Journal of Mathematics Research; Vol. 11, No. 3; June 2019
ISSN 1916-9795 E-ISSN 1916-9809
Published by Canadian Center of Science and Education

Atomic Proposition and 1-Order Predicate Function for Dialectical Logic

Yaozhi Jiang

Correspondence: Yaozhi Jiang, Shijiazhuang High-Tech District, Hebei, China

Received: April 14, 2019 Accepted: May 8, 2019 Online Published: May 13, 2019

doi:10.5539/jmr.v11n3p50 URL: https://doi.org/10.5539/jmr.v11n3p50

Abstract

This paper has completed main fields of making dialectical logic pure mathematically, it is involved both atomic proposition and 1-order predicate function for dialectical logic, and by state-dual, true-valued function vector, state-contradiction law into basic logic law. In addition, also defines true-valued function for logic operators so that more easy to represent atomic proposition. Some examples are given and shown that Boolean algebra, as a special case of dialectical logic, is how to operate hybridize-able with dialectical logic.

Keywords: dialectical logic, atomic proposition, compound proposition, 1-order predicate function, operators valued by true-valued function

1. Introduction

As a further successor of author's work (Yaozhi Jiang, 2019), this paper explains atomic proposition and 1-order predicate function in dialectical logic, hence completes the essential methods pure mathematically for dialectical logic.

2. State, Event and True-Valued Function

The word "state" as we known is defined as phenomenon of some systems, such as occurrence of some things or some matters, variant process dynamically under defined conditions. "Event" is the critical point of state. Between two of steady-states is transient, and of that either steady-state or transition state all produced by some conditions contradicted each other. For this situations formal logic, only 0-1 binary logic state expression, is at a loss what to do. We have not ability to do anything logical without dialectical logic, however it is argument.

In order to express the "truth-or-falsity" mathematically, thus we need true-valued function.

Boolean algebra is the earliest true-valued function by a 0-1 binary system for formal logic and it is very successful, and later the multiple true-valued function have been established for logic system. Both of them are special cases of dialectical logic were proved by author (Yaozhi Jiang, 2019). But now true-valued function for dialectical logic, it must be a continuous function, has been built for variant dynamical logic problems. In addition, the true-valued function of dialectical logic neither "probability" in probability theory nor "membership function" in fuzzy mathematics, it is an expression of state-contradiction by state-dual.

Dialectical logic permits state-contradiction, even if thinks of that state-contradiction is either driving force or resistant force and produces variants of states and events. In any cases dialectical logic never permit the existence of contradiction from causality law as well as formal logic does.

3. Some Definitions and Symbols and the Four Basic Laws in Dialectical Logic

3.1 Some Definitions, Symbols and Theorems About Proposition

As same as usual, connective operators: $\wedge$ is "conjunction operator", and $\vee$ is "disjunction operator", and $\neg$ is "negation operator", and $\Rightarrow$ is "implication operator", and $\Leftrightarrow$ is "logic equivalent operator", "logic quantifiers" $\forall, \exists$, etc.

Property 3.1.1. Property of implication operator $\Rightarrow$

For atomic proposition function P, Q, E,

1) The meaning of implication operator $P \Rightarrow Q$ is: if P, then Q;

2) Reflection law: $P \Rightarrow P$;

3) Transmission law: If $P \Rightarrow Q \Rightarrow E$, then $P \Rightarrow E$;

4) Semi-commutation law: If $(P \Rightarrow Q) \Rightarrow (Q \Rightarrow P)$, then $P \Leftrightarrow Q$;

5) Anti-distribution law(in broad sense of $\neg$):

$$\neg(P \Rightarrow Q) \Leftrightarrow \begin{cases} \neg P \overset{\neg}{\Rightarrow} \neg Q \\ P \overset{\neg}{\Rightarrow} \neg Q \\ \neg P \Rightarrow \neg Q \\ \neg P \overset{\neg}{\Rightarrow} Q \\ \neg P \Rightarrow Q \\ P \overset{\neg}{\Rightarrow} Q \\ P \Rightarrow \neg Q \end{cases}$$

6) Anti-distribution law (in narrow sense of $\neg$)

$$\neg(P \Rightarrow Q) \Leftrightarrow \left(P \overset{\neg}{\Rightarrow} Q\right)$$

Property 3.1.2. Denote true-valued function of atomic proposition P, Q by $T(P)$ and $T(Q)$ separately, if $T(P \Rightarrow Q) = 1$, then $T(\neg(P \Rightarrow Q)) = 0$.

