Cognitive Binary Logic - The Natural Unified Formal Theory of Propositional Binary Logic

NICOLAIE POPESCU-BODORIN, *Member*, *IEEE*
Spiru Haret University
Dept. of Mathematics and Computer Science
13 Ion Ghica, Bucharest 3
ROMANIA
http://fmi.spiruharet.ro/bodorin/

LUMINIŢA STATE
University of Piteşti
Dept. of Mathematics and Computer Science
1 Targu din Vale, Piteşti, Argeş
ROMANIA
lstate@clicknet.ro

Abstract: This paper presents a formal theory which describes propositional binary logic as a semantically closed formal language, and allows for syntactically and semantically well-formed formulae, formal proofs (demonstrability in Hilbertian acception), deduction (Gentzen's view of demonstrability), CNF-ization, and deconstruction to be expressed and tested in the same (computational) formal language, using the same data structure. It is also shown here that *Cognitive Binary Logic* is a self-described theory in which the *Liar Paradox* is deconstructed.

Key–Words: cognitive binary logic, inductive/deductive discourse, liar paradox, computational logic

1 Introduction

This paper presents a unified natural approach to propositional binary logic, in which syntactically and semantically well-formed formulae, formal proofs (demonstrability in Hilbertian acception), deduction (Gentzen's view of demonstrability, [1]), CNF-ization [2], and deconstruction are expressed and tested in the same (computational) formal language - Computational Cognitive Binary Logic ($CCBL$), using the same data structure: the deductive discourse.

It is also shown here that *Cognitive Binary Logic* is a self-described theory in which the *Liar Paradox* [3] is deconstructed.

The prerequisites of this paper are the following concepts: *formal language* [4], *formal theory* (formal system) [5], *formal proof* [6], *deduction* [1], *resolution* [2], *valid argumentation* [6], *Gentzen's natural deduction system* (sequent calculus) [1], *completeness* [6], *consistency* (soundness) [6] - all of them considered in the context of propositional binary logic, *equivalence relations*, *modal logic* [9], *Lukasiewicz's* [7] and *Hilbert's* [8] classical formalizations of binary logic. Since they are so many and the space here is limited, we would like to refer to bibliographic sources instead of reproducing some redundant contents.

Despite the vast universe of discourse, this paper is based on a single assumption which we all know to be true: both Lukasiewicz's and Hilbert's formalizations of propositional binary logic are complete and sound formal theories.

1.1 Outline

The structure of this paper follows an imaginary process of reverse engineering the Propositional Binary Logic (BPL). We start by analyzing how it works in order to see what is it made of.

Section 2 discuss the prerequisites of the proposed formal theory. Cognitive implication is introduced here as a natural way of modelling human logical thinking as opposite to the trivial sintactic truth *ex contradictione quodlibet*.

A dual Semantic-Computational formalization of Cognitive Binary Logic ($SCBL, CCBL$) is presented in the third section of the paper. It is shown here that both $SCBL$ and $CCBL$ theories cover BPL in a complete and consistent manner. Also, in $CCBL$ the inductive/deductive proof of any theorem is algoritmically computable. In the end, it is shown that even if $CCBL$ theory qualify the language of PBL as being semantically closed, the Liar Paradox is still successfully deconstructed.

2 Setting the semantic framework

From the beginning we must say that present computers can only parse and evaluate data structures that are truly meaningful only in human understanding.

Hence, our world is a semantic one, while the computational world is defined, regulated, controlled and described through syntactic rules. But even when we think about syntactic rules, we still operate at the semantic level for the simple reason that symbols themselves are not very important to us, while their meaning is our actual concern.

2.1 Formalization

The symbols become important only when it comes to formulate our semantic knowledge into something computable by expressing our understanding into a computational language. In case of binary logic, the challenge is to translate human reasoning into something a little bit more sophisticated than (two-element) Boolean algebra while achieving a certain degree of semantic fidelity measured in terms of similitude or analogy with human thinking. This is what we usually call *formalization* and the result is a *formal theory* defined over a *formal language*.

Since we cannot formalize something unless we are fully aware of its meanings, a formalization will never come easy because it requires a triple effort.

The first step is to understand the world that is to be formalized, and this is what we do here, throughout the second section.

The second step is to associate semantic charges (the meanings) to the assemblies of symbols belonging to a pure syntactic world while setting it up.

