strict subsets is possible.*

*A possible context that has no strict extension that checks E_{Lc} is called a **maximum context**.*

*A possible context is called **the credible context** if it has no join with a strict impossible context and if all its strict extensions have a join with a strict impossible context.*

In the following, and in accordance with common practice, we will invariably use the notions of conjunction of formulae (for example: $c_1 \wedge c_2$) or of set of formulae (for example: $\{c_1, c_2\}$) to designate the same object. A conjunction of thoughts also means a context.

Example *Let a, b and c be three atomic propositions of L_p, and c_1, c_2, c_3, c_4 and c_5 be five thoughts. Consider the following set:*

$$E_{Lc} = \{c_1 \rightarrow a, c_2 \rightarrow \neg a, c_3 \rightarrow b, c_4 \rightarrow \neg b, c_5 \rightarrow c\}$$

$c_1 \wedge c_2$ and $c_3 \wedge c_4$ are the only two strict inconsistent contexts. So, c_5 is the credible context, and there are four maximal contexts: $\{c_1, c_3, c_5\}$, $\{c_1, c_4, c_5\}$, $\{c_2, c_3, c_5\}$ and $\{c_2, c_4, c_5\}$.

We see that, for a given theory, there are possibly several maximal contexts (potentially empty) and a single credible context (potentially empty). They are obtained by calculating the strict incoherent contexts in a first step. The different possible combinations of thoughts then produce them.

We use these definitions to define the function that identifies the contexts considered most relevant for semantic interpretation.

Definition *Let E_{Lc} be a theory of L_c, S_{Lc} be a stimulus and T_c and T_m be two integers such that $0 < T_c < T_m$. The relevant contexts are defined as follows:*

- *calculation of the maximal contexts of rank T_m and above on $\{E_{Lc}, S_{Lc}\}$. This defines a set of contexts $\{C_k\}$,*
- *then enrichment of each maximal context C_k, by the credible context of rank T_c to T_m-1 on $\{E_{Lc}, S_{Lc}, C_k\}$.*

*This defines the set of **epistemic contexts**. It is denoted $C_{ELc, SLc, Tc, Tm}$.*

This definition presents the notion of epistemic contexts. They are *the most relevant sets of thoughts*, which meets the need we identified earlier.

Other definitions are possible, for example by using the ranks of thoughts more finely. Epistemic contexts are sufficient for the modelling needs presented in this article.

We are now able to define the semantic interpretation function of L_c.

Definition *Let E_{Lc} be a theory, S_{Lc} be a stimulus and T_c and T_m be two integers such that $0 < T_c < T_m$. A well-formed formula f is said:*

- ***conceivable*** *if there is at least one epistemic context C_1 such that $\{E_{Lc}, S_{Lc}, C_1\} \vDash_{Lp} f$ and there is at least one epistemic context C_2 such that $\{E_{Lc}, S_{Lc}, C_2\} \vDash_{Lp} \neg f$,*
- ***credible*** *if there is at least one epistemic context C_1 such that $\{E_{Lc}, S_{Lc}, C_1\} \vDash_{Lp} f$ and there is no epistemic context C_2 such that $\{E_{Lc}, S_{Lc}, C_2\} \vDash_{Lp} \neg f$,*
- ***improbable*** *if there is at least one epistemic context C_1 such that $\{E_{Lc}, S_{Lc}, C_1\} \vDash_{Lp} \neg f$ and there is no epistemic context C_2 such that $\{E_{Lc}, S_{Lc}, C_2\} \vDash_{Lp} f$,*
- ***not interpretable*** *in other cases.*

*$\{E_{Lc}, S_{Lc}, C \in C_{ELc, SLc, Tc, Tm}\}$ is said a **semantic perspective**.*

This definition presents the basic semantic interpretation function of L_c. It can be enriched, for example by distinguishing true formulae in all semantic perspectives. This version is sufficient for the modelling needs presented in this article.

Example *Let a and b be two atomic propositions of L_p, c_1 and c_2 be two thoughts of rank 1, and c_3 and c_4 be two thoughts of rank 2. Consider the following set:*

$$E_{Lc} = \{c_1 \rightarrow a, c_2 \rightarrow \neg b, c_3 \rightarrow c_1, c_4 \rightarrow \neg c_1\}$$

Let T_c=1 and T_m=2, and we consider the stimulus is empty. $\{c_3, c_4\}$ is incoherent, so $\{c_3\}$ and $\{c_4\}$ are the two maximal contexts of rank 2. Let's add to each the credible context of rank 1 associated with it to calculate the two epistemic contexts. We obtain:

- *$\{c_3, c_1, c_2\}$. The associated semantic perspective says that a is true and b is false (according to $\vDash_{Lp}$),*
- *$\{c_4, c_2\}$. The associated semantic perspective says that b is false (according to $\vDash_{Lp}$).*

So, according to the semantic vocabulary of L_c, a is credible and b is improbable.

The semantic interpretation function has a mathematical definition and is therefore very rigorous. This is not necessarily compatible with our natural language habits. Therefore, we will allow ourselves some linguistic shortcuts, for example:

- a perspective designates a semantic perspective,
- a well-formed formula in the sense of L_c is a piece of knowledge, so a theory is a set of knowledges,
- a belief is a formula which expresses a thought or a set of thoughts,
- a conceivable expression is also said true and false or possible,
- a credible expression is also said true, conceivable, or possible,
- an improbable expression is also said false, conceivable, or impossible.

We will use them in a way that does not create confusion.

We now present the properties of L_c. Its contribution to the family of non-classical logics is that it is the only formal language that simultaneously verifies these behaviors.

We remain on a technical observation and not discuss their relevance. Indeed, each property echoes philosophical concepts and deserves a dedicated article. The debates are rich, and there are as many defenders as detractors. We do not bring new elements to enrich these exchanges. The interested reader will easily find in literatures in-depth presentations of these topics.

The syntax production function is monotonic

L_c respects the syntax of the propositional logic and is therefore syntactically monotonic: whatever f and g are contextually well-formed formulae, if a theory E_{Lc} produces f then $\{E_{Lc}, g\}$ produces f.

Note that the syntactic interpretation function is mechanically also monotonic.

The semantic interpretation function is non-monotonic

L_c decorrelates the syntactic interpretation function from the semantic interpretation function. Syntax produces a set of formulae according to the rules of the contextualized formalism.

Semantics then provides an interpretation by analyzing the models of the theory. They are considered as they are produced by the syntax. We do not employ the concept of extension sometimes used by non-classical formalisms.

The models of a theory can change if a new piece of knowledge is introduced. So, $C_{ELc,\, Tc,\, Tm}$ must be recalculated in this case, and L_c has a non-monotonic semantic: considering the same stimulus S_{Lc}, a formula f can be credible considering $\{E_{Lc}, S_{Lc}\}$ and incredible considering $\{(E_{Lc}, g), S_{Lc}\}$, for g a new piece of knowledge. We will see examples of use in the following paragraphs.

L_c brings a reflexive capacity to reasoning

The contextual postulate models the relationship between thoughts and the sentences that express them. By distinguishing expression and thought, and by identifying thought, it brings a reflexive capacity to formal language: thoughts can reason about themselves using the constraints carried by the sentences that express them.

It thus brings formal language closer to natural language: formalism is used as a means of expression. The properties carried by the syntax of language model the reasoning mechanisms of thought (J. Fodor [5]). They generate, by opportunity, the ability to reason about *what is said*.

The syntax has a *de dicto* behavior

De dicto and *de re* are two locutions that distinguish two important modalities of statements and the reasoning behind them. *De dicto* means in Latin *about what is said*, and *de re* means *about the fact*.