Property 3.1.2 above shows that the well-known implication operator true-valued table, can be seen in any text-book about formal logic, is indeed an **Implication Paradox.**

Definition 3.1.1. True-valued function of operators: if logic operator set O, $\forall O_s \in O$, $\exists T_d(O_s) \in [-1,+1]$, make $T_d(O_s)$ is true-valued function to express truth-or-falsity of the logic operator O_s.

Definition 3.1.2. $T(P) \wedge T(Q) = \min\{T(P), T(Q)\}$; $T(P) \vee T(Q) = \max\{T(P), T(Q)\}$.

Property 3.1.3. $\wedge \vee = 1$, $\wedge = \dfrac{1}{\vee} = \vee^{-1}$

Property 3.1.4. $\neg\neg = 1$, $\neg = \dfrac{1}{\neg} = \neg^{-1}$

Property 3.1.5. $\neg\wedge = \vee$, $\neg\vee = \wedge$

Property 3.1.6. In dialectical logic proposition variable x, if a contradiction is composed of positive factor x^+ and negative factor x^-, and if denote true-valued function of positive factor by $T_d(x^+)$, then $T_d(x^-) = T_d(x^+) - 1$, therefor any state-contradiction can be represented by state-dual $(\alpha^+, (\alpha^+ - 1))$, especially if $T_d(x^+) = 0^+$, then $T_d(x^-) = -1$; if $T_d(x^-) = 0^-$, then $T_d(x^+) = +1$, this case just is Boolean algebra.

Property 3.1.7. If $T(x^\pm) = (\alpha^+, \alpha^+ - 1)$, then $\neg T(x^\pm) = T(\neg x^\pm) = (-(\alpha^+), -(\alpha^+ - 1)) = (1 - \alpha^+, -\alpha^+)$

3.2 The Four Basic Laws in Dialectical Logic

Aristotelian three logic laws added to state-contradiction law makes the four basic laws in dialectical logic below, and these laws are taken form of predication proposition formula.

1) Law of identity (reflexive law)

$(\forall x)(F(x) \Leftrightarrow F(x))$

2) Law of non-contradiction

$(\forall x)(\neg(F(x) \wedge (\neg(F(x)))))$

3) Law of excluded middle

$(\forall x)(F(x) \vee (\neg(F(x))))$

4) Law of state-contradiction

$(\forall x^\pm)(F(x^\pm) \Leftrightarrow F(x))$, iif $T(x^\pm) = (\alpha^+, \alpha^+ - 1)$

4. Atomic Proposition

Now we define three predicative operators, $\xrightarrow{be}$, $\xrightarrow{do}$ and $\xrightarrow{imply}$ for atomic proposition. Author divides not only predicative operator into three types, but also think interrogative pattern as proposition, thus defines an

operator as interrogative operator?. For negative operator $\neg$, we see it acts not only upon predicative operators, but also on antecedent and on consequent in atomic proposition, therefor there will produce $2^3-1=7$ kinds of negative forms in broad sense, except of positive form.

Definition 4.1. Predictive operators $\xrightarrow{be}$, $\xrightarrow{do}$, $\xrightarrow{imply}$ and their true-valued function $T\left(\xrightarrow{be}/\xrightarrow{do}/\xrightarrow{imply}\right)$.

Definition 4.2. Interrogative operator? both interrogative operator and answer also is a proposition.

4.1 Atomic Proposition

Absolutely-defined relationship, also objective definition, can be defined by nouns, prepositions, or verbs in ordinary.

Relatively-defined relationship, also subjective definition, can be defined by adjectives, or adverbs in ordinary.