Last, but not least, we must test the semantic efficiency of the newly created formal system by verifying its limits in terms of completeness and consistency.

The second step is particularly difficult, and probably this is the reason why the syntactic rules (the ways of constructing well-formed formulae) and the reasoning rules (the ways of constructing well-formed proofs) are very different in the present formal theories of the propositional binary logic.

2.2 Implication and human reasoning

Let us comment on a well-known example: *I think, therefore I am*. It is inevitably true that I am, since I know that I think and, on the other hand, I also know that if I did not exist, then certainly I would not think.

Now, let us consider that p and q are propositional variables with q being a tautology and p being logically false. Hence, $a = (p \rightarrow q)$ is true but still, in human understanding a is not a valid argument for q. Syntactically, an implication can be parsed and evaluated as true, but this does not necessarily qualify that implication as a valid argument.

It can be seen in the above examples that a valid argument in human logical thinking is a true implication with true premise. Humans think semantically, not syntactically, by using some implications as valid arguments. By the way, nobody has said: *I don't doubt or I think, I don't think or I am*.

2.3 One more challenge

We should ask ourselves why the following truth is told surprisingly rarely: *the natural logical framework for analyzing the formulae of propositional binary logic language is modal logic* [9].

The above assertion is true because any formula within the language falls into one of the following three categories: formulae which are *always true* (tautologies), formulae which are *sometimes true* (and sometimes false) and formulae which are *always false*. By reformulating in classical terminology of modal logic, a formula could be: *necessary* true (an axiom or a theorem, or, generally speaking, a tautology) or *possibly* true (a contingent, or a formula demonstrable under some hypotheses which give *the context* of that *satisfiable* truth) or *impossibly* true (i.e. *necessary* false, or contradiction).

If $FORM$ denotes the set of formulae within the language of propositional binary logic, then: $FORM = \hat{f} \cup \hat{c}_t \cup \hat{t}$, where $\hat{f}$ is the class of necessary false formulae, $\hat{c}_t$ is the class of contextual truths (or the class of proper variables whose truth values are not constant) and $\hat{t}$ is the class of all tautologies.

Hence, one more challenge here is to put this three-valued state of truth in the binary framework of the propositional binary logic and to explain what makes this possible.

2.4 Cognitive implication

A cognitive implication is assumed here to be a formula of the following type:

$$\left(\bigwedge_{i=1}^{n} h_i\right) \rightarrow \left(\bigvee_{i=1}^{m} c_j\right), \tag{1}$$

where $\{h_i\}_{1 \leq i \leq n}$ and $\{c_j\}_{1 \leq j \leq m}$ are two sets of formulae (hypotheses and conclusions, respectively).

The name is suggested by the fact that investigating a universe is a matter of finding a suitable collection of hypotheses assumed to be simultaneously true (hence, their conjuntion is true) and studying the possibilities to infer some conclusions which are not necessarily simultaneously true. To demonstrate how suitable this structure is for describing human cognitive processes, we give the following example: we want to build a formal theory over a formal language. But what does that mean exactly? Nothing more than identifying a set of simoultaneously true premises (axioms and argumentation rules) that enable us to prove or to disprove well-formed sentences (formulae) written in that language.

2.5 Implication as a SAT problem

A trivial cognitive implication in propositional binary logic is that no matter the hypothesis h, the *tertium non datur* is always true and so it is the implication:

$$h \rightarrow (a \vee \neg a)\,. \tag{2}$$

Nothing changes if *tertium non datur* is replaced by any other tautology t:

$$h \rightarrow t, \tag{3}$$

Also, no matter the conclusion c, if the hypotesis is a contradiction, the following implication is true:

$$f \rightarrow c. \tag{4}$$

Hence, there is no doubt that formulae (2)-(4) are trivial Boolean satisfiability (SAT) problems.