In L_c, a piece of knowledges ($c \rightarrow f$) does not express a fact, but models that the thought c is expressed by a sentence f. Note that contextual postulate supposes that language is unable to capture thoughts completely ($c \rightarrow f$ is not $c \leftrightarrow f$).

Contextual logic thus distinguishes between the thought c, which is the whole that one wishes to express, and the sentence f, which is the way it is said. The reasoning then exploits the logical relations that appear in the sentence which is said.

The semantic is fallibilism

Non-monotonicity is a matter of completeness of knowledge: a belief that is true in one state may be false in an enriched state. Fallibilism (K. Popper [18]) is a more radical philosophical principle. It assumes that absolute knowledge is impossible: all belief can, at any time, be questioned – and possibly contradicted.

The syntax of L_c is that of L_p. It is therefore based on axiomatic principles which it considers as absolute. However, a consequence of the contextual postulate is that every proposition is possibly false. L_c thus proposes the paradox of relying on a syntax considered as absolutely true to model knowledge interpreted semantically as absolutely uncertain.

To avoid this, the solution is to consider that what is not explicitly false is credible and will remain so until it is explicitly contradicted or challenged.

We will illustrate this with some examples which we will develop in the following paragraphs.

The semantic is perspectivist

Perspectivism (F. Kaulbach [10]) refers to philosophical doctrines that defend the idea that our perception of reality is composed of the sum of the perspectives we have on it.

In L_c, the semantic interpretation is obtained by considering the interpretations, possibly contradictory, of each epistemic context: truth is not the consequence of a global point of view built on the whole of thoughts, but the juxtaposition of several points of view from distinct subsets of thoughts.

Propositions are attributes, not assertions

In the most adopted mathematical approach, an atomic proposition is an assertion apprehended in its content. Considering a theory, its semantic interpretation admits a truth value: it is true or false (or possibly another value in the case of multi-valued formalisms).

In L_c, it is not possible to deduce that a proposition is true or false according to L_p. This is a mechanical consequence of the application of the contextual postulate. A proposition (or a formula) can only be interpreted in relation to a set of thoughts, called a context. *It characterizes it*.

So, in L_c, a proposition is not an assertion in the strict sense of the term. It must be understood as a characteristic, or an attribute, of the context.

Consider, for example, the sentence "*If Tweety is a bird, then it flies*". Its modeling in predicate logic can be:

$$Bird(Tweety) \rightarrow Flying(Tweety)$$

In L_c, this assertion is modeled by:

$$c_1 \rightarrow (Tweety \rightarrow Bird)$$
$$c_2 \rightarrow (Bird \rightarrow Flying)$$

which allows for several readings, for example: attributes *Bird* and *Flying* are attributes of the context $\{c_1, c_2\}$ if we consider the stimulus *Tweety*, or are attributes of the stimulus *Tweety* if we consider the context $\{c_1, c_2\}$.

We have finished with the presentation of L_c. The following sections show how to use its properties to provide answers to the difficulties identified in paragraph III.

Considering the definition of epistemic contexts, T_c and T_m can theoretically take any value. According to the work of J. Pitrat [17], there are probably cognitive thresholds limiting the capacities of human reasoning. In the rest of this document, we will use the thresholds *2* and *3*, which are sufficient to cover the expected level of expressiveness expected in this article. And by writing convention, we will henceforth note $c_{i,j}$ the thoughts. *i* singularizes the proposition and *j* indicates its rank.

V. MODELING AN INCONSISTENT KNOWLEDGE

For example, consider a set of L_p's propositions $\{a, b, c\}$ and let be the following set:

$$E_{Lp} = \{a \rightarrow b, a \rightarrow \neg b, c, a\}$$

It is inconsistent because $E_{Lp} \vdash_{Lp} b \wedge \neg b$. According to the explosion principle, whatever f an assertion, $E_{Lp} \vdash_{Lp} f$. This is not acceptable.

E_{Lp} has no meaning according to the contextual postulate. Let us now place ourselves in the contextual logic framework. Considering the set of thoughts $\{c_{10,2}, c_{20,2}, c_{30,2}, c_{40,2}\}$, we assume the following theory:

$$E_{Lc} = \{ \begin{array}{l} c_{10,2} \rightarrow (a \rightarrow b), \\ c_{20,2} \rightarrow (a \rightarrow \neg b), \\ c_{30,2} \rightarrow c, \\ c_{40,2} \rightarrow a \quad \} \end{array}$$

E_{Lc} is consistent, and there is an incoherence between the three thoughts $c_{10,2}$, $c_{20,2}$, and $c_{40,2}$ because:

$$\{E_{Lc}, c_{10,2}, c_{20,2}, c_{40,2}\} \vdash_{Lp} b \wedge \neg b$$

$\{c_{10,2}, c_{20,2}, c_{40,2}\}$ is a strict impossible context. So, $\{c_{30,2}\}$ is the only epistemic context. If the stimulus is empty, we obtain one perspective which says $\{c\}$, and $\{a, c\}$ if the stimulus is $\{a\}$.

This is a first result showing the possibility of exploiting inconsistent beliefs in L_c. The solution is to get around the problem by considering that the thoughts $c_{10,2}$, $c_{20,2}$ and $c_{40,2}$ are not credible because they carry an inconsistency.

We now want to address this inconsistency, by modelling that $a \rightarrow b$ (i.e., the thought $c_{10,2}$) is not always true - or, put differently, is sometimes true and sometimes false. Let's use two new thoughts, $c_{11,3}$ and $c_{12,3}$:

$$E_{Lc} = \{ \begin{array}{l} c_{10,2} \rightarrow (a \rightarrow b), \\ c_{11,3} \rightarrow c_{10,2}, \\ c_{12,3} \rightarrow \neg c_{10,2} \\ c_{20,2} \rightarrow (a \rightarrow \neg b), \\ c_{30,2} \rightarrow c, \\ c_{40,2} \rightarrow a \quad \} \end{array}$$

Considering a is the stimulus, let's calculate the epistemic contexts. There are 2 maximum contexts of rank 3: $\{c_{11,3}\}$ and $\{c_{12,3}\}$. Let us extend each of these contexts to their associated credible contexts of rank 2:

- considering $\{E_{Lc}, a, c_{11,3}\}$, $\{c_{20,2}, c_{40,2}\}$ is the only strict impossible context, so $\{c_{10,2}, c_{30,2}\}$ is the credible context of the rank 2 in this case,
- considering $\{E_{Lc}, a, c_{12,3}\}$, $\{c_{10,2}\}$ is the only strict impossible context, and $\{c_{20,2}, c_{30,2}, c_{40,2}\}$ is the credible context in this.

In fine, considering the stimulus $\{a\}$, we obtain two epistemic contexts:

- $\{c_{11,3}, c_{10,2}, c_{30,2}\}$ which says $\{a, b, c\}$ is true,
- $\{c_{12,3}, c_{20,2}, c_{30,2}, c_{40,2}\}$ which says $\{a, \neg b, c\}$ is true.

We conclude that, taking a as the stimulus, c is credible (or true), and that b is conceivable (or true and false). The formalism does this by modeling an epistemic information: the belief $a \rightarrow b$ (i.e., $c_{10,2}$) is true (what $c_{11,3}$ formalizes) and false (what $c_{12,3}$ formalizes).