Denote proposition variables by small letters $x, y, z, \cdots$, and atomic proposition functions by capital letters $P(x, y, z, \cdots)$, $Q(x, y, z, \cdots)$, $R(x, y, z, \cdots)$

1) Positive-defined pattern

$$P(x, f): x \xrightarrow{be} f; T(P(x, f)) = (\alpha^+, \alpha^+ - 1), \alpha^+ \in [0, +1]$$

2) Negative-defined pattern

$$\neg P(x, f): \begin{cases} \neg x \xrightarrow{be} f \\ \neg x \xrightarrow{\neg be} f \\ \neg x \xrightarrow{\neg be} \neg f \\ x \xrightarrow{\neg be} \neg f \\ x \xrightarrow{be} \neg f \\ \neg x \xrightarrow{be} \neg f \\ x \xrightarrow{\neg be} f \end{cases}$$

3) Positive-defined with uncertainty pattern

$$P_U(x, f): \begin{cases} x_u \xrightarrow{be} f \\ x_u \xrightarrow{be_u} f \\ x_u \xrightarrow{be_u} f_u \\ x_u \xrightarrow{be} f_u \\ x \xrightarrow{be_u} f_u \\ x \xrightarrow{be_u} f \\ x \xrightarrow{be} f_u \end{cases}$$

4) Negative-defined with uncertainty pattern

$$P_{\neg U}(x, f) \Leftrightarrow P(x, f)$$

5) Positive do-pattern

$$P_d(x, y): x \xrightarrow{do} y$$

6) Negative do-pattern

$$\neg P_d(x,y):\begin{cases}\neg x \xrightarrow{do} y\\ \neg x \xrightarrow{\neg do} y\\ \neg x \xrightarrow{\neg do} \neg y\\ \neg x \xrightarrow{do} \neg y\\ x \xrightarrow{\neg do} \neg y\\ x \xrightarrow{\neg do} y\\ x \xrightarrow{do} \neg y\end{cases}$$

7) Positive-interrogative pattern

$$P_?(x, f/y):\begin{cases}x_? \xrightarrow{be/do} (f/y)\\ x_? \xrightarrow{(be/do)_?} (f/y)\\ x_? \xrightarrow{(be/do)_?} (f/y)_?\\ x \xrightarrow{(be/do)_?} (f/y)\\ x \xrightarrow{be/do} (f/y)_?\\ x_? \xrightarrow{(be/do)} (f/y)_?\\ x \xrightarrow{(be/do)_?} (f/y)_?\end{cases}$$

8) Negative-interrogative pattern

$$P_{\neg ?}(x, f/y) \Leftrightarrow P(x, f/y)$$

9) Positive-causality descriptive pattern

$$P_s(x, f/y): x \xrightarrow{imply} f/y$$

10) Negative-causality descriptive pattern

$$\neg P_s(x, f/y):\begin{cases}\neg x \xrightarrow{imply} (f/y)\\ \neg x \xrightarrow{\neg imply} (f/y)\\ \neg x \xrightarrow{\neg imply} \neg(f/y)\\ x \xrightarrow{\neg imply} (f/y)\\ x \xrightarrow{\neg imply} \neg(f/y)\\ x \xrightarrow{imply} \neg(f/y)\\ \neg x \xrightarrow{imply} \neg(f/y)\end{cases}$$

4.2 True-Valued Function Analysis for Atomic Proposition

4.2.1 Without Loss Generality, We Can Obtain Below

$$P_{1.1}(x, f/y): x \xrightarrow{v} (f/y) \qquad P_{1.2}(x, f/y): \neg x \xrightarrow{\neg v} \neg(f/y)$$
$$P_{2.1}(x, f/y): \neg x \xrightarrow{v} (f/y) \qquad P_{2.2}(x, f/y): x \xrightarrow{\neg v} \neg(f/y)$$
$$P_{3.1}(x, f/y): x \xrightarrow{v} \neg(f/y) \qquad P_{3.2}(x, f/y): \neg x \xrightarrow{\neg v} (f/y)$$
$$P_{4.1}(x, f/y): \neg x \xrightarrow{v} \neg(f/y) \qquad P_{4.2}(x, f/y): x \xrightarrow{\neg v} (f/y)$$

