New Multivalued Logic System in Boolean Class by Regarding Fixed-point Binary Numbers as Truth Values

Hisashi Suzuki and Kin-ya Sugimoto, *Member, IEEE*

***Abstract*–This article, by regarding fixed-point binary numbers as truth values, defines a new multivalued logic system in Boolean class such that the set of logic formulas forms a Boolean algebra. This article also shows some typical examples demonstrating that, on the proposed logic system, after learning we can handle inferences readily because of Boolean effect.**

***Index Terms*–Boolean multivalued logic system, fuzzy logic, inference, truth value.**

I. INTRODUCTION

Multivalued logic systems usually are defined in non-Boolean class. In contrast, this article, by regarding fixed-point binary numbers as truth values, defines a new multivalued logic system in Boolean class such that the set of logic formulas forms a Boolean algebra, cf. II.

This article also shows some typical examples demonstrating that, on the proposed logic system, after learning we can handle inferences readily because of Boolean effect, cf. III.

The proposed logic system corresponds to a bitwise-processable realization of complementary fuzzy logic system, cf. IV.

II. BOOLEAN MULTIVALUED LOGIC SYSTEM

This section defines newly a multivalued logic system, and verifies that the set of logic formulas forms a Boolean algebra. (Note that usual Boolean logic systems are not multivalued.)

A. Definition

Each *atom* has a *truth value* that is an arbitrary binary number $0.x^{(1)}x^{(2)}\cdots x^{(n)}{}_{\rm b}$ ($x^{(1)}\in\{0,1\}$, $x^{(2)}\in\{0,1\}$, $\cdots$, $x^{(n)}\in\{0,1\}$) in fixed-point notation, where n is a constant positive integer called *dimension*. *Logic formulas* are results of applying the following logic operations to atoms and to logic formulas themselves.

The *negation* $\neg x$ of a logic formula $x = 0.x^{(1)}x^{(2)}\cdots x^{(n)}{}_{\rm b}$ has the truth value $0.z^{(1)}z^{(2)}\cdots z^{(n)}{}_{\rm b}$ that comprises the Boolean binary negation

$$z^{(i)} = \neg x^{(i)} \tag{1}$$

of $x^{(i)}$ for $i = 1, 2, \cdots, n$. For example, when

$$\begin{aligned} x &= 0.0000\ 0000\ 0000\ 1111_{\rm b} \\ &= 0.0002288818359375\,, \end{aligned} \tag{2}$$

we have

$$\begin{aligned} \neg x &= 0.1111\ 1111\ 1111\ 0000_{\rm b} \\ &= 0.9997711181640625\,. \end{aligned} \tag{3}$$

The *conjunction* $x \wedge y$ of logic formulas $x = 0.x^{(1)}x^{(2)}\cdots x^{(n)}{}_{\rm b}$ and $y = 0.y^{(1)}y^{(2)}\cdots y^{(n)}{}_{\rm b}$ has the truth value $0.z^{(1)}z^{(2)}\cdots z^{(n)}{}_{\rm b}$ that comprises the Boolean binary conjunction

$$z^{(i)} = x^{(i)} \wedge y^{(i)} \tag{4}$$

of $x^{(i)}$ and $y^{(i)}$ for $i = 1, 2, \cdots, n$. For example, when

$$\begin{aligned} x &= 0.0000\ 0000\ 0000\ 1111_{\rm b} \\ &= 0.0002288818359375 \end{aligned} \tag{5}$$

and

$$\begin{aligned} y &= 0.1001\ 1001\ 1001\ 1001_{\rm b} \\ &= 0.5999908447265625\,, \end{aligned} \tag{6}$$

we have

$$\begin{aligned} x \wedge y &= 0.0000\ 0000\ 0000\ 1001_{\rm b} \\ &= 0.0001373291015625\,. \end{aligned} \tag{7}$$

Similarly, the *disjunction* $x \vee y$ has the truth value $0.z^{(1)}z^{(2)}\cdots z^{(n)}{}_{\rm b}$ that comprises the Boolean binary disjunction

$$z^{(i)} = x^{(i)} \vee y^{(i)} \tag{8}$$

for $i = 1, 2, \cdots, n$.