We have used a single contradiction $\{c_{11,3}, c_{12,3}\}$ to illustrate our point. If multiple contradictions (two contradictions $\{c_{x1,3}, c_{x2,3}\}$ and $\{c_{y1,3}, c_{y2,3}\}$ for example), the different cases are managed on the maximal contexts on T_m ($\{c_{x1,3}, c_{y1,3}\}$, $\{c_{x1,3}, c_{y2,3}\}$, $\{c_{x2,3}, c_{y1,3}\}$, and $\{c_{x2,3}, c_{y2,3}\}$ with the example). We obtain by combination the set of relevant perspectives. This is illustrated in the example that we develop at the end of the article.

Comparison with other formalisms

In this section, we point out the major gaps in the treatment of inconsistent or incomplete information between L_c and other non-classical formalisms.

Paraconsistent and multivalued logics aim to tolerate inconsistencies by escaping the principle of explosion. The approach, theorized by J. Lukaszewicz [13], is either to weaken Aristotle's principles to limit the inferential capacities of language or to add a third truth value to indicate that the knowledge concerned is both true and false.

L_c addresses the issue of uncertainty and inconsistency through its perspectivist property: it models that something is simultaneously true according to some thoughts and false according to others.

L_c is therefore not a paraconsistent or a multivalued formalism. It strictly preserves the syntax of L_p. Therefore, it does not escape the principle of explosion. If one retains a reference context that syntactically produces $f \wedge \neg f$, then it produces any belief g whatsoever. Inconsistency is accepted in the semantic interpretation of L_c, it remains non-tolerable in its syntax.

Another fundamental difference between L_c and other non-classical formalisms is its fallibilistic property: noting that there is no certainty, it takes as credible what is not explicitly false. In other words, it takes as true everything that is possible and not explicitly impossible. This property gives L_c a particular behavior, which does not allow it to fully capture modal logics or default logics for example.

Default logic is proposed by R. Reiter [20]. To reason with uncertain information, he extends production rules by expressions of the form $(a : b \,/\, c)$ which read: "if a is true and if b is possible then c is produced". In L_c, the thought that b is possible "generates the thought b". However, related to R. Reiter's syntax, L_c's expressiveness is limited to normal default rules, of the form $(a : b \wedge c \,/\, b \wedge c)$ [11].

Modal epistemic logics extend the expressiveness of languages by adding a new connector for reasoning about the quality of the interpretative value. The most widely used epistemic modal connector is the alethic connector $\Box$. $\Box f$ usually expresses that f is necessary, and its dual $\neg\Box\neg f$, denoted $\Diamond f$, that f is possible. The language relies on the semantics of possible worlds of S.A. Kripke [12] to benefit from a syntactic interpretation function of $\Box$.

We have proposed a relationship between modal epistemic logics and L_c [11]. This requires an evolution of the definition of epistemic context, using ranks to capture the imbrications of the monadic connector (rank i for $\Box$, rank $i+1$ for $\Box\Box$, etc.). It models the sets $\{\Box f\}$ and $\{\Diamond f, \Diamond\neg f\}$, but the set restricted to $\{\Diamond f\}$ is interpreted as $\{\Box f\}$: f is considered necessary if the possibility of its opposite is not explicitly expressed. L_c adds a default rule to the K system: $\{\neg\Box\neg f : \Box f / \Box f\}$. In the framework of Kripke's semantics, this expression reads: if I know an accessible world in which f is true, and I do not know an accessible world in which $\neg f$ is true, then I consider that f is true in all accessible worlds.

Default and modal epistemic logics deal with the issue of incoherence by evolving the syntactic capabilities of the language. D. Batens proposes another approach [3]. He considers that there are several reasoning strategies, and that the solution consists in choosing the one that is best adapted to the situation. These are adaptive logics.

Consider, for example, the following set of formulae:

$$E_{La} = \{\neg p, \neg q, p \vee q, p \vee r, q \vee r\}$$

It is incoherent, and therefore explosive in the context of propositional logic. If one adopts a strategy favoring reliable reasoning, it is not possible to deduce r: it would be unwise to conclude anything using the first three formulae. However, if we choice a strategy that minimizes abnormalities, and assume that at least two of the first three formulae are true, then r is produced.

In contextual logic, the set becomes:

$$E_{Lc} = \{ \quad c_{1,2} \rightarrow \neg p,\; c_{2,2} \rightarrow \neg q,\; c_{3,2} \rightarrow p \vee q,\; c_{4,2} \rightarrow p \vee r,\; c_{5,2} \rightarrow q \vee r \quad \}$$

Assume that the stimulus is empty. $\{c_{4,2}, c_{5,2}\}$ is the reference context because $\{c_{1,2}, c_{2,2}, c_{3,2}\}$ is a minimal impossible context. As far as we know, r is not interpretable. Using epistemic contexts that retain the maximal credible contexts at rank 2 is therefore a prudent strategy.

So, L_c is not an adaptive logic. Both formalisms have the capacity to adapt their semantic interpretation to local characteristics: L_c chooses to use or not a thought depending on the stimulus. But its principle is not to adapt his reasoning to a typology of situation. It uses a unique analysis strategy, based on the definition of epistemic contexts.

We end this comparative section with the circumscription logic of J. McCarthy [14]. It consists in extending the set of atomic propositions by propositions that indicate the epistemic character of a formula. For example, "$a \rightarrow b$ is true with some exceptions" is modeled by $((a \rightarrow b) \vee abnormal)$. The atomic proposition *abnormal* indicates that the formula $(a \rightarrow b)$ allows for exceptional behavior.

The models of the theory are then analyzed to select those that minimize the abnormalities. This approach, which consists of looking for a solution in the set of atomic propositions, coupled with an analysis of the theory's models, is most certainly the closest to our own. We have shown that contextual logic can capture its expressive capacity by adapting the definition of epistemic contexts to meet the minimality criterion [11].

However, beyond this result, the choice to minimize abnormalities seems reasonable but can easily be questioned with use cases. This difficulty is shared with adaptive logics, or more generally with the concept of epistemic rooting proposed by P. Gardenfors and D. Makinson [7].

Indeed, these methods suppose the existence of an order relation (on formulae or on reasoning strategies) which would oversee selecting the information in case of incoherence.

In L_c, syntactic consistency is guaranteed. It is therefore not necessary to manage this in the formalism. However, the existence of an order relation is supported by experiments in cognitive science. We shall see this point in paragraph VIII.

VI. MODELING A PREDICATE KNOWLEDGE

L_p sees a proposition as a whole, which is given a universal value. It is then necessary to decompose this whole when we wish to use a singular value. To this end, predicate logic meets this need by allowing the desired relationship to be modeled directly in the elementary proposition. It then becomes possible to model that *Socrates is a man*, and to deduce that *Socrates is mortal* because *a man is a mortal*:

$$Man(Socrates)$$
$$Man(Socrates) \rightarrow Mortal(Socrates)$$

This syllogism uses the link between *Man* and *Socrates* to deduce the association with *Mortal*. In this context, G. Frege [6] theorized the notion of universal quantifier. As a classical example of use:

$$\forall\, x\ Man(x) \rightarrow Mortal(x)$$

which reads: whatever x is, if x is a man then x is mortal. We note two enrichments with respect to the native modeling capabilities of the propositional logic:

- the notion of the universal quantifier $\forall$. We will come back to the subject of universal connectors at the end of the article,
- the possibility of breaking down a proposition into several singular instances. In our example, *Men are mortal* is modeled by using two distinct units, *Man* and x. The expression *Man(x)* creates a syntactic relationship, which formalizes a semantic link, between these two-unit elements.

We have seen in paragraph III that it is not possible to model a relation between two atomic propositions in L_p because of the symmetric behavior of connectors. We now present how this point can be solved in L_c.