If we denote $\xleftarrow{complement}$ to express complement relationship, and $\xleftarrow{negation}$ to express negation relationship,

then we have below:

$$P_{1.1} \xleftrightarrow{complement} P_{1.2};\ P_{2.1} \xleftrightarrow{complement} P_{2.2};\ P_{3.1} \xleftrightarrow{complement} P_{3.2};\ P_{4.1} \xleftrightarrow{complement} P_{4.2};$$

And

$$P_{1.1} \xleftrightarrow{negation} P_{4.2};\ P_{2.1} \xleftrightarrow{negation} P_{3.2};\ P_{3.1} \xleftrightarrow{negation} P_{2.2};\ P_{4.1} \xleftrightarrow{negation} P_{1.2};$$

Because of existence of predicative-paradox, an impossibility and the absurdity of proposition, such as

1) Dragon can fly.

2) Tree can become into mankind.

3) High temperature will imply male is pregnant.

have been excluded out in semantics in this paper

4.2.2 Examples for Dialectical Logic Atomic Proposition

Definition 4.2.2.1. In atomic proposition P, the vector $V(P)$ of true-valued function $T(P)$ is a row sequence of state-dual of each term in P, i.e.

$$\begin{aligned} V(P) &= (T(a);T(\xrightarrow{\vee});T(b/f);T(P)) \\ &= ((\alpha_1^+,\alpha_1^+-1);(\alpha_2^+,\alpha_2^+-1);(\alpha_3^+,\alpha_3^+-1);(\alpha_T^+,\alpha_T^+-1)) \end{aligned}$$

Examples

1) Proposition P: Mankind do not eat grasses.

Denote $T(\text{mankind}) = T(a) = (+1,0^-)$; $T(\text{do not eat}) = T(\xrightarrow{\neg do}) = (0^+,-1)$, $T(\text{grass}) = T(f/b) = (0.1,-0.9)$, because of sometimes mankind can eat 10% kinds of grasses, and 90% kinds of grasses can not be eaten, then

true-valued function in dialectical logic of atomic proposition P is $T(P) = T_P(P) = (0.9,-0.1)$, then

$$V(P) = ((+1,0^+);(0^+,-1);(0.1,-0.9);(0.9,-0.1))$$

and meaning of the true-valued function $T(P)$ is that proposition P is included by 90% partial-truth and 10% partial-falsity, and its true-valued function in form of Boolean algebra will be

$$T(P) = \begin{cases} 0, & \text{if we think of the proposition is full - false;} \\ 1, & \text{if we think of the proposition is full - true;} \end{cases}$$

2) Proposition Q: The person who scuffed John almost is Mary. $T(\text{almost is Mary}) = (0.8,-0.2)$,

$T(\xrightarrow{scuff}) = (+1,0^-)$, $T(\text{John}) = (+1,0^-)$, then $T(Q) = (+1,0^-)$, and $V(Q) = ((0.8,-0.2);(+1,0^-);(+1,0^-);(+1,0^-))$.

As well as we have noted that Boolean algebra is a special case of dialectical logic in (Yaozhi Jiang, 2019), now Boolean algebra and dialectical logic are laid on same proposition in non-contradictory and can be operated in hybrid operations.

Remark1: The true-valued function is a evaluation function for truth-or-falsity of atomic proposition, then the true-valued function can not been introduced from true-valued function of each term of the proposition as above. There are differences between semiotics and semantics. In addition the ambiguousness often be produced by variant evaluators so that must be influenced true-valued function, especially relative-defined case.

5. Serial True-Valued Function and Parallel True-Valued Function

5.1 Multiple-Order True-Valued Function

If real number set $N = \{a_1, a_2, \cdots, a_k \mid a \text{ is real number}\}$, then

arithmetic average operator O_a, $O_a : N = \frac{1}{k}\sum_{s=1}^{k} a_s$;

geometric average operator O_g: $O_g : N = \sqrt[k]{\prod_{s=1}^{k} a_s}$.

Operation rule: In the formulation of O_a and O_g, all of true-valued function are taken its positive, if zero then it is stroke out.