Since formula (4) is by default a tautology, it follows that *there is only one type of implication that deserves to be studied in binary logic*:

$$s_t \rightarrow c, \tag{5}$$

where $s_t \in \hat{s}_t$ and:

$$\hat{s}_t = \hat{c}_t \cup \hat{t}, \tag{6}$$

2.6 SAT problem vs. cognitive implication

Let $a \in FORM = \hat{f} \cup \hat{c}_t \cup \hat{t}$ an arbitrary formula. Its corresponding SAT problem can be stated as a decision problem as follows:

$$(a \in \hat{t}) \oplus (a \in \hat{c}_t) \oplus (a \in \hat{f}), \tag{7}$$

where $\oplus$ denotes exclusive disjunction. On the other hand, $\forall (a, f, t) \in FORM \times \hat{f} \times \hat{t}$:

$$[(t \rightarrow (a \vee f)) \leftrightarrow a] \in \hat{t} \tag{8}$$

In other words, the formulae (7), (8) tell us that the demonstration of *the truth* or of a *contextual truth*, and also the *resolution* of a contextual truth (the process of finding a context which makes a formula satisfiable), all of them can be carried out by parsing the same data structure, which is syntactically a cognitive implication and, semantically, is an attemp to find a valid argument for that truth.

Also, all three modal states of truth $(\hat{t}, \hat{f}, \hat{c_t})$ are decidable in propositional binary logic using the same test: a cognitive implication (8).

3 Cognitive Binary Logic

A dual formalization of Cognitive Binary Logic (CBL) is defined here as a pair of formal theories given over the same language (propositional binary logic), using the same set of axioms (*tertium non datur*) and two different sets of valid arguments which still produce the same set of theorems.

3.1 The axiom

There is a single axiom in CBL: *tertium non datur*, considered in the classical form:

$$a \vee \neg a, \tag{9}$$

or in the simplest form of a cognitive implication:

$$a \rightarrow a, \tag{10}$$

or in a more general form:

$$\left[a \wedge \left(\bigwedge_{i=1}^{n} h_i\right)\right] \rightarrow \left[a \vee \left(\bigvee_{i=1}^{m} c_j\right)\right]. \tag{11}$$

where the meanings of symbols appearing in (9)-(11) are already introduced in the previous subsections.

3.2 Semantic formalization of CBL

The first set of valid arguments used here to define a first formalization of CBL is meaningful in human understanding. It is further denoted as $\mathcal{A}_1$ and contains the following equivalences:

1. The law of double negation:

$$a \Leftrightarrow \neg\neg a. \tag{12}$$

2. The law of the contrapositive:

$$(a \rightarrow b) \Leftrightarrow (\neg b \rightarrow \neg a)\,. \tag{13}$$

3. The law of deduction-resolution:

$$[(h \wedge a) \rightarrow b] \Leftrightarrow [h \rightarrow (a \rightarrow b)]\,. \tag{14}$$

4. Distributivity law:

$$\{h \rightarrow [c \vee (a \wedge b)]\} \Leftrightarrow$$

$$\Leftrightarrow \{[h \rightarrow (c \vee a)] \wedge [h \rightarrow (c \vee b)]\}\,. \tag{15}$$

Let us denote the semantic formalization of the CBL as being the triplet formed with the language of propositional logic (L), *tertium non datur* axiom (TND) and the set of valid arguments $\mathcal{A}_1$ from above:

$$SCBL = \{L, TND, \mathcal{A}_1\}. \tag{16}$$

Let LBL be Lukasiewicz's classical formalization of propositional binary logic:

$$LBL = \{L, Axiom, MP\}, \tag{17}$$

where L denotes the formal language of propositional binary logic, MP stands for *Modus Ponens*, and $Axiom$ is the following set of axioms:

$$p \rightarrow (q \rightarrow p), \tag{18}$$

$$[p \rightarrow (q \rightarrow r)] \rightarrow [(p \rightarrow q) \rightarrow (p \rightarrow r)], \tag{19}$$

$$(\neg p \rightarrow \neg q) \rightarrow (q \rightarrow p). \tag{20}$$

Since TND and the formulae of $\mathcal{A}_1$ are theorems in LBL, it is clear that all theorems of $SCBL$ will be theorems in LBL, i.e. $SCBL$ is a sub-theory of LBL, fact that will be denoted as:

$$SCBL \subset LBL. \tag{21}$$

$SCBL$ is an inductive (Hilbertian) theory. Writting a formal proof in $SCBL$ is a matter of experience and intuition. We 'guess' what axioms and what substitutions we must use in order to prove a theorem. This approach is not suitable for algorithmic computation because from this point of view, 'guessing' means an exhaustive search in an infinite set of axiom instances. Still, the formulae within the second set of valid arguments ($\mathcal{A}_2$, given below) are very easy to prove using $SCBL$ formalization.