Although the above logic operations are enough to specify one Boolean logic system, we furthermore introduce two logic operations for efficient handling of inferences. The *implication* $x \rightarrow y$ from a logic formula $x = 0.x^{(1)}x^{(2)}\cdots x^{(n)}{}_{\rm b}$ to another logic formula $y = 0.y^{(1)}y^{(2)}\cdots y^{(n)}{}_{\rm b}$ has the truth value $0.z^{(1)}z^{(2)}\cdots z^{(n)}{}_{\rm b}$ that comprises the Boolean binary implication

$$z^{(i)} = x^{(i)} \rightarrow y^{(i)} \tag{9}$$

from $x^{(i)}$ to $y^{(i)}$ for $i = 1, 2, \cdots, n$. Similarly, the *equivalence* $x \leftrightarrow y$ has the truth value $0.z^{(1)}z^{(2)}\cdots z^{(n)}{}_{\rm b}$ that comprises the Boolean binary equivalence

$$z^{(i)} = x^{(i)} \leftrightarrow y^{(i)} \tag{10}$$

H. Suzuki is with Department of Information and System Engineering, Chuo University, Tokyo 112-8551, Japan (e-mail: suzuki@ise.chuo-u.ac.jp).

K. Sugimoto is with Graduate School of Science and Engineering, Chuo University, Tokyo 112-8551, Japan (e-mail: ksugimoto@suzuki-lab.ise.chuo-u.ac.jp).

between $x^{(i)}$ and $y^{(i)}$ for $i = 1, 2, \cdots, n$.

Based on the above definition of logic operations, a logic formula has a certain binary number $0.x^{(1)}x^{(2)}\cdots x^{(n)}{}_{\mathrm{b}}$ ($x^{(1)}\in\{0,1\}$, $x^{(2)}\in\{0,1\}$, $\cdots$, $x^{(n)}\in\{0,1\}$) in fixed-point notation. Computers can describe such number with an n-bit sequence

$$\begin{array}{|c|c|c|c|}\hline x^{(1)} & x^{(2)} & \cdots & x^{(n)} \\ \hline \end{array} \tag{11}$$

that represents the fraction part. The truth values described in this format are 0, Δ, 2Δ, 3Δ, $\cdots$, $1-2\Delta$, $1-\Delta$ that are 2^n points extracted from $[0,1)$ with interval $\Delta = 2^{-n}$. For example, Δ is 0.0000152587890625 when $n = 16$. Although truth values cannot achive exactly 1, the maximum $1-\Delta$ approaches 1 by increasing n. In this sense, we denote $1-\Delta$ by 1_n, e.g.

$$1_{16} \;=\; 0.9999847412109375\,. \tag{12}$$

(Note that, when $n = 1$, the range of truth values is not $\{0,1\}$, but $\{0, 0.5\}$. Thus the proposed multivalued logic system is not a straightforward extension of the Boolean binary logic system.)

In addition to the above five logic operations, for efficient handling of Bayesian theories (cf. [1]), we introduce an arithmetic operation called *conditional*. The conditional from a logic formula x to another logic formula y is

$$y\,|\,x \;=\; \frac{x\wedge y}{x}\,. \tag{13}$$

As a noticeable property, since

$$x\wedge(x\wedge y) \;=\; x\wedge y \tag{14}$$

and

$$\begin{aligned} x\wedge(x\rightarrow y) &= x\wedge(\neg x\vee y) \\ &= (x\wedge\neg x)\vee(x\wedge y) \\ &= x\wedge y\,, \end{aligned} \tag{15}$$

we have

$$(x\wedge y)\,|\,x \;=\; (x\rightarrow y)\,|\,x \;=\; y\,|\,x\,. \tag{16}$$

This suggests that the proposed multivalued logic system has a rich vocabulary that can reflect delicate nuances in natural languages and, simultaneously, can keep a mathematical consistency.