Suppose in L_p the set of propositions $\{a, b, c, d, e, f\}$ and the following theory:

$$E_{Lp} = \{\ a \rightarrow b,\ c \rightarrow d,\ a \rightarrow \neg c,\ c \rightarrow (a \rightarrow e),\ c \rightarrow (a \rightarrow f)\ \}$$

We want to model that $a \rightarrow e$ is a predicate of c – i.e., c is the subject of $a \rightarrow e$. The problem is that the formula $c \rightarrow (a \rightarrow e)$ is syntactically equivalent to $a \rightarrow (c \rightarrow e)$.

Now, consider the following set in L_c:

$$E_{Lc} = \{\ c_{10,2} \rightarrow (a \rightarrow b),\ c_{20,2} \rightarrow (c \rightarrow d),\ c_{30,2} \rightarrow (a \rightarrow \neg c),\ c_{40,2} \rightarrow (c \rightarrow c_{41,1}),\ c_{41,1} \rightarrow (a \rightarrow e),\ c_{50,2} \rightarrow (c \rightarrow (a \rightarrow f))\ \}$$

$c_{40,2}$ and $c_{41,1}$ model the predicative knowledge. They break the syntactic symmetry, and introduce two new pieces of information:

- the predicate is syntactically distinguished by $c_{41,1}$,
- $c_{40,2}$ says that $c_{41,1}$ is true if c is true.

$c_{41,1}$ is of rank 1, so, by definition, it does not participate in epistemic contexts.

Let $c \wedge a$ be the stimulus. $\{c_{10,2}, c_{20,2}, c_{40,2}, c_{50,2}\}$ is its epistemic context. The perspective associated says $\{c, a, b, d, e, f, c_{41,1}\}$. The presence of $c_{41,1}$ indicates that $a \rightarrow e$ is a predicate, but the relationship with its subject c is not apparent.

It can be found by applying the following method:

- calculate the perspectives of the stimulus c. It says $\{c, \neg a, d, c_{41,1}\}$. The predicate $c_{41,1}$ is obtained by $c_{40,2}$,
- calculate the perspectives of the stimulus a. It says $\{a, \neg c, b\}$. The thought $c_{41,1}$ is not syntactically produced,
- apply the predicates associated with each perspective to the other perspectives. This produce $\{e\}$ by applying $c_{41,1}$ to the perspective of a. e is associated with the stimulus c associated with the applied predicate,
- and finally, calculate the perspectives of the stimulus $c \wedge a$. This produce $\{f\}$.

If the thought of a predicate appears in a perspective, then it expresses that its subject is its stimulus. Unlike in predicate logic, the notion of predicate is not modelled in the syntax of L_c. It is carried by the semantic meaning of the perspectives.

This method extends the consumption of epistemic contexts by a recursive function:

The semantics of a complex universe is obtained by crossing the semantics of the objects that compose it.

The semantics of a universe composed of three objects A, B and C can only be partially obtained by analyzing the perspectives of $A \wedge B \wedge C$. To obtain a complete perception, we must analyze $\{A\}$, $\{B\}$, $\{C\}$, $\{A, B\}$, $\{A, C\}$, $\{B, C\}$ and $\{A, B, C\}$ separately, and cross-reference the properties associated with these seven different objects.

We call this method generalized contextual semantics. It allows to exhaustively capture the characteristics of each object, of each combination of objects, and to calculate cross-predictive inferences. It seems to produce redundancies. We have not studied whether technical optimizations are possible.

VII. INDUCTION AND ABDUCTION

The contribution of induction and abduction to deduction is the imaginative capacity. The *modus ponens* rule only produces what is already inscribed in the premises. In other words, it is not capable of anything new: it says what is already known. This is the essential condition to guarantee the maintenance of the syntactic coherence of a theory of L_p.

In L_c, syntactic coherence is guaranteed as soon as new information is associated with a new thought. This opens the doors to the imagination, making any new thought possible.

Abduction has been formalized by C.S. Peirce [24]. In logic, it can be expressed as follows:

If $a \rightarrow b$ is true and if b is true, then a is true

"*If I see something flying, I assume it is a bird because birds fly*". This is obviously a shortcut in reasoning.

Induction is identified and studied in logical approach since Aristotle. An example of application is the following:

If $a \rightarrow b$ is true and if $a \rightarrow c$ is true, then $b \rightarrow c$ is true

Aristotle showed that this is not correct and can lead to false conclusions: "*The donkey, the mule and the horse live long; they are animals without gall; therefore, animals without gall live long*".

In fact, any inductive or abductive production is potentially false. Therefore, the statistical tool (at the basis of Neuronal Artificial Intelligence) seems to be better suited to deal with this subject than the logical tool.

However, L_c allows to assume a result of the induction of the abduction, and to test its credibility.

"*Humans decide on provisional and fallible intuitions, even in logical matters*" – D. Battens [3]

Given its fallibilistic property, L_c can consider a thought to be true if it is not explicitly contradicted. The use of induction and abduction rules is therefore formally possible.

However, this can lead to completely absurd, even explosive, conclusions. For example: if b is true, then, whatever a, $a \rightarrow b$ is true, so a is true. This case can be dealt with by restricting the application of the rule to minimal clausal forms, but this is not the only pitfall. We will study this topic on another occasion.

VIII. THE ALGORITHMIC COMPLEXITY

We have not solved the conjecture $p=np$. The algorithms of L_c are those of L_p, thus of exponential complexity.

Note that our aim is not to model a formal mathematical language, but to capture human reasoning through a mathematical formalism. In this context, the question becomes: why does algorithmic complexity cause a problem in the use of a formal language in Artificial Intelligence?

A first need is to integrate the new piece of knowledge. If it generates an inconsistency with existing knowledge, L_p faces the problem of the epistemic rooting: what to choose, between the old and the new? A mix of the two?

The contextual postulate guarantees the maintenance of the syntactic consistency of the database, whatever the new thought integrated. This first need is therefore covered.

The second need is to be able to analyze the information, and its semantic impact. L_c must calculate inconsistent contexts, which requires the use of the algorithms of L_p.

The application of the semantic interpretation function on a few dozen formulae will therefore lead to a response time problem. But it is possible to take advantage of the properties of L_c to concentrate the semantic analysis on a few selected pieces of knowledge. The result may be incorrect.

However, this is what reasoning does all the time. Not a day goes by that we do not use conclusions based on incomplete assumptions, even though we are aware of all the information that would have allowed us to conclude correctly.

In the context of Artificial Intelligence, this second need is therefore covered if we accept the fact that human reasoning is imperfect: we need to define a function that selects, for a given stimulus, a subset of knowledges. The semantic interpretation is calculated only on this subset, not on the whole knowledge base.

To achieve this, we propose to exploit the results obtained in cognitive science. L_c is well suited to model human memory (D. Norman [15]). Indeed, the guarantee of syntactic consistency allows a safe distinction between long-term and short-term memory. This makes it possible to integrate into the algorithm cognitive thresholds from cognitive science research:

- the minimal change in beliefs between t_n and t_{n+1},
- the evolution criterion, which favors the use of the most recent or "primitive" thoughts,
- the technical incapacity threshold, which limits the number of simultaneous thoughts that can be used simultaneously,
- the semantic thresholds: J. Pitrat shows that a human is not able to reason on more than four levels of meta-knowledge [17],
- the threshold of proportionate reasoning: when confronted with a stimulus, the objective is not to perform the best theoretical analysis, but to reach a level of analysis sufficient to cause a reaction.