In the computational formalization of the CBL, we aim to make any formal proof algorithmically computable. The general ideea is to use formula 8, i.e. the fact that any formula a is equivalent to a cognitive implication $(t \rightarrow (a \vee f))$, which is further decomposable as a conjunction of cognitive implications. Therefore, the set of valid arguments will be appropriate for this purpose, more redundant, and less intelligible at a first glance.

3.3 Computational formalization of CBL

Computational formalization of Cognitive Binary Logic ($CCBL$) is defined by a second set of valid arguments, denoted $\mathcal{A}_2$, any argument being an equivalence and also a rule of syntactic and semantic simplification/complexification (elimination/introduction rule for a logical connective):

1. First distributivity law (Right-side conjunction elimination / introduction rule):

$$\{h \rightarrow [c \vee (\alpha \wedge \beta)]\} \Leftrightarrow$$

$$\Leftrightarrow \{[h \rightarrow (c \vee \alpha)] \wedge [h \rightarrow (c \vee \beta)]\}. \tag{22}$$

2. Second distributivity law (Left-side disjunction elimination / introduction rule):

$$\{[h \wedge (\alpha \vee \beta)] \rightarrow c\} \Leftrightarrow$$

$$\Leftrightarrow \{[(h \wedge \alpha) \rightarrow c] \wedge [(h \wedge \beta) \rightarrow c]\}. \tag{23}$$

3. First reformulation of deduction-resolution law (Left-side negation elimination / introduction rule):

$$\{[h \wedge (\neg\alpha)] \rightarrow c\} \Leftrightarrow \{h \rightarrow [c \vee \alpha]\}. \tag{24}$$

4. Second reformulation of deduction-resolution law (Right-side negation elimination / introduction rule):

$$\{h \rightarrow [c \vee (\neg\alpha)]\} \Leftrightarrow \{[h \wedge \alpha] \rightarrow c\}. \tag{25}$$

5. Third reformulation of deduction-resolution law (Left-side implication elimination / introduction rule):

$$\{[h \wedge (\alpha \rightarrow \beta)] \rightarrow c\} \Leftrightarrow$$

$$\Leftrightarrow \{[(h \wedge \beta) \rightarrow c] \wedge [h \rightarrow (c \vee \alpha)]\}. \tag{26}$$

6. Generalized deduction-resolution law (Right-side implication elimination / introduction rule):

$$\{h \rightarrow [c \vee (\alpha \rightarrow \beta)]\} \Leftrightarrow$$

$$\Leftrightarrow \{[h \wedge \alpha] \rightarrow [c \vee \beta]\}. \tag{27}$$

The formulae within $\mathcal{A}_2$ are theorems of $SCBL$, hence we get:

$$CCBL \subset SCBL \subset LBL. \tag{28}$$

Definition 1 *CCBL Formal Theory (N. Popescu-Bodorin):*
Let $L = (B, V, C, S, G_L, FORM)$ be the classical formal language of propositional binary logic, and let $LBL = \{L, Axiom, MP\}$ be the Lukasiewicz formalization of propositional binary logic, where:

- *$B = \{0, 1\}$ is the set of binary truth values;*
- *V is the set of propositional variables;*

- *C is the collection of logical connectives, $C = \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$, which are all defined using truth tables (logical gates);*
- *$S = \{(,)\}$ is the list of separators;*
- *G_L is the formal grammar which qualifies the strings from $V \cup C \cup S$ as well-formed formulae (the elements of $FORM$, i.e. syntactically valid sentences) through recursive structural complexification with logical connectives from C and symbols (propositional variables) from V.*
- *$FORM = \hat{t} \cup \hat{c}_t \cup \hat{f}$ is the set of all well-formed formulae classified by their modal state of truth: tautologies, contextual truths and contradictions.*

Let $\mathcal{L} = (\mathcal{B}, \mathcal{R}, \mathcal{V}, C, S, \mathcal{G}_{\mathcal{L}}, CI, \mathcal{D}, D)$ be the formal language of $CCBL$, where:

$$CCBL = \{\mathcal{L}, TND, \mathcal{A}_2\}, \tag{29}$$

and:

- *$\mathcal{A}_2$ is the set of valid arguments (22)-(27);*
- *TND stands for tertium non datur (9)-(11);*
- *$\mathcal{B} = \{\hat{t}, \hat{c}_t, \hat{f}\}$, i.e. the discourse in $\mathcal{L}$ is focused on tautologies, contextual truths and contradictions;*
- *$\mathcal{R} = \{t, c_t, f\}$ is the list of reserved symbols used to refer a generic variable from $\hat{t}$, $\hat{c}_t$ and $\hat{f}$, respectively;*
- *$\mathcal{V} = FORM$, i.e. the discourse in $\mathcal{L}$ is a meta-discourse on L asserting something about the formulae within L;*
- *$\mathcal{G}_{\mathcal{L}}$ is the formal grammar that qualifies the cognitive implications (units of discourse, like simple sentences in natural language) as the simplest well-formed formulae within $\mathcal{L}$ using a single rule of immersion of $FORM$ in CI:*

$$\{\alpha \in FORM\} \Leftrightarrow$$

$$\Leftrightarrow \{\alpha_{ci} = [t \rightarrow (\alpha \vee f)] \in CI\}. \tag{30}$$

$\mathcal{G}_{\mathcal{L}}$ also recursively qualifies more complex structure of discourse (like complex sentences in natural language) as being well-formed by using six rules of phrasal grammar, namely formulae (22)-(27) from the set $\mathcal{A}_2$;

- *CI contains the simplest formulae of the language which are cognitive implications and conjunctions of cognitive implications;*
- *The full deductive discourse about a cognitive implication α_{ci} within $\mathcal{L}$ is defined as a tree having the following properties:*
  - The root is labeled with α_{ci}, or else, α is the label of the root and the expansion rule to be applied is the immersion law (30) and, consequently, the outer degree of the root is 1 and the only descendant of the root is labeled with α_{ci};
  - Starting with the vertex labeled with α_{ci}, tree expansion is conducted by the rules of $\mathcal{A}_2$ (applied from the left to the right), up to the point where they are no longer applicable;
  Any partial expansion is simply called a deductive discourse about α_{ci}. We denote by $\mathcal{D}$, the set of all possibles deductive discourses about the formulae within $\mathcal{V} = FORM$.
- *The summary of a full deductive discourse (with n leaves) expanded for a formula $a \in FORM$ can be written as:*

$$a \leftrightarrow \bigwedge_{i=1}^{n} \left(t \rightarrow \bigvee_{j=1}^{m} c_i^j \right), \tag{31}$$

where the variables c_i^j may contain negation but no other connective. Let D be the set of all possible summaries of this kind.

- *If α and α_{ci} are defined by (30), then $\alpha \Leftrightarrow \alpha_{ci}$. Hence, it is true that a deductive discourse about α_{ci} is also a deductive discourse about α;*

3.4 Inductive (Hilbertian) and deductive (Gentzen) discourse in CCBL

This section aims to show the analogy between deductive/inductive thinking and deductive/inductive logical computing in $CCBL$.

Let us consider an axiom of Lukasiewicz's formalization of propositional binary logic:

$$\alpha = \{[p \rightarrow (q \rightarrow r)] \rightarrow [(p \rightarrow q) \rightarrow (p \rightarrow r)]\}, \tag{32}$$

Human/Computational deductive discourse about α try to equivalate α with (to 'decompose' α as) a conjunction of simpler formulae with reduced structural complexity, up to the point where any component of this conjunction can be easily proved, or tested, or reduced to a true hypothesis. An example is given in Fig.1. The deductive discourse presented there is not a *full deductive discourse* because it still contains an expandable (a non-closed) leaf:

$$\{[t \wedge (p \rightarrow |q \rightarrow r|) \wedge \mathbf{p}] \rightarrow (\mathbf{p} \vee r \vee f)\}$$

on which the expansion rule (26) is still applicable. All of the other leaves are closed (non-expandable). All leaves are instances of TND axiom (see formulae 9-11).