B. Boolean Properties

Since each of Boolean binary conjunction and disjunction satisfies a commutative law, also each of multivalued conjunction $\wedge$ and disjunction $\vee$ defined in the preceding subsection satisfies a *commutative law*. Since each of Boolean binary conjunction and disjunction satisfies an associative law, also each of multivalued conjunction $\wedge$ and disjunction $\vee$ satisfies an *associative law*. Further, since Boolean binary conjunction and disjunction satisfy absorption laws, also multivalued conjunction $\wedge$ and disjunction $\vee$ satisfy *absorption laws*. Thus the whole set of logic formulas forms a *lattice* with multivalued conjunction $\wedge$ and disjunction $\vee$ regarded respectively as "meet" and "join" in terms of describing the lattice.

Further, since

$$x\wedge\neg x \;=\; 0.00\cdots0_{\mathrm{b}} \;=\; 0 \tag{17}$$

and

$$x\vee\neg x \;=\; 0.11\cdots1_{\mathrm{b}} \;=\; 1_n \tag{18}$$

for an arbitrary logic formula x and its *complement* $\neg x$, the whole set of logic formulas forms a *complemented lattice* with *identity elements* 0 and 1_n.

Since Boolean binary conjunction and disjunction satisfy distributive laws, also multivalued conjunction $\wedge$ and disjunction $\vee$ satisfy *distributive laws*. Thus the whole set of logic formulas forms a *distributive lattice*.

Since the whole set of logic formulas is both a complemented lattice and a distributive lattice, it forms a *Boolean algebra*. In this sense, we call the proposed logic system a *Boolean multivalued logic system*.

III. INFERENCES

This section shows some typical examples demonstrating that, on the proposed logic system, after learning we can handle inferences (cf. [2]) readily because of Boolean effect.

See Fig. 1 for the flow of learning and inferences.

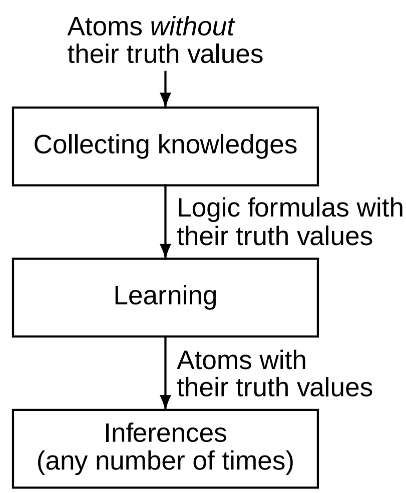

Fig. 1. Flow of learning and inferences.

A. Knowledges

In this article, we define a *knowledge* $x \approx r$ as a constraint such that: the left hand side x is a logic formula; the right hand side r is a desired truth value in $[0,1]$; and, both sides are connected by "$\approx$" meaning that the logic formula x should take the desired truth value r.

For example, we consider the following set of eleven knowledges consisting of seven atoms, *falcon*, *pigeon*, *penguin*, *bird*, *fly*, *good_reflexes*, *ride_bicycle*:

$$\begin{aligned} \textit{bird} \,|\, \textit{falcon} &\approx 1\,, &(19)\\ \textit{fly} \,|\, \textit{falcon} &\approx 0.95\,, &(20)\\ \textit{bird} \,|\, \textit{pigeon} &\approx 1\,, &(21)\\ \textit{fly} \,|\, \textit{pigeon} &\approx 0.95\,, &(22)\\ \textit{good\_reflexes} \,|\, \textit{fly} &\approx 0.8\,, &(23)\\ \textit{bird} \,|\, \textit{penguin} &\approx 1\,, &(24)\\ \textit{fly} \,|\, \textit{penguin} &\approx 0\,, &(25)\\ \textit{falcon} \wedge \textit{ride\_bicycle} &\approx 0\,, &(26)\\ \textit{falcon} &\approx 0.5\,, &(27)\\ \textit{pigeon} &\approx 0.5\,, &(28)\\ \textit{penguin} &\approx 0.25\,. &(29) \end{aligned}$$

Semantic interpretation of (19)–(29) is as follows.

Knowledge (19) means that if observed one is a falcon, it is absolutely ($= 1$) a bird. Also knowledges (21) and (24) have similar meanings.

Knowledge (20) means that if observed one is a falcon, it almost ($= 0.95$) flies — an injured falcon may not fly. Also knowledge (22) has a similar meaning.

Knowledge (25) means that if observed one is a penguin, it never ($= 0$) flies.