We see a three-level architecture emerging:

- a long-term memory (LTM), in which knowledge is stored as it arrives,
- a short-term memory (STM), fed by a function that selects knowledge from the long-term memory using cognitive thresholds,
- and a working memory (WM), which semantically interprets the content of the short-term memory.

This is a simple first approach, but it already gives some results that we find interesting in the context of human reasoning - for example, deducing, because we have just passed an airport, that the object in the distance in the sky is a plane (and not a bird). Note that we use a production rule by abduction to obtain this result.

We are currently enriching the algorithm to integrate these notions. Our objective is to be able to study its behavior when faced with practical use cases, and to verify that it reproduces (with all the limitations associated with this type of exercise) reasoning that is more or less identical to that which would be observed in a human being.

IX. EXAMPLE OF AN APPLICATION

After presenting the theoretical principles of contextual logic, we propose to develop an example of application to clarify our purpose, and to illustrate the knowledge modeling capabilities of L_c.

To do this, we use the example of the bird Tweety, a classic case study in the literature on non-monotonicity and belief revision. It would of course have been possible to do this with any other subject.

But underneath its apparent simplicity, the interest of the Tweety case is to be understandable by all, while gathering all the theoretical problems of non-monotonicity.

In this paragraph, we will describe in detail the different steps, taking the risk of making it sometimes a bit tedious to read.

Example *Consider the following knowledge, which we call the E_{NL} (for Natural Language) set:*

Birds and felines are animals [01 and 02]*. Birds are not felines* [03]*. Animals are generally diurnal* [04]*. Diurnal animals are not nocturnal* [05]*. Birds usually fly* [06]*. They are generally insectivorous* [07] *and gregarious* [08]*. Felines are carnivorous* [09] *and usually solitary* [10]*. Solitaires are not gregarious* [11]*. Insectivores are generally not carnivorous* [12]*.*

Swallows, sparrows, ostriches, and owls are bird [13, 14, 15 and 16]*. Swallows are not sparrows* [17]*, ostriches* [18]*, or owls* [19]*. Sparrows are not ostriches* [20] *or owls* [21]*. Ostriches are not owls* [22]*. Ostriches do not fly* [23]*. Owls are solitary* [24]*, nocturnal* [25]*, carnivorous* [26]*, and insectivorous* [27]*.*

Cats and lions are felines [28 and 29]*. Cats are not lions* [30]*. Cats are nocturnal* [31]*. Lions are gregarious* [32]*.*

Carnivores are hunters [33]*. Herbivores are prey for hunters* [34]*. Hunters generally attack prey* [35]*. If the prey is larger than the hunter, the latter does not attack* [36]*. Ostriches are larger than cats* [37] *and owls* [38]*.*

E_{NL} is deemed to be not exploitable in the syntax of L_p: it contains non-coherent and predicative information. It might be possible to model it using the default predicate logic, for example.

We have not attempted this. Our aim is not to compare the capabilities of the formalisms, but to show that the syntax of L_p is sufficient to model certain knowledge that was, until now, deemed to escape it.

We propose to translate this knowledge into the syntax of L_p by the following formulae.

01 02	*Bird* ∨ *Feline* → *Animal*
03	*Bird* → ¬ *Feline*
04	*Animal* → *Diurnal*
05	*Diurnal* → ¬ *Nocturnal*
06	*Bird* → *Flying*
07 08	*Bird* → (*Insectivore* ∧ *Gregarious*)
09 10	*Feline* → (*Carnivore* ∧ *Solitary*)
11	*Gregarious* → ¬ *Solitary*
12	*Insectivore* → ¬ *Carnivore*
13 14 15 16	(*Swallow* ∨ *Sparrow* ∨ *Ostrich* ∨ *Owl*) → *Bird*
17 18 19	*Swallow* → (¬ *Sparrow* ∧ ¬ *Ostrich* ∧ ¬ *Owl*)
20 21	*Sparrow* → (¬ *Ostrich* ∧ ¬ *Owl*)
22	*Ostrich* → ¬ *Owl*
23	*Ostrich* → ¬ *Flying*
24 25 26 27	*Owl* → (*Solitary* ∧ *Nocturnal* ∧ *Carnivore* ∧ *Insectivore*)
28 29	(*Cat* ∨ *Lion*) → *Feline*
30	*Cat* → ¬ *Lion*
31	*Cat* → *Nocturnal*
32	*Lion* → *Gregarious*
33	*Carnivore* → *Hunter*
34	*Herbivore* → (*Hunter* → *Prey*)
35	*Hunter* → (*Prey* → *Attack*)
36	*Hunter* → (*Prey* → (*Larger* → ¬ *Attack*))
37 38	*Ostrich* → ((*Cat* ∨ *Owl)* → *Larger*)

This wording models the knowledge expressed in E_{NL} in an "abrupt way". The next step is to transform this set into its normal form. We obtain:

01	*Bird → Animal*	
02	*Feline → Animal*	
03	*Bird → ¬ Feline*	
04	*Animal → Diurnal)*	*(em)*
05	*Diurnal → ¬ Nocturnal*	
06	*Bird → Flying*	*(em)*
07	*Bird → Insectivore*	*(em)*
08	*Bird → Gregarious*	*(em)*
09	*Feline → Carnivore*	
10	*Feline → Solitary*	*(em)*
11	*Gregarious → ¬ Solitary*	
12	*Insectivore → ¬ Carnivore*	*(em)*
13	*Swallow → Bird*	
14	*Sparrow → Bird*	
15	*Ostrich → Bird*	
16	*Owl → Bird*	
17	*Swallow → ¬ Sparrow*	
18	*Swallow → ¬ Ostrich*	
19	*Swallow → ¬ Owl*	
20	*Sparrow → ¬ Ostrich*	
21	*Sparrow → ¬ Owl*	
22	*Ostrich → ¬ Owl*	
23	*Ostrich → ¬ Flying*	
24	*Owl → Solitary*	
25	*Owl → Nocturnal*	
26	*Owl → Carnivore*	
27	*Owl → Insectivore*	
28	*Cat → Feline*	
29	*Lion → Feline*	
30	*Cat → ¬ Lion*	
31	*Cat → Nocturnal*	
32	*Lion → Gregarious*	
33	*Carnivore → Hunter*	
34	*Herbivore* → (*Hunter* → *Prey*)	*(pk)*
35	*Hunter* → (*Prey* → *Attack*)	*(em) and (pk)*
36	*Hunter* → (*Prey* → (*Larger* → ¬ *Attack*))	*(pk)*
37	*Ostrich* → (*Cat* → *Larger*)	*(pk)*
38	*Ostrich* → (*Owl* → *Larger*)	*(pk)*

This transformation has a linear complexity (P. Siegel [22]). We have introduced two new pieces of information, denoted by (*em*) and (*pk*). The explanations will come later.

For the time being, let us consider that the knowledge is entered in the order in which it appears, according to the following algorithm:

```
For each formula fi
    If fi has a subject, then PK
    Else
        Creating the thought ci0,2
        Creating the formula ci0,2 → fi
        If there is a contradiction, then EM
```

The sentences are processed one after the other.