If we see a true-valued function $T_d(P)$ as a new proposition, then we can define a 2-order true-valued function by $T_d(T_d(P))$, and 3-order true-valued function by $T_d(T_d(T_d(P)))$, and so on, till the $i-$order true-valued function by $\underbrace{T_d(T_d(T_d(\cdots T_d(P))))}_{i}$. This is a serial true-valued function. The multiple-order true-valued function is geometric average,

i.e. $O_g = \sqrt[i]{T_{d(i)}T_{d(i-1)}T_{d(i-2)}\cdots T_{d(1)}}$, in which if some true-valued function is zero then it is stroke out.

We can also define a parallel true-valued function by arithmetic average $O_a = \frac{1}{k}\left(T_{d(1)} + T_{d(2)} + \cdots + T_{d(k)}\right)$

In which if some true-valued function is zero then it is stroke out.

5.2 Some Special Applications by True-Valued Function Average Value

We have defined geometric average O_g and arithmetic average O_a above, then we have some applied examples below.

Example 5.2.1. Some developing plan needs be estimated and approved by higher levels one-level-by-one-level, then the total results estimated and approved can use geometric average to express, if existence of veto power we can not strike the zero out.

Example 5.2.2. In some sport games, the system of average points by referees must be used, then the arithmetic average O_a is useful.

6. Predicative Function for Compound Proposition of Dialectical Logic

Now we are going to consider compound proposition as usual, a compound proposition is produced by several atomic propositions connected by connective operators $\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$, etc. For these operators we do define a true-valued function is equal to $T(\wedge/\vee/\neg/\Rightarrow/\Leftrightarrow)=1$, it is because of avoiding the more complexity.

6.1 Definitions and Symbols

As usual we use definitions and symbols below:

1) "individual" expresses discussed properties or elements, denoted by small letter $a, b, c, \cdots$, logic constants, or by small letter $x, y, z, \cdots$, logic variables;

2) "Individual domain" is denoted by D, and "Universal domain" is denoted by D_U, they are defined-domains of logic variables;

3) "$n-$place predicate function or $n-$place proposition function" is predicate function or proposition function included $n-$variable and denoted by $P(x_1, x_2, x_3, \cdots, x_n)$, $\{x_1, x_2, x_3, \cdots, x_n\} \in D^n$, where D^n valued on closed interval $[-1,+1]$; defined-domain D/D_U of logic variable/predicate function/proposition function, especially in which if logic variable x is displaced by logic constant a, then it is a $n-$place proposition, in addition $0-$place proposition function also is a proposition;

4) "interpretation" is that displace every logic variable by a logic constant, and valued them with state-dual to every constant in a symbol string formula, therefor the predicate function become into a proposition;

5) "Characteristic predicate function $\Lambda(x)$" is a restricted condition for logic variable x by $\Lambda(x)$;

6) Symbol "$\mapsto$", e.g. $P(x) \mapsto Q(x)$, means that from $P(x)$ can lead to $Q(x)$ by reasoning;

6.2 Some Properties and Theorems for Predicate Formulas

A dialectical logic predicate formula S_i, can also be called as a predicate symbol string S_i.

Property 6.2.1. For a dialectical logic predicate formula S_i, every atomic proposition s_i, excluded brackets and connective operators out in S_i, we can obtain a true-valued function expression, i.e.

$T(S_i)=\{s_1,s_2,\cdots,s_i\}$

$\forall s_i,\ \exists T(s_i)=\{T(s_1),T(s_2),\cdots,T(s_i)\}\in[-1,+1],\ \ s_i\in S_i;$

Property 6.2.2. Distribution law

$T(P(x_i)\wedge P(x_j))=T(P(x_i))\wedge T(P(x_j))$

$T(P(x_i)\vee P(x_j))=T(P(x_i))\vee T(P(x_j))$

Property 6.2.3. The true-valued function of De Morgan law

1) $T(\neg(P\wedge Q))=T((\neg P)\vee(\neg Q))$

2) $T(\neg(P\vee Q))=T((\neg P)\wedge(\neg Q))$

Property 6.2.4. Logic quantifiers $\forall,\exists$

We see $\forall,\exists$, rather logic operator than logic quantifiers, in broad sense below