Knowledge (23) means that if observed one flies, it probably ($= 0.8$) has good reflexes.

Although knowledges are describable efficiently with the conditional, they do not necessarily consist of the conditional, and the following is such an example. Knowledge (26) means that there is no ($= 0$) one which is a falcon and, simultaneously, rides a bicycle. Note that this formalization differs from

$$\textit{ride\_bicycle} \,|\, \textit{falcon} \approx 0\,, \qquad (30)$$

but (26) and (30) in effect are equivalent unless $\textit{falcon} = 0$ (cf. (16)).

Knowledge (27) simply means that observed one may ($= 0.5$) be a falcon or not. (Note that truth values are not necessarily empirical probabilities.) Also knowledge (28) has a similar meaning. On the other hand, knowledge (29) means that observing a penguin is unusual ($= 0.25$).

B. Learning

Temporarily supposing a set of truth values of atoms, we can calculate the truth value of an arbitrary logic formula, and hence can calculate the truth value of x for each knowledge $x \approx r$. The gap is the absolute value $|x - r|$ of the subtraction of the desired truth value r from the calculated truth value of x, which should be as small as possible. *Learning* is the process of searching for the truth values of atoms so as to minimize the gaps concerning all knowledges. It is a kind of mathematical programming problem, to which algorithms of linear programming [3] and other methods are applicable.

The following shows the truth values of atoms obtained by a trial-and-error method that tries to minimize locally the deviation of gaps concerning all knowledges (19)–(29):

$$\begin{aligned} \textit{falcon} &= 0.1000\ 0011\ 1110\ 1111_{\mathrm{b}}\,, &(31)\\ \textit{pigeon} &= 0.1000\ 0011\ 1111\ 1111_{\mathrm{b}}\,, &(32)\\ \textit{penguin} &= 0.0100\ 0010\ 1000\ 1000_{\mathrm{b}}\,, &(33)\\ \textit{bird} &= 0.1100\ 0011\ 1111\ 0000_{\mathrm{b}}\,, &(34)\\ \textit{fly} &= 0.1001\ 1100\ 0000\ 0000_{\mathrm{b}}\,, &(35)\\ \textit{good\_reflexes} &= 0.1010\ 0000\ 1110\ 0011_{\mathrm{b}}\,, &(36)\\ \textit{ride\_bicycle} &= 0.0011\ 1001\ 0001\ 0111_{\mathrm{b}}\,, &(37) \end{aligned}$$

that is,

$$\begin{aligned} \textit{falcon} &= 0.5153656005859375\,, &(38)\\ \textit{pigeon} &= 0.5156097412109375\,, &(39)\\ \textit{penguin} &= 0.2598876953125\,, &(40)\\ \textit{bird} &= 0.765380859375\,, &(41)\\ \textit{fly} &= 0.609375\,, &(42)\\ \textit{good\_reflexes} &= 0.6284637451171875\,, &(43)\\ \textit{ride\_bicycle} &= 0.2230072021484375 &(44) \end{aligned}$$

in decimal. In this case, the calculated truth values of logic formulas (19)–(29) are as follows, where the slant numerals denote the desired truth values:

$$\begin{aligned} \textit{bird} \,|\, \textit{falcon} &= 0.9995574951171875\\ &= \mathit{1} - 0.0004425048828125, &(45)\\ \textit{fly} \,|\, \textit{falcon} &= 0.970184326171875\\ &= \mathit{0.95} + 0.020184326171875, &(46)\\ \textit{bird} \,|\, \textit{pigeon} &= 0.9995574951171875\\ &= \mathit{1} - 0.0004425048828125, &(47)\\ \textit{fly} \,|\, \textit{pigeon} &= 0.9697265625\\ &= \mathit{0.95} + 0.0197265625, &(48)\\ \textit{good\_reflexes} \,|\, \textit{fly} &= 0.8205108642578125\\ &= \mathit{0.8} + 0.0205108642578125, &(49)\\ \textit{bird} \,|\, \textit{penguin} &= 0.9995269775390625\\ &= \mathit{1} - 0.0004730224609375, &(50)\\ \textit{fly} \,|\, \textit{penguin} &= 0 \;=\; \mathit{0} \pm 0, &(51)\\ \textit{falcon} \wedge \textit{ride\_bicycle} &= 0.0040130615234375\\ &= \mathit{0} + 0.0040130615234375, &(52)\\ \textit{falcon} &= 0.5153656005859375\\ &= \mathit{0.5} + 0.0153656005859375, &(53)\\ \textit{pigeon} &= 0.5156097412109375\\ &= \mathit{0.5} + 0.0156097412109375, &(54)\\ \textit{penguin} &= 0.2598876953125\\ &= \mathit{0.25} + 0.0098876953125\,. &(55) \end{aligned}$$