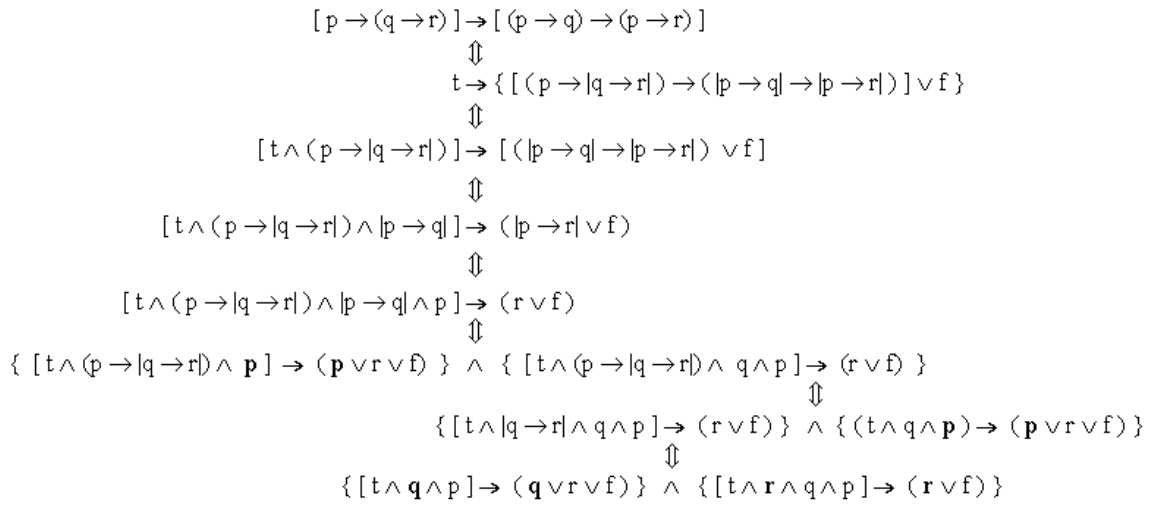

Figure 1: A deductive discourse for one of Lukasiewicz's axioms. From top to buttom, the following expansion rules (valid arguments) are applied: 8, 27, 27, 27, 26, 26, 26. Bold-face is used to mark instances of TND.

Definition 2 *Elements related to the deductive discourse in $CCBL$:*

- *A node within a deductive discourse is a terminal node / a closed node/ a closed leaf if the expansion rules of $\mathcal{A}_2$ are no longer applicable for that node, or else, it is an expandable node (a continuation/non-terminal point of the deductive discourse).*
- *A deductive discourse is full if all of its leaves are closed.*
- *A deductive proof in $\mathcal{L}$ (a demonstration in CCBL) is a full deductive discourse in $\mathcal{L}$ in which all leaves are instances of TND.*
- *A formula α (within L or $\mathcal{L}$) is said to be deductively provable (or, simply demonstrable in $CCBL$) if there is a deductive proof for α.*
- *A deconstruction in $\mathcal{L}$ is a full deductive discourse in $\mathcal{L}$ in which some leaves are logical but mutually irreconcilable, or some leaves are non-logical.*
- *An illegal syntax or a logical nonsense (in L, $\mathcal{L}$, LBL, or $CCBL$) is a string whose full deductive discourse is a deconstruction.*
- *A contextual truth (in L, $\mathcal{L}$, LBL, or $CCBL$) is a well-formed formula whose full deductive discourse is neither a deductive proof, nor a deconstruction.*

At a first glance, the expression '*closed leaf*' used above does not seem to be necessary, but in fact, it is mandatory, because in any partial expansion of a tree there are always current leaves. Some of them are expandable (becoming non-terminal nodes), or else, the expansion is full, not partial. The quality of being closed is not a geographic/pictographic attribute, but a semantic one.

A deductive proof of a formula is a failed deconstruction attempted for that formula which, in the end, instead of being deconstructed, proves to be a conjunction of true formulae (axiom instances). In a genetic view, a deductive proof of a formula shows that the formula carries only the logical genes of the truth, while a successful deconstruction reveals irreconcilable and contradictory meanings (logical genes of the false) and possibly non-logical genes of the given 'formula' (see the *Liar Paradox* example in section 3.9). But generally speaking, the deductive discourse is neutral: it could be a deductive proof, or a successful logical deconstruction, or it could be none of them. The deductive discourses of the contextual truths fall into the latter category: their genes (the leaves) are reconcilable under certain hypotheses.

A demonstration in Hilbertian acception is given by what we call a *formal proof*, i.e. an inductive way of increasing semantic and syntactic complexity starting with some instances of the axioms, ending with the given formula and using a set of inference rules (valid arguments). In a way, a formal proof is the opposite of deduction.

Since the valid arguments within $\mathcal{A}_2$ are equivalences, nothing could prevent us from reading them from the right to the left (as collapsing rules instead of expansion rules). Therefore, we can also consider $CCBL$ to be an inductive (Hilbertian) formal theory.

Fortunately, the inductive formal proof can be parsed as a buttom-to-top traversal of the deductive proof. Consequently, in $CCBL$, inductive and deductive discourses of a formula perfectly match each other. This is why the distinction between the terms 'deductively provable' and 'inductively provable' is no longer necessary here.