1) $$T(\neg((\forall x)(\Lambda(x)\Rightarrow P(x))))=T\left(\begin{cases}\neg\forall x\begin{cases}(\neg\Lambda(x)\Rightarrow P(x))\\(\Lambda(x)\Rightarrow\neg P(x))\\(\neg\Lambda(x))\Rightarrow\neg P(x)\end{cases}\\ \forall\neg x\begin{cases}(\neg\Lambda(x)\Rightarrow P(x))\\(\Lambda(x)\Rightarrow\neg P(x))\\(\neg\Lambda(x))\Rightarrow\neg P(x)\end{cases}\\ \neg\forall\neg x\begin{cases}(\neg\Lambda(x)\Rightarrow P(x))\\(\Lambda(x)\Rightarrow\neg P(x))\\(\neg\Lambda(x)\Rightarrow\neg P(x))\end{cases}\end{cases}\right)$$

2) $$T(\neg((\exists x)(\Lambda(x)\Rightarrow P(x))))=T\left(\begin{cases}(\neg\exists X)\begin{cases}\Lambda(x)\Rightarrow\neg P(x)\\\neg\Lambda(x)\Rightarrow\neg P(x)\\\Lambda(x)\Rightarrow P(x)\end{cases}\\ (\exists\neg x)\begin{cases}\Lambda(x)\Rightarrow\neg P(x)\\\neg\Lambda(x)\Rightarrow\neg P(x)\\\Lambda(x)\Rightarrow P(x)\end{cases}\\ \neg\exists\neg x\begin{cases}\Lambda(x)\Rightarrow\neg P(x)\\\neg\Lambda(x)\Rightarrow\neg P(x)\\\Lambda(x)\Rightarrow P(x)\end{cases}\end{cases}\right)$$

Property 6.2.5. Basic equivalence formulas

1) $(\forall x)P(x)\wedge(\forall x)Q(x)\Leftrightarrow(\forall x)(P(x)\wedge Q(x))$, 1') $(\exists x)(P(x))\vee(\exists x)(Q(x))\Leftrightarrow(\exists x)(P(x)\vee Q(x))$;

2) $(\forall x)(P(x))\Leftrightarrow\neg(\exists x)\neg P(x)$,

2') $(\exists x)P(x)\Leftrightarrow\neg(\forall x)\neg P(x)$;

3) $\neg((\forall x)(P(x)))\Leftrightarrow(\exists x)(\neg P(x))$,

3') $\neg((\exists x)(P(x)))\Leftrightarrow(\forall x)(\neg P(x))$;

4) $(P(x)\Leftrightarrow Q(x))\Leftrightarrow(P(x)\Rightarrow Q(x))\wedge(Q(x)\Rightarrow P(x))$,

4') $(P(x)\Rightarrow Q(x))\Leftrightarrow(\neg P(x)\vee Q(x))$;

5) $(\forall x)(\forall y)P(x,y)\Leftrightarrow(\forall y)(\forall x)P(x,y)$,

5') $(\exists x)(\exists y)P(x,y)\Leftrightarrow(\exists y)(\exists x)P(x,y)$.

6.3 Some Properties for Quantifier Phrase

For quantifier phrase $\forall x/\exists x$, the formula in brackets followed upon the quantifier phrase is called as scope of the quantifier phrase; and the variable x is called as guide-variable of its quantifier; and variable same as guide-variable in brackets is called bound-variable; the other variable, excluded bound-variable out, is called as free-variable. The occurrence of bound-variable/free-variable is called as bound-occurrence/free-occurrence.

Property 6.3.1. Basic equivalence formulas for quantifiers

1) $\neg(\forall x)P(x) \Leftrightarrow (\exists x)\neg P(x)$,

1') $\neg(\exists x)P(x) = (\forall x)\neg P(x)$.