The deviation of gaps is 0.0128328134040519 and the maximum in gaps is 0.0205108642578125 for (49).

C. Inductive Inferences

After learning, that is, optimizing the truth values of atoms (31)–(37), we can calculate the truth value of an arbitrary logic formula. *Inferences* are processes of simply calculating truth values of logic formulas after the learning. (Note that inferences on many logic systems need complex processes, e.g. Prolog.) This flow shown in Fig. 1 is common to the inductive inferences and the deductive inferences.

The followings are some examples of inductive inferences.

From (31)–(34),

$$\begin{aligned}
&\textit{falcon} \vee \textit{pigeon} \vee \textit{penguin} \\
=\;& 0.1000\,0011\,1110\,1111_{\mathrm{b}} \;\vee\; 0.1000\,0011\,1111\,1111_{\mathrm{b}} \\
&\vee 0.0100\,0010\,1000\,1000_{\mathrm{b}} \\
=\;& 0.1100\,0011\,1111\,1111_{\mathrm{b}}\,, \qquad (56)
\end{aligned}$$

hence

$$\begin{aligned}
&(\textit{falcon} \vee \textit{pigeon} \vee \textit{penguin}) \,|\, \textit{bird} \\
=\;& 0.1100\,0011\,1111\,1111_{\mathrm{b}} \,|\, 0.1100\,0011\,1111\,0000_{\mathrm{b}} \\
=\;& \frac{0.1100\,0011\,1111\,1111_{\mathrm{b}} \wedge 0.1100\,0011\,1111\,0000_{\mathrm{b}}}{0.1100\,0011\,1111\,0000_{\mathrm{b}}} \\
=\;& \frac{0.1100\,0011\,1111\,0000_{\mathrm{b}}}{0.1100\,0011\,1111\,0000_{\mathrm{b}}} \;=\; 1\,, \qquad (57)
\end{aligned}$$

which means that a bird is absolutely one of falcon, pigeon, and penguin. (Note that falcon, pigeon, and penguin are given as samples of birds in knowledges (19)–(29).)

From (34) and (35),

$$\begin{aligned}
&\textit{fly} \,|\, \textit{bird} \\
=\;& 0.1001\,1100\,0000\,0000_{\mathrm{b}} \,|\, 0.1100\,0011\,1111\,0000_{\mathrm{b}} \\
=\;& \frac{0.1100\,0011\,1111\,0000_{\mathrm{b}} \wedge 0.1001\,1100\,0000\,0000_{\mathrm{b}}}{0.1100\,0011\,1111\,0000_{\mathrm{b}}} \\
=\;& \frac{0.1000\,0000\,0000\,0000_{\mathrm{b}}}{0.1100\,0011\,1111\,0000_{\mathrm{b}}} \;=\; 0.6532\cdots\,, \qquad (58)
\end{aligned}$$

which means that birds, if anything, fly.

Let "$\oplus$" mean the *exclusive-or* operation defined as

$$x \oplus y \;=\; \neg(x \leftrightarrow y) \;=\; (x \wedge \neg y) \vee (\neg x \wedge y) \qquad (59)$$

for arbitrary logic formulas x and y. From (34) and (35),

$$\begin{aligned}
\textit{bird} \oplus \textit{fly} \;=\;& 0.1100\,0011\,1111\,0000_{\mathrm{b}} \\
&\oplus 0.1001\,1100\,0000\,0000_{\mathrm{b}} \\
=\;& 0.0101\,1111\,1111\,0000_{\mathrm{b}}\,, \qquad (60)
\end{aligned}$$