01	$c_{010.2}$ → *(Bird → Animal)*
02	$c_{020.2}$ → *(Feline → Animal)*
03	$c_{030.2}$ → *(Bird → ¬ Feline)*
04	$c_{040.2}$ → *(Animal → Diurnal)*
05	$c_{050.2}$ → *(Diurnal → ¬ Nocturnal)*
06	$c_{060.2}$ → *(Bird → Flying)*
07	$c_{070.2}$ → *(Bird → Insectivore)*
08	$c_{080.2}$ → *(Bird → Gregarious)*
09	$c_{090.2}$ → *(Feline → Carnivore)*
10	$c_{100.2}$ → *(Feline → Solitary)*
11	$c_{110.2}$ → *(Gregarious → ¬ Solitary)*
12	$c_{120.2}$ → *(Insectivore → ¬ Carnivore)*
13	$c_{130.2}$ → *(Swallow → Bird)*
14	$c_{140.2}$ → *(Sparrow → Bird)*
15	$c_{150.2}$ → *(Ostrich → Bird)*
16	$c_{160.2}$ → *(Owl → Bird)*
17	$c_{170.2}$ → *(Swallow → ¬ Sparrow)*
18	$c_{180.2}$ → *(Swallow → ¬ Ostrich)*
19	$c_{190.2}$ → *(Swallow → ¬ Owl)*
20	$c_{200.2}$ → *(Sparrow → ¬ Ostrich)*
21	$c_{210.2}$ → *(Sparrow → ¬ Owl)*
22	$c_{220.2}$ → *(Ostrich → ¬ Owl)*
23	$c_{230.2}$ → *(Ostrich → ¬ Flying)*
24	$c_{240.2}$ → *(Owl → Solitary)*

For each sentence, the system asks if there is a subject, and the answer is no. The formula is integrated, and the system tested its semantic consistency. Until formula 23, there is no question. The formula 24 is then integrated. There are several possibilities:

- the STM (see paragraph VIII) can only contain a finite number of formulae, for example the last ten arrivals. The system does not detect any problem, and the process continues with the following sentences.
- the LTM has more capacity, or benefits from a more complex selection function (e.g., using the atomic propositions of the incoming formula to search the LTM for related sentences).
  A potential inconsistency is then detected: owls are birds [16], birds are gregarious [08], owls are solitary [24] and gregarious is not solitary [11]. So, owls do not exist, or they are solitary and not solitary.
  It then asks the instructor, who has two choices:
  - he does not know, or he considers it is normal: the semantic inconsistency is accepted, and the system moves on to the next step. Note that it is the maintenance of syntactic consistency that allows this,
  - it indicates that one of the pieces of knowledge involved in the inconsistency is true and false. In this case, birds are gregarious [08] is sometimes true and sometimes false. In this case, the system applies the EM module to the indicated formula.

The EM module is :

if f_i is of type *em*
  Creating the thoughts $c_{i1,3}$ et $c_{i2,3}$
  Creating the formula $c_{i1,3} \rightarrow c_{i0,2}$
  Creating the formula $c_{i2,3} \rightarrow \neg c_{i0,2}$

We have previously identified by (*em*) the formulae that are affected by this module. Let's continue the treatment. We obtain:

01 $c_{010.2}$ → *(Bird → Animal)*
02 $c_{020.2}$ → *(Feline → Animal)*
03 $c_{030.2}$ → *(Bird → ¬ Feline)*
04 $c_{040.2}$ → *(Animal → Diurnal)*
$c_{041.3} \rightarrow c_{040.2}$
$c_{042.3} \rightarrow \neg c_{040.2}$
05 $c_{050.2}$ → *(Diurnal → ¬ Nocturnal)*
06 $c_{060.2}$ → *(Bird → Flying)*
$c_{061.3} \rightarrow c_{060.2}$
$c_{062.3} \rightarrow c_{060.2}$
07 $c_{070.2}$ → *(Bird → Insectivore)*
$c_{071.3} \rightarrow c_{070.2}$
$c_{072.3} \rightarrow \neg c_{070.2}$
08 $c_{080.2}$ → *(Bird → Gregarious)*
$c_{081.3} \rightarrow c_{080.2}$
$c_{082.3} \rightarrow \neg c_{080.2}$
09 $c_{090.2}$ → *(Feline → Carnivore)*
10 $c_{100.2}$ → *(Feline → Solitary)*
$c_{101.3} \rightarrow c_{100.2}$
$c_{102.3} \rightarrow \neg c_{100.2}$
11 $c_{110.2}$ → *(Gregarious → ¬ Solitary)*
12 $c_{120.2}$ → *(Insectivore → ¬ Carnivore)*
$c_{121.3} \rightarrow c_{120.2}$
$c_{122.3} \rightarrow \neg c_{120.2}$
13 $c_{130.2}$ → *(Swallow → Bird)*
14 $c_{140.2}$ → *(Sparrow → Bird)*
15 $c_{150.2}$ → *(Ostrich → Bird)*
16 $c_{160.2}$ → *(Owl → Bird)*
17 $c_{170.2}$ → *(Swallow → ¬ Sparrow)*
18 $c_{180.2}$ → *(Swallow → ¬ Ostrich)*
19 $c_{190.2}$ → *(Swallow → ¬ Owl)*
20 $c_{200.2}$ → *(Sparrow → ¬ Ostrich)*
21 $c_{210.2}$ → *(Sparrow → ¬ Owl)*
22 $c_{220.2}$ → *(Ostrich → ¬ Owl)*
23 $c_{230.2}$ → *(Ostrich → ¬ Flying)*
24 $c_{240.2}$ → *(Owl → Solitary)*
25 $c_{250.2}$ → *(Owl → Nocturnal)*
26 $c_{260.2}$ → *(Owl → Carnivorous)*
27 $c_{270.2}$ → *(Owl → Insectivore)*
28 $c_{280.2}$ → *(Cat → Feline)*
29 $c_{290.2}$ → (*Lion* → *Feline*)
30 $c_{300.2}$ → (*Cat* → ¬ *Lion*)
31 $c_{310.2}$ → (*Cat* → *Nocturnal*)
32 $c_{320.2}$ → (*Lion* → *Gregarious*)
33 $c_{330.2}$ → (*Carnivore* → *Hunter*)

We come to the formula number 34: *Herbivore* → (*Hunter* → *Prey*). Before integration, the system asks if it contains a subject. The answer is yes, for *Herbivore*. The PK module is then applied:

if f_i is of type *pk*
  // f_i is a clause, so it is of type $g \rightarrow h$
  // with g is a conjunction of literals
  // and h is a disjunction of literals
  // g is indicated by the instructor
  Creating the thoughts $c_{i0,2}$ et $c_{i3,1}$
  Creating the formula $c_{i0,2} \rightarrow g \rightarrow c_{i3,1}$
  Creating the formula $c_{i3,1} \rightarrow h$

We have previously identified by (*pk*) the formulae that are affected by this module.