Property 6.3.2. Basic equivalence formulas for quantifier scope

1) $(\forall x)(P(x) \vee Q) \Leftrightarrow (\forall x)P(x) \vee Q$,

1') $(\forall x)(P(x) \wedge Q) \Leftrightarrow (\forall x)P(x) \wedge Q$;

2) $(\forall x)(P(x) \Rightarrow Q) \Leftrightarrow (\exists x)P(x) \Rightarrow Q$,

2') $(\forall x)(Q \Rightarrow P(x)) \Leftrightarrow Q \Rightarrow (\forall x)P(x)$;

3) $(\exists x)(P(x) \vee Q) \Leftrightarrow (\exists x)P(x) \vee Q$,

3') $(\exists x)(P(x) \wedge Q) \Leftrightarrow (\exists x)P(x) \wedge Q$;

4) $(\exists x)(P(x) \Rightarrow Q) \Leftrightarrow (\forall x)P(x) \Rightarrow Q$,

4') $(\exists x)(Q \Rightarrow P(x)) \Leftrightarrow Q \Rightarrow (\exists x)P(x)$.

Where variable x is free-occurrence in $P(x)$, and x is not included in Q .

6.4 Predicate Function Reasoning

Semiotics and semantics: in semiotics we see a predicate function as formula consisted by a series of symbols, but in semantics we must consider the logical meaning of the predicate formula.

6.4.1 Reasoning Theorem and Some Rules for Quantifier Phrase

For predicate formula $P(x, y)$, if logic variable x is not occurrence in scope of $\forall y$ or $\exists y$ in $P(x, y)$, then we call the $P(x, y)$ is free to y .

Definition 6.4.1.

1) $\bigvee_{i=1}^{n} A_i = A_1 \vee A_2 \vee \cdots \vee A_n$ 2) $\bigwedge_{i=1}^{n} A_i = A_1 \wedge A_2 \wedge \cdots \wedge A_n$

Basic reasoning theorem: If $\bigwedge_{i=1}^{n} A_i \Rightarrow B$ is a logic validity, and B is a right reasoning result, then $\bigwedge_{i=1}^{n} A_i \Rightarrow B$ is a logic validity implication, denoted by $\bigwedge_{i=1}^{n} A_i \overset{V}{\Rightarrow} B$. This is just the reasoning theorem.

Property 6.4.1.1. A corollary from De Morgan's law

If $D = \{x_1, x_2, \cdots, x_n\}$, and D is countable, in semantics we have formulas below

1) If $\bigwedge_{i=1}^{n} A(x_i) \Leftrightarrow (\forall x)A(x) \Rightarrow B(x)$, then $\neg\left(\bigwedge_{i=1}^{n} A(x_i)\right) = \bigvee_{i=1}^{n} \neg A(x_i) \Leftrightarrow (\exists x)(\neg A(x)) \Rightarrow \neg B(x)$;

2) If $\bigvee_{i=1}^{n} A(x_i) \Leftrightarrow (\exists x)A(x) \Rightarrow B(x)$, then $\neg\left(\bigvee_{i=1}^{n} A(x_i)\right) = \bigwedge_{i=1}^{n} \neg A(x_i) \Leftrightarrow (\forall x)\neg A(x) \Rightarrow \neg B(x)$.

Property 6.4.1.2.

In same scope and in semantics,

$\neg(\forall x) = (\exists x)$, $\neg(\exists x) = (\forall x)$

6.4.2 The Four Rules

A) Rule of universal quantifier $\forall$ elimination (**UE**)

$(\forall x)P(x) \mapsto P(y)$, $(\forall x)P(x) \mapsto P(c)$

In which y is any individual variable no bound-occurrence in $P(x)$, and c is any individual constant; the operation

by **US** must be that $P(x)$ is free to y.

B) Rule of universal quantifier $\forall$ introduction (**UI**)

$$P(y) \overset{V}{\Rightarrow} (\forall x)P(x)$$

In which $P(x)$ is free to y, and individual variable x that displace y in $P(y)$ is no bound-occurrence, and if $\Lambda(x) \mapsto P(y)$ then y is free to $\Lambda(x)$, and not introduce in by rule of **ES**.

C) Rule of existence quantifier introduction (**EI**)

$$P(c) \overset{V}{\Rightarrow} (\exists x)P(x), \quad P(y) \overset{V}{\Rightarrow} (\exists x)P(x)$$

In which $P(y)$ is free to x, and individual variable x that displace c is no occurrence in $P(c)$.