hence

$$\begin{aligned}
&(\textit{bird} \oplus \textit{fly}) \,|\, \textit{bird} \\
=\;& 0.0101\,1111\,1111\,0000_{\mathrm{b}} \,|\, 0.1100\,0011\,1111\,0000_{\mathrm{b}} \\
=\;& \frac{0.1100\,0011\,1111\,0000_{\mathrm{b}} \wedge 0.0101\,1111\,1111\,0000_{\mathrm{b}}}{0.1100\,0011\,1111\,0000_{\mathrm{b}}} \\
=\;& \frac{0.0100\,0011\,1111\,0000_{\mathrm{b}}}{0.1100\,0011\,1111\,0000_{\mathrm{b}}} \;=\; 0.3467\cdots\,, \qquad (61)
\end{aligned}$$

which means that being birds and flying is, if anything, not different in birds.

From (34) and (37),

$$\begin{aligned}
&\textit{ride_bicycle} \,|\, \textit{bird} \\
=\;& 0.0011\,1001\,0001\,0111_{\mathrm{b}} \,|\, 0.1100\,0011\,1111\,0000_{\mathrm{b}} \\
=\;& \frac{0.1100\,0011\,1111\,0000_{\mathrm{b}} \wedge 0.0011\,1001\,0001\,0111_{\mathrm{b}}}{0.1100\,0011\,1111\,0000_{\mathrm{b}}} \\
=\;& \frac{0.0000\,0001\,0001\,0000_{\mathrm{b}}}{0.1100\,0011\,1111\,0000_{\mathrm{b}}} \;=\; 0.0054\cdots\,, \qquad (62)
\end{aligned}$$

which means that birds hardly ride bicycles.

From (36) and (37),

$$\begin{aligned}
&\textit{ride_bicycle} \,|\, \textit{good_reflexes} \\
=\;& 0.0011\,1001\,0001\,0111_{\mathrm{b}} \,|\, 0.1010\,0000\,1110\,0011_{\mathrm{b}} \\
=\;& \frac{0.1010\,0000\,1110\,0011_{\mathrm{b}} \wedge 0.0011\,1001\,0001\,0111_{\mathrm{b}}}{0.1010\,0000\,1110\,0011_{\mathrm{b}}} \\
=\;& \frac{0.0010\,0000\,0000\,0011_{\mathrm{b}}}{0.1010\,0000\,1110\,0011_{\mathrm{b}}} \;=\; 0.1989\cdots\,, \qquad (63)
\end{aligned}$$

which means that having good reflexes probably does not ensure riding a bicycle.

D. Complementary Law as a Deductive Inference

Any of the laws, the theorems, and other properties obtained on a traditional Boolean binary logic system are valid also on the proposed logic system as implied by its bitwise processing in logic operations.

A complementary law is

$$\neg x \;=\; 1_n - x \qquad (64)$$

for an arbitrary logic formula $x = 0.x^{(1)}x^{(2)}\cdots x^{(n)}{}_{\mathrm{b}}$, which is obtained from $\neg x^{(i)} = 1 - x^{(i)}$ for every $i = 1, 2, \cdots, n$.

For example, when (31)–(37), we have

$$\begin{aligned}
&\neg\,\textit{falcon} \\
=\;& \neg\,0.1000\,0011\,1110\,1111_{\mathrm{b}} \\
=\;& 0.0111\,1100\,0001\,0000_{\mathrm{b}} \;=\; 0.484619140625 \\
=\;& 0.9999847412109375 - 0.5153656005859375 \\
=\;& 1_{16} - \textit{falcon} \qquad (65)
\end{aligned}$$

that verifies (64).

E. Chain Rule as a Deductive Inference

A chain rule is

$$(x \to y) \wedge (y \to z) \to (x \to z) \;=\; 1_n \qquad (66)$$

for arbitrary logic formulas $x = 0.x^{(1)}x^{(2)}\cdots x^{(n)}{}_{\mathrm{b}}$, $y = 0.y^{(1)}y^{(2)}\cdots y^{(n)}{}_{\mathrm{b}}$, and $z = 0.z^{(1)}z^{(2)}\cdots z^{(n)}{}_{\mathrm{b}}$ on a Boolean multi-valued logic system, which suggests

$$(x \to y) \wedge (y \to z) \;\le\; x \to z\,. \qquad (67)$$

Indeed, we have the following for $i = 1, 2, \cdots, n$.