After application to the whole of E_{NL}, we obtain:

01 $c_{010.2}$ → *(Bird → Animal)*
02 $c_{020.2}$ → *(Feline → Animal)*
03 $c_{030.2}$ → *(Bird → ¬ Feline)*
04 $c_{040.2}$ → *(Animal → Diurnal)*
$c_{041.3} \rightarrow c_{040.2}$
$c_{042.3} \rightarrow \neg c_{040.2}$
05 $c_{050.2}$ → *(Diurnal → ¬ Nocturnal)*
06 $c_{060.2}$ → *(Bird → Flying)*
$c_{061.3} \rightarrow c_{060.2}$
$c_{062.3} \rightarrow c_{060.2}$
07 $c_{070.2}$ → *(Bird → Insectivore)*
$c_{071.3} \rightarrow c_{070.2}$
$c_{072.3} \rightarrow \neg c_{070.2}$
08 $c_{080.2}$ → *(Bird → Gregarious)*
$c_{081.3} \rightarrow c_{080.2}$
$c_{082.3} \rightarrow \neg c_{080.2}$
09 $c_{090.2}$ → *(Feline → Carnivore)*
10 $c_{100.2}$ → *(Feline → Solitary)*
$c_{101.3} \rightarrow c_{100.2}$
$c_{102.3} \rightarrow \neg c_{100.2}$
11 $c_{110.2}$ → *(Gregarious → ¬ Solitary)*
12 $c_{120.2}$ → *(Insectivore → ¬ Carnivore)*
$c_{121.3} \rightarrow c_{120.2}$
$c_{122.3} \rightarrow \neg c_{120.2}$
13 $c_{130.2}$ → *(Swallow → Bird)*
14 $c_{140.2}$ → *(Sparrow → Bird)*
15 $c_{150.2}$ → *(Ostrich → Bird)*
16 $c_{160.2}$ → *(Owl → Bird)*
17 $c_{170.2}$ → *(Swallow → ¬ Sparrow)*
18 $c_{180.2}$ → *(Swallow → ¬ Ostrich)*
19 $c_{190.2}$ → *(Swallow → ¬ Owl)*
20 $c_{200.2}$ → *(Sparrow → ¬ Ostrich)*
21 $c_{210.2}$ → *(Sparrow → ¬ Owl)*
22 $c_{220.2}$ → *(Ostrich → ¬ Owl)*
23 $c_{230.2}$ → *(Ostrich → ¬ Flying)*
24 $c_{240.2}$ → *(Owl → Solitary)*
25 $c_{250.2}$ → *(Owl → Nocturnal)*
26 $c_{260.2}$ → *(Owl → Carnivorous)*
27 $c_{270.2}$ → *(Owl → Insectivore)*
28 $c_{280.2}$ → *(Cat → Feline)*

29 $c_{290.2} \rightarrow (Lion \rightarrow Feline)$
30 $c_{300.2} \rightarrow (Cat \rightarrow \neg\ Lion)$
31 $c_{310.2} \rightarrow (Cat \rightarrow Nocturnal)$
32 $c_{320.2} \rightarrow (Lion \rightarrow Gregarious)$
33 $c_{330.2} \rightarrow (Carnivore \rightarrow Hunter)$
34 $c_{340.2} \rightarrow (Herbivore \rightarrow c_{343.1})$
$c_{343.1} \rightarrow (Hunter \rightarrow Prey)$
35 $c_{350.2} \rightarrow (Hunter \rightarrow c_{353.1})$
$c_{351.3} \rightarrow c_{350.2}$
$c_{352.3} \rightarrow \neg\ c_{350.2}$
$c_{353.1} \rightarrow (Prey \rightarrow Attack)$
36 $c_{360.2} \rightarrow (Hunter \rightarrow c_{363.1})$
$c_{363.1} \rightarrow (Prey \rightarrow (Larger \rightarrow \neg\ Attack))$
37 $c_{370.2} \rightarrow (Ostrich \rightarrow c_{373.1})$
$c_{373.1} \rightarrow (Cat \rightarrow Larger)$
38 $c_{380.2} \rightarrow (Ostrich \rightarrow c_{383.1})$
$c_{383.1} \rightarrow (Owl \rightarrow Larger)$

We have converted E_{NL} into a set E_{Lc} that is fully compatible with the syntax of propositional logic. It is L_p consistent. This process requires some comments:

- thoughts are indeed *silent* atomic propositions. They appear completely automatically in the syntax. The communication interface with the instructor does not see them, and only acts through the atomic propositions of L_p,
- the properties of L_c simplify the revision of beliefs: it is only done by adding the new pieces of knowledge, without ever modifying the knowledge already entered in the system.
  In case of error, it is possible to “cancel” a knowledge $c_{i,2} \rightarrow f$ by creating a thought $c_{ix,3}$, and integrating the formula $c_{ix,3} \rightarrow \neg\, c_{i,2}$. The combinatorics on the maximal contexts of T_m means that $c_{i,2}$ is not retained in the epistemic contexts.
- *Hunter* → (*Prey* → *Attack*) (formula number 35) uses simultaneously a predicate and an epistemic modality: it is false if the prey is large (formula number 36).

To obtain epistemic contexts is very tedious, so we have computerized the process to overcome this difficulty. To do this, we use a classical propositional logic solver (to calculate the minimum inconsistent contexts) and a classical combinatorial algorithm (using the minimum inconsistent contexts to calculate the epistemic perspectives).

Here are some examples obtained by applying general contextual semantic interpretation function (we suppose the STM is the LTM, i.e., paragraph VIII):

- if the stimulus is *Bird* or *Swallow*: birds (or swallows) are animals, diurnal, gregarious, insectivore, and fly,
- if the stimulus is *Ostrich*: ostriches are birds, insectivore, gregarious, diurnal, and do not fly,
- if the stimulus is *Owl*: owls are birds, carnivorous, solitary, nocturnal, and fly,
- if the stimulus is {*Cat*, *Sparrow*}: the sparrow is attacked. We use the general contextual semantic.
  With the stimulus *Cat*, we obtain all properties of cats. It is a hunter; it is not a sparrow; etc. And if it meets a prey it attacks.
  With *Sparrow*, we obtain all properties of Sparrow. It is a prey, and it is not à cat.
  Finally, we use generalized contextual semantics to cross the perspectives of *Cat* and *Sparrow*. We obtain a context that verifies the predicate $c_{353,1}$, and associates this predicate with the hunting cat through the thought of $c_{350,2}$.
- if the stimulus is {*Cat*, *Owl*, *Sparrow*, *Ostrich*}: the sparrow is in a bad way, but the ostrich can go about its business.
  To obtain the set of knowledges from this stimulus, we need to interpret the relevant models of {*Cat*}, {*Owl*}, {*Sparrow*} and {*Ostrich*} separately, then the pairs {*Cat*, *Owl*}, {*Cat*, *Sparrow*}, {*Cat*, *Ostrich*}, {*Owl*, *Sparrow*}, {*Owl*, *Ostrich*} and {*Sparrow*, *Ostrich*} then the triplets, etc. The order in which the proposals are processed is not important,
- and if the stimulus is {*Lion*, *Ostrich*}, the ostrich would have some reason to be worried.

X. CONCLUSION

By applying contextual postulate to propositional logic, we obtain a formal language which, while strictly respecting the syntax of L_p, allows to model the notions of incoherence and predicate. L_c's contribution to the family of non-monotonic logics is to propose a fallibilist and perspectivist formalism.

This result echoes a fundamental question: is human reasoning fallibilist and perspectivist? It is still open (see H. Albert [1] and W. Quine [19] for example).

If the answer is no, contextual logic is probably to be studied as a mathematical curiosity given its specific properties. If the answer is yes, the contextual postulate offers an alternative to Symbolic Artificial Intelligence.

<u>Some points of clarification</u>

We have certainly not answered all the questions that this work raises. But there are also some topics that we have deliberately skimmed over so as not to make our remarks totally indigestible. We list them below, in a thought-provoking format for interested readers.

1) We indicate that we associate a new thought with each clause of the normal form of the theory. This has an impact on the semantic interpretation: for the same stimulus, the interpretation on the set $\{c_1 \rightarrow f, c_2 \rightarrow g\}$ can be different from the one that would be obtained on the set $\{c \rightarrow f \wedge g\}$.

We have no theoretical argument to justify the choice we have made but only a pragmatic explanation: the conversion of a set of formulae into its minimal normal form is achieved by a linear algorithm, and each clause identifies a unique formula and therefore a unique thought.