D) Rule of existence quantifier $\exists$ elimination(**EE**)

$$1)\ (\exists x)P(x) \overset{V}{\Rightarrow} P(c), \qquad 2)\ (\exists x)P(x) \overset{V}{\Rightarrow} P(y),$$

In which c is an individual constant make $P(x)$ is true; and except x, while there are other free-occurrence individual constants do not use the formula 1), or while there are other free-occurrence individual variable do not use the formula 2); and $P(y)$ is free to x.

6.5 Interpretation to Predicate Formula of Dialectical Logic

Definition 6.5.1. An interpretation to predicate symbol string $P(x_i)$ of dialectical logic is denoted by $I : P(x_i)$, and defined below:

1) Every symbol of individual constant a_i assigned to a variable of domain D, $D \neq \varnothing$;

2) Every symbol of $n-$place function is assigned to a mapping

$f : D^n \to D$;

3) Every symbol of $n-$place predicate function is assigned a mapping

$f_p : D^n \to [-1,+1]$.

Example 6.5.1.

K: $p\%$ of hares run faster than tortoises.

$P(x)$: x is hare; $Q(y)$: y is tortoise; $R(x,y)$: $p\%$ of x run faster than y;

Symbolization K: $(\forall x_p)(\forall y)(P(x_p) \wedge Q(y) \Rightarrow R(x_p, y))$

Simplification

$$\begin{aligned} &(\forall x_p)(\forall y)(P(x) \wedge Q(y) \Rightarrow R(x,y)) \\ &\Leftrightarrow (\forall x_p)(\forall y)(\neg(P(x_p) \vee \neg Q(y)) \vee R(x_p, y)) \\ &\mapsto (\neg P(a) \vee \neg Q(b)) \vee R(a,b) \end{aligned}$$

I_K: $(\neg P(a) \vee \neg Q(b)) \vee R(a,b)$

D: $a_1 = \text{hare}_1$, $b_1 = \text{tortoise}_1$,

$a_2 = 60\%$ of hare_2, i.e. hare_2 with 60% probabilit y is hare or 40% probabilit y is tortoise;

$$\frac{P(a_1)}{(+1,0^-)} \quad \frac{Q(b_1)}{(+1,0^-)} \quad \frac{P(a_2)}{(60\%,60\%-1)} \quad \frac{\neg P(a_1)}{(0^+,-1)} \quad \frac{\neg Q(b_1)}{(0^+,-1)} \quad \frac{\neg P(a_2)}{(1-60\%,-60\%)}$$

$$\frac{R(a_1,b_1)}{(p\%,\ p\%-1)} \quad \frac{R(a_2,b_1)}{(p\%\times 60\%,\ p\%\times 60\%-1)}$$

$$I_{K(a_1)} : K(a_1,b_1) = R(a_1,b_1) = (p\%,\ p\%-1)$$

$$I_{K(a_2)}: K(a_2,b_1)=(\neg P(a_2)\vee\neg Q(b_1))\vee R(a_2,b_1)=$$
$$=((1-60\%,-60\%)\vee(0^+,-1))\vee(p\%\times 60\%,\ p\%\times 60\%-1)$$
$$=(40\%,-60\%)\vee(p\%\times 60\%,\ p\%\times 60\%-1)$$

Especially while p is a variable, then propositions above will be various and broad in meanings.

7. Conclusion

Both this paper and author's last paper, Yaozhi Jiang (2019), author has established basic principles and formulas for dialectical logic and proved that Boolean algebra is special case of dialectical logic via different methods. The fact shows that dialectical logic is not only laid on philosophy, but also laid on mathematics, and making dialectical logic mathematically will make dialectical logic into a powerful tool in artificial intelligence. To calm down the argument, dialectical logic does belong to mathematics or not, the two papers maybe useful.

References

Peter, S. (2010). *An Introduction to Formal Logic (Third Printing, 2010),* Cambridge University Press, New York.

Richard, L. E. (2015). *Reasoning and Formal Logic (Essays on Logic as The Art of Reasoning Well),* Advanced Reasoning Forum.

Yaozhi, J. (2019). Dialectical Logic and Boolean Algebra. *Journal of Mathematics Research, 11*(2).

Zhao. Z. K. (1995). Introduction to Dialectical Mathematical Logic, Renmin University Press, Beijing.