$x^{(i)}$	0	0	0	0	1	1	1	1
$y^{(i)}$	0	0	1	1	0	0	1	1
$z^{(i)}$	0	1	0	1	0	1	0	1
$x^{(i)} \to y^{(i)}$	1	1	1	1	0	0	1	1
$y^{(i)} \to z^{(i)}$	1	1	0	1	1	1	0	1
$(x^{(i)} \to y^{(i)}) \wedge (y^{(i)} \to z^{(i)})$	1	1	0	1	0	0	0	1
$x^{(i)} \to z^{(i)}$	1	1	1	1	0	1	0	1

(68)

The last two lines prove

$$(x^{(i)} \to y^{(i)}) \wedge (y^{(i)} \to z^{(i)}) \;\leq\; x^{(i)} \to z^{(i)} \tag{69}$$

for $i = 1, 2, \cdots, n$, hence (67), no matter what x, y, and z are.

For example, when (31)–(37), since

$$\begin{aligned} & (\textit{falcon} \to \textit{fly}) \wedge (\textit{fly} \to \textit{good\_reflexes}) \\ = \;& 0.1110\;0000\;0001\;0000_{\mathrm{b}} \\ = \;& 0.875244140625 \end{aligned} \tag{70}$$

and

$$\begin{aligned} \textit{falcon} \to \textit{good\_reflexes} \;&=\; 0.1111\;1100\;1111\;0011_{\mathrm{b}} \\ &=\; 0.9880828857421875\,, \end{aligned} \tag{71}$$

we have

$$\begin{aligned} & (\textit{falcon} \to \textit{fly}) \wedge (\textit{fly} \to \textit{good\_reflexes}) \\ \leq \;& \textit{falcon} \to \textit{good\_reflexes} \end{aligned} \tag{72}$$

that verifies (67).

F. Bayesian Tautology as a Deductive Inference

A tautology modeled on a Bayesian network, which we call a *Bayesian tautology* in this article, is

$$\begin{aligned} & ((x \to y) \wedge (y \to z)) \vee ((x \to \neg y) \wedge (\neg y \to z)) \\ & \leftrightarrow\; (x \to z) \\ = \;& 1_n \end{aligned} \tag{73}$$

for arbitrary logic formulas $x = 0.x^{(1)}x^{(2)} \cdots x^{(n)}{}_{\mathrm{b}}$, $y = 0.y^{(1)} y^{(2)} \cdots y^{(n)}{}_{\mathrm{b}}$, and $z = 0.z^{(1)}z^{(2)} \cdots z^{(n)}{}_{\mathrm{b}}$, that is,

$$\begin{aligned} & ((x \to y) \wedge (y \to z)) \vee ((x \to \neg y) \wedge (\neg y \to z)) \\ = \;& x \to z\,. \end{aligned} \tag{74}$$

Indeed, we have the following for $i = 1, 2, \cdots, n$.

$x^{(i)}$	0	0	0	0	1	1	1	1
$y^{(i)}$	0	0	1	1	0	0	1	1
$z^{(i)}$	0	1	0	1	0	1	0	1
$\neg y^{(i)}$	1	1	0	0	1	1	0	0
$x^{(i)} \to \neg y^{(i)}$	1	1	1	1	1	1	0	0
$\neg y^{(i)} \to z^{(i)}$	0	1	1	1	0	1	1	1
$(x^{(i)} \to \neg y^{(i)}) \wedge (\neg y^{(i)} \to z^{(i)})$	0	1	1	1	0	1	0	0
$(x^{(i)} \to y^{(i)}) \wedge (y^{(i)} \to z^{(i)})$	1	1	0	1	0	0	0	1
$x^{(i)} \to z^{(i)}$	1	1	1	1	0	1	0	1

(75)

The last three lines prove

$$\begin{aligned} & ((x^{(i)} \to y^{(i)}) \wedge (y^{(i)} \to z^{(i)})) \\ & \vee ((x^{(i)} \to \neg y^{(i)}) \wedge (\neg y^{(i)} \to z^{(i)})) \\ = \;& x^{(i)} \to z^{(i)} \end{aligned} \tag{76}$$

for $i = 1, 2, \cdots, n$, hence (74), no matter what x, y, and z are.