2) Concerning the semantic interpretation function, we propose the definition of epistemic contexts to identify the relevant sets of thoughts. Other definitions are possible. We have retained the one that gave us the closest results to human reasoning on the different tests we carried out.

Another approach, perhaps more purist, would have consisted in giving the different possible definitions and comparing their mathematical properties. There is a lot of methodical work to be done here.

3) Another point is about the concept of universal connector. For example, the difference between predicate logic and contextual logic is that the latter has no universal quantifier. In fact, L_c natively models a form of quantifier. For example, assume a knowledge base consisting of a single piece of information: $\{c \rightarrow Bird \rightarrow Flying\}$. This says that *Birds fly*. Given this piece of knowledge, it would be the same to say *All birds fly*.

In L_c, the notion of universality is not carried by a syntactic quantifier. It is deduced from the semantic interpretation: a state is universal as long as it is not explicitly contradicted.

The same applies to the modal connector of necessity: in a fallibilist approach, since nothing is certain, nothing can be written as universal in the syntax. This notion can only fall within the scope of semantic interpretation.

4) As far as natural languages are concerned, the contextual postulate reinterprets their relationship with formal languages. Contrary to the most common opinion, our intuition is that a formal language is a language "like any other" as soon as it is used in its contextual form.

In effect, the postulate repositions the medium of expression as the primary goal.

The ability to reason takes a back seat, as an opportunity effect of the properties of the language. We are currently studying the work of J. Piaget [16] to challenge this intuition – and to move automatically from E_{Lc} to E_{NL}.

5) Contextual logic introduces the notion of stimulus. We use it in the different examples as an imposed external event. It generates a need for the system to react, and the semantic interpretation focuses on identifying the perspectives that verify the stimulus.

It can also be used to analyze a hypothesis. In this case, the stimulus does not have to be in a form f, but in a form $c \rightarrow f$. The semantic interpretation will then look for the different perspectives that verify or not verify c. Note that not checking c does not prejudge the truth value of f.

6) In the example developed in paragraph XI, we have only brought in the notion of subject from formula 34. In absolute terms, this should have been done from the first sentence. We could have covered the different possible answers by extending the list of formulae. But this would have made the writing unreadable, without any contribution to what we want to demonstrate.

In fact, this remark is not insignificant. It raises some questions as: are we saying the same thing with $a \rightarrow b$ and $\neg b \rightarrow \neg a$? The question is about the meaning of connectors. It has been extensively studied, notably by J. Lukaszewicz [13].

Without elaborating on the subject, let us point out to readers interested in this topic that L_c proposes an answer. $c \rightarrow f$ says that if c is true then f is true. But if c is false, f can be true or false: the fact that a thought is false does not imply that the expression that describes it is false in the sense of syntactic interpretation.

7) We use some results from cognitive science to get around the problem of algorithmic complexity. In parallel, we are working on a technical solution that uses the properties of Horn clauses (A. Horn [9], used for example by A. Colmerauer and his teams [8] for the Prolog programming language).

We have indicated that we transform the formulae to their normal form, and then consider the individual clauses. Each has a form $f \rightarrow g$, for f a conjunction of positive literals and g a disjunction of positive literals. We propose to decompose $f \rightarrow g$ by a set of formulae $f \rightarrow a_i$, for a_i each literal of g.

Thus, a formula like $a \rightarrow (b \vee c)$ is expressed by two distinct knowledges $c_1 \rightarrow (a \rightarrow b)$ and $c_2 \rightarrow (a \rightarrow c)$. The sense of this operation is:

If a is true and if $a \rightarrow (b \vee c)$ is true, then b and c are true or conceivable unless explicitly contradicted.

It is a weakened *modus ponens* rule. Its use as it stands leads to the equivalent of the explosion principle. Its formal drafting therefore requires further study, limiting for example the application of this rule to minimal clausal forms. We analyze the level of loss of expression that this induces to evaluate its semantic relevance in the context of human reasoning.

Its technical contribution would be considerable: at the cost of a linear multiplication of the size of the theory, we would benefit from a resolution algorithm of polynomial complexity – *versus* the exponential complexity of the original algorithms of L_p.

REFERENCES

[1] H. Albert, "Traktat über kritische Vernunft", Utb Fuer Wissenschaft, 1968

[2] D. Andler, "Introduction aux sciences cognitives", Gallimard, 2004

[3] D. Batens, "Une caractérisation générale des logiques adaptatives", Logique & Analyse 173–174–175, 2001

[4] R. Descartes, "Méditations métaphysiques. Objections et Réponses (I à VI)", édition sous la direction de Jean-Marie Beyssade et de Denis Kambouchner, éd. Gallimard Tel (2018), 1641

[5] J. Fodor, "The Language of Thought", Harvard University Press, 1975

[6] G. Frege, "Grundgesetze der Arithmetik", 1903

[7] P. Gardenfors, D. Makinson, "Relations between the logic of theory change and nonmonotonic logic", in André Fuhrmann and Michael Morreau, eds. The Logic of Theory Change, LNAI-465, p. 185-205, 1991

[8] F. Giannesini, H. Kanoui, R. Pasero, M. Van Caneghem, préface d'A. Colmerauer, "Prolog", Interéditions, 1985

[9] A. Horn, "On sentences which are true of direct unions of algebras", in Journal of Symbolic Logic, number 16, pages 14-21, 1951

[10] F. Kaulbach, "Philosophie des Perspektivismus: Wahrheit und Perspektive bei Kant, Hegel und Nietzsche", Mohr Siebeck, Tübingen, 1990

[11] A. Kohler, "Proposition d'une structure de représentation de la connaissance pour les raisonnements non classiques", Université d'Aix-Marseille I, France, thèse de doctorat en informatique, 1995

[12] S.A. Kripke, Saul Aaron, "Semantical considerations on modal logic", Reference and Modality, L. linsky (éditeur), Oxford University Press, London, 1971

[13] J. Lukaszewicz, "Many-Valued Systems of Propositional Logic", eds. S. Mc Call, polish Logic, Oxford University Press, Oxford, 1967

[14] J. McCarthy, "Circumscription - A Form of Non-Monotonic Reasonnig", Artificial Intelligence 13, pp. 27-39, 1980

[15] D. Norman, "Models of human memory", Academic Press, New York, 1970

[16] J. Piaget, R. Garcia, et Al., "Vers une logique des significations", Genève: Murionde, 1987

[17] J. Pitrat, "Méta-connaissance, Futur de l'Intelligence Artificielle", Hermes, 1990

[18] K. Popper, "The Logic of Scientific Discovery", Routledge, coll. Routledge Classics, 1934

[19] W. Quine, "From a logical point of view", Harvard Univiersity Press, Cambridge, Massachussets and London, England, 1980

[20] R. Reiter, "A logic for default reasoning", Artificial Intelligence 13, 1980

[21] D.E. Rumelhart, G.E. Hinton, R.J. Williams, "Learning Internal Representations by Error Propagation", in D.E Rumelhart and J.

McClelland (eds.), Parallel Distributed Processing - Explorations in the Microstructure of Cognition, Cambridge, MA:MIT Press, 1986

[22] P. Siegel, ”Représentation et utilisation de la connaissance en calcul propositionnel”, Université de Luminy, Thèse d’état, 1987

[23] A. Thayse et co-auteurs, ”Approche logique de l’Intelligence Artificielle”, DUNOD, 1990

[24] C. Tiercelin, “C. S. Peirce et le pragmatisme”, Paris, puf, p. 96, 1993

[25] L. Wittgenstein, “Tractatus logico-philosophicus”, traduction Gilles Gaston Granger, éd. Gallimard Tel (1993), 1921