3.5 Completeness and Soundness of CCBL

The deductive discourse presented in Fig.1 can be easily expanded to a deductive proof. Modus Ponens and Lukasiewicz's axioms (18-20) can be deductively proved by following the expansion rules of $\mathcal{A}_2$. Hence, Lukasiewicz's formal theory of binary propositional logic is a sub-theory of $CCBL$, and therefore (see formula 28):

$$LBL \subset CCBL \subset SCBL \subset LBL. \tag{33}$$

Hence, $CCBL$ and $SCBL$ theories are sound and complete.

3.6 CNF-ization

Definition 3 *A Cognitive Conjunctive Normal Form (CCNF) is a conjunction of cognitive implications:*

$$\bigwedge_{i=1}^{n}\left(t \rightarrow \bigvee_{j=1}^{m} c_i^j\right), \tag{34}$$

where the variables c_i^j may contain negation but no other connective.

A full deductive discourse of a well-formed formula of $CCBL$ describe a conversion of that formula into a $CCNF$, i.e. a CNF-ization procedure.

D is the set of all equivalences (31) between formulae within $FORM$ and their corresponding CNF. Any element of D is the summary of a full deductive discourse.

3.7 Semantic Closure of Propositional Binary Logic

Ultimately, binary logic is a study of two-element Boolean algebra and also a study of all the possible worlds which are compatible with logical formalization - a process of abstraction in which the facts and/or the rules within a universe are represented as elements of a *logical formal language*, namely: logical constants, logical variables and logical formulae. In this context, one may be led to believe that a logical variable and a logical formula are two different things. But, in fact, it is shown below that they aren't.

Let us recap what we have so far: $B = \{0, 1\}$ - the set of binary truth values, V - a set of truth-functional variables, $FORM$ - the set of well-formed formulae, $CI \subset FORM$ - the set of cognitive implications, D - the set containing the summaries of the deductive discourses. But, what exactly is V? V is a collection of symbols (labels) representing all kind of things which are true or false, but never simultaneously true and false. Therfore, in $CCBL$: $FORM \subset V$ and $D \subset V$, or in other words, $CCBL$ is the natural unified universe of discourse about binary truth values, truth-functional variables, logical formulae, deductive/inductive discourses and CNF-ization.

In other words, the vocabulary of propositional binary logic is rich enough to express its meanings. Since the vocabulary of L contains all the elements of its meta-language $\mathcal{L}$, it is clear now that $CCBL$ formalization is nothing but an instrument meant to make obvious the semantic richness of binary logic and the structural complexity hidden behind the definition of V. We started by thinking that V was a simple set of propositional variable and, at the end, we saw that the ways of reasoning about elements of V belong in V. This is why we said that the (vocabulary of) *propositional binary logic is semantically closed* (or self-described). Hence, there is no need to create/use a markup language for describing the meanings of propositional binary logic.

3.8 The fundamental property of binary logic vocabulary

If p belongs to the vocabulary of propositional binary logic ($p \in V$), the following formulae are deductively provable in $CCBL$:

$$t \rightarrow [(t \rightarrow p) \rightarrow (\neg(p \rightarrow f))] \tag{35}$$

$$t \rightarrow [(\neg(p \rightarrow f)) \rightarrow (t \rightarrow p)] \tag{36}$$

The proof is that both formulae (35),(36) admit a deductive proof and the consequence is that, for any logical variable p, the following formula holds true:

$$(t \rightarrow p) \leftrightarrow \neg(p \rightarrow f), \tag{37}$$

or, in other words, for any $p \in V$, the following formula is false:

$$(t \rightarrow p) \wedge (p \rightarrow f) \tag{38}$$

Of course, the fact that formula (38) is false can be proved directly in $CCBL$. Fig.2 shows a deductive discourse (easily expandable to a deductive proof) of $CCBL$ formula:

$$((t \rightarrow p) \wedge (p \rightarrow f)) \rightarrow f \tag{39}$$

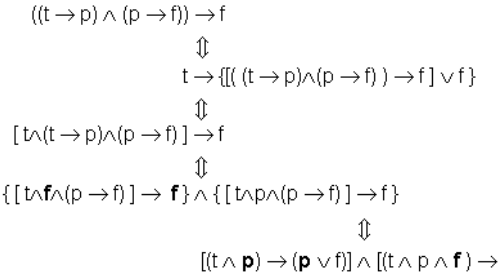

Figure 2: A deductive discourse for formula (39)

3.9 There is no Liar Paradox in CCBL!

One might believe that the *Liar Paradox* gives an example of a well-formed formula of propositional binary logic which does not have a truth value. This is completely wrong. Let us investigate the existence of a true propositional variable $p \in V$ which says about itself that it is false. Since $CCBL$ is a complete and sound theory, if such a propositional variable exists, then it should be *reachable* through a proof or a resolution.