For example, when (31)–(37), since

$$\begin{aligned} & ((\textit{falcon} \to \textit{fly}) \wedge (\textit{fly} \to \textit{good\_reflexes})) \\ & \vee ((\textit{falcon} \to \neg\textit{fly}) \wedge (\neg\textit{fly} \to \textit{good\_reflexes})) \\ = \;& 0.1111\;1100\;1111\;0011_{\mathrm{b}} \\ = \;& 0.9880828857421875 \end{aligned} \tag{77}$$

and

$$\begin{aligned} \textit{falcon} \to \textit{good\_reflexes} \;&=\; 0.1111\;1100\;1111\;0011_{\mathrm{b}} \\ &=\; 0.9880828857421875\,, \end{aligned} \tag{78}$$

we have

$$\begin{aligned} & ((\textit{falcon} \to \textit{fly}) \wedge (\textit{fly} \to \textit{good\_reflexes})) \\ & \vee ((\textit{falcon} \to \neg\textit{fly}) \wedge (\neg\textit{fly} \to \textit{good\_reflexes})) \\ = \;& \textit{falcon} \to \textit{good\_reflexes} \end{aligned} \tag{79}$$

that verifies (74).

The only difference between the Boolean binary logic system and the proposed logic system is that the former can take two truth values and the latter can take 2^n truth values. But this difference is crucial since being multivalued widens applications of Boolean logic systems essentially.

IV. Expansion to Fuzzy

A typical multivalued logic system extended from Kleene's three-valued logic system [4] is defined as $\neg x = 1 - x$, $x \wedge y = \min(x, y)$, and $x \vee y = \max(x, y)$ for logic formulas x and y having truth values in $[0, 1]$, on which the set of logic formulas forms no Boolean algebra. Fuzzy logic systems usually are based on such multivalued logic systems in non-Boolean class.

A fuzzy logic system means an arbitrary logic system defined with terms of *fuzzy subsets* (cf. [5], [6]), where a fuzzy subset is denoted by $\{a_1\!:t_1,\ a_2\!:t_2,\ \cdots,\ a_n\!:t_n\}$ for members $a_1, a_2, \cdots, a_n$ and membership values $t_1, t_2, \cdots, t_n$. We can define a new fuzzy logic system by regarding Boolean multivalued logic formulas as members and their truth values as membership values, which we call a *Boolean fuzzy logic system*.

This fuzzy logic system in contrast with usual fuzzy logic systems inherits all the properties obtained on traditional Boolean binary logic systems, which corresponds to a bitwise-processable realization of complementary fuzzy logic system [7], [8].

V. Conclusion

This article, by regarding fixed-point binary numbers as truth values, defined a new multivalued logic system in Boolean class such that the set of logic formulas forms a Boolean algebra and that, after learning, we can handle inferences readily because of Boolean effect.

A bottleneck is that the learning needs a large computational complexity if the number of atoms or that of knowledges is large.

REFERENCES

[1] Rudolf Carnap, *Logical Foundations of Probability*, The Univ. of Chicago, 1967.

[2] Wesley C. Salmon, "Confirmation," *Scientific American*, pp. 75–83, May 1973.

[3] George B. Dantzig, *Linear Programming and Extensions*, Princeton Univ. Press, 1963.

[4] Stephen C. Kleene, *Introduction to Metamathematics*, New York: Van Nostrand, 1952.

[5] Lotfi A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[6] George J. Klir and Tina A. Folger, *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall, Jan. 1988.

[7] Hisashi Suzuki and Suguru Arimoto, "Embedment of a fuzzy logic system into a boolean lattice for satisfying a complementary law," *Information Sciences; An International Journal*, vol. 78, pp. 257–268, 1994.

[8] Hisashi Suzuki, "A complementary fuzzy logic system," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 27, no. 2, pp. 293–295, Apr. 1997.