Let us assume that there is such a propositional variable $p \in V$ so that it is true and says about itself that it is false. Let us translate from natural language - 'it is true that p is true and p claims that it is false', to the language of $CCBL$:

$$t \rightarrow \{[(t \rightarrow p) \wedge (p \rightarrow f)] \vee f\} \tag{40}$$

The deductive way to disprove the existence of p is to expand and to analyze the full deductive discourse of (40). It should be clear that the full deductive discourse of (40) is a succesful deconstruction revealing that $p \notin V$, and more precisely that '*formula*' (40) carries genes of a paraconsistent logic [10].

On the other hand, the existence of p can be disproved in an inductive manner: by showing that assuming (40) will rapidly lead to contradiction. Indeed, by applying the syllogism principle it is clear that, if (40) is assumed to be true, the following four equivalent formulae should also be true:

$$t \rightarrow \{(t \rightarrow f) \vee f\}; t \rightarrow (t \rightarrow f); t \rightarrow f; f \vee f. \tag{41}$$

But obviously, they aren't. Hence, the assumption that $p \in V$ is false and consequently, even if 'formula' (40) *apparently seems to be well-formed* (as a pictogram), it is not semantically (hence, nor sintactically) well-formed because it contains a symbol extraneous to V. Consequently, 'formula' (40) is a string which confuses us with its pictographic mimetism, but nothing more.

Hence, in propositional binary logic, the so called *Liar Paradox* is just an abuse of formalization, a wrong attempt to apply logical principles beyond the boundaries of the vocabulary of classical propositional binary logic, a dangerous way to claim that classical binary logic (classical logical thinking) should be applicable to the vocabulary of a paraconsistent [10] logical theory.

In $CCBL$ theory, the so-called *Liar Paradox* is totally and definitely deconstructed. An important consequence is that any formal theory in which the *Liar Paradox* is a well-formed true formula doesn't contain a sufficient amount of binary logic. More precisely, if the *Liar Paradox*, the syllogism principle, and formula $[(t \rightarrow \alpha) \leftrightarrow \alpha]$ are theorems within a theory, then that theory is inconsistent because, as shown above, in such a theory, the false formula $(t \rightarrow f)$ is provable.

4 Conclusion

Tarski held that the possibility of formulating paradoxically self-reference sentences belongs to semantically closed languages. He thought that a distinction between the language and the metalanguage is needed in order to avoid such faults. $CCBL$ formal theory gives a relevant counter-example. Humans can make mistakes but a deterministic machine can only do what is designed to do.

References:

[1] M. E. Szabo, *The collected papers of Gerhard Gentzen*, North-Holland Publishing Company, Amsterdam, 1970.

[2] J. H. Gallier, *Logic for computer science: foundations of automatic theorem proving*, Harper & Row Computer Science And Technology Series, New York, 1985.

[3] R. M. Sainsbury, *Paradoxes*, Cambridge University Press, 2009.

[4] W. J. M. Levelt, *Introduction to Theory of Formal Languages and Automata*, John Benjamins Publishing Company, 2008.

[5] H. B. Curry, *A Theory of Formal Deducibility*, University of Notre Dame (Indiana), 1950.

[6] R. Bornat, *Proof and Disproof in Formal Logic*, Oxford University Press, 2005.

[7] R. Stansifer, *Completeness of Propositional Logic as a Program*, Technical Report, Department of Computer Sciences, Florida Institute of Technology, March 2001.

[8] D. Hilbert, W. Ackermann, *Principles of Mathematical Logic*, Chelsea Publishing Company, 1950.

[9] P. Blackburn, M. de Rijke, Y. Venema, *Modal Logic*, Cambridge University Press, 2000.

[10] J. Woods, *Paradox and paraconsistency: conflict resolution in the abstract sciences*, Cambridge University Press, 2003.