

# Trilattice-Based Access Control Models: How to Secure Current Computer Network

Mikhail M. Kucherov\* and Nina A. Bogulskaya

Siberian Federal University, School of Space and Information Technologies, 660074 26B Kirenskogo Krasnoyarsk, Russia

**Abstract.** Designing security, from the hardware level, is essential to ensure the integrity of the intelligent cyber-physical infrastructure that is the Industrial Internet of Things (IIoT). If intelligent cyber-physical infrastructure fails to do the right things because it is insecure and vulnerable, then there will be negative social consequences [1]. Security is, in a sense, the access control to IIoT systems, which increasingly relies on the ability to compose different policies. Therefore, the advantage in any framework for compiling policies is that it is intuitive, formal, expressive, application-independent, as well as expandable to create domain-specific instances. Recently, such a scheme was proposed based on Belnap logic  $FOUR_2$  [2]. Four values of the Belnap bilattice have been interpreted as *grant*, *deny*, *conflict*, or *unspecified* with respect to access-control policy. Belnap's four-valued logic has found a variety of applications in various fields, such as deductive database theory, distributed logic programming, and other areas. However, it turns out that the truth order in  $FOUR_2$  is a truth-and-falsity order at the same time [3]. The smallest lattice, where the orders of truth and falsity are independent of each other, which is especially important for security policy, is that of Shramko-Wansing's  $SIXTEEN_3$ . This generalization is well-motivated and leads from the bilattice  $FOUR_2$  with an information and a truth-and-falsity ordering to another algebraic structure, namely the trilattice  $SIXTEEN_3$  with an information ordering together with a truth ordering and a (distinct) falsity ordering. Based on  $SIXTEEN_3$  and new Boolean predicates to control access [4], we define an expressive access-control policy language, having composition statements based on the statements of Schramko-Wansing's logic. Natural orderings on policies are obtained by independent lifting the orders of truth and falsity of trilattice, which results in a query language in which conflict freedom analysis can be developed. The reduction of formal verification of queries to that on predicates over access requests enables to carry out policy analysis. We evaluate our approach through examples of control access model policy.

## 1 Introduction

By access control we understand methods or mechanisms that decide whether requests to access some resource should be granted or denied. For example, operating systems need to control which users and applications can read, write, or delete which files; networks need to govern which packets can pass through a physical or logical perimeter; and managers need to control which employees can perform which workflows within an organization.

We take responses of an access control system to be the values 'T, F, Both, None', according to Belnap logic. The meaning of these values can be described as follows:

- an atomic sentence is stated to be true only (T)
- an atomic sentence is stated to be false only (F)
- an atomic sentence is stated to be both true and false, for instance, by different sources, or in different points of time (Both), and
- an atomic sentence status is unknown. That is, neither true, nor false (None).

These values and their two orderings form a distributive, interlaced bilattice [2, 3].

Thus, bilattice  $FOUR_2$  has many convenient properties, for example:  $\wedge$  and  $\vee$  distribute as in propositional logic, as do  $\otimes$  and  $\oplus$ , and all these operators are also monotone to both the information and logic orderings.

In this paper we extend use of multivalued logics for dealing with incomplete and inconsistent information to  $SIXTEEN_3$  [5, 10]. The five information levels, five logical levels and five levels of falsity in trilattice of sixteen truth values are arranged in three meet-distributive lattices. One of the interesting observations involves computation of '*an access is granted*  $\wedge$  *an access is not rejectable*'. The conclusion is based on the rule that a conjunction is true if and only if both conjuncts are true. Since there is no truth value such that both *an access is granted*, and *an access is not rejectable* has it, the result is the empty set.

*Motivation.* Imagine two access control policies, each of which reports a Boolean representing whether an access should be permitted or not. We can form four possible sets by collecting the statements: {grant}, *conflict*– {grant, deny}, {deny}, and *undefined*– {}. For example, the set {grant, deny} arises when one policy permits the access and the other denies it. The empty set

\* Corresponding author: [mkucherov@sfu-kras.ru](mailto:mkucherov@sfu-kras.ru)

$\{\}$  arises when the access is outside the domain of both policies. We can order these sets in two different ways. If we order by the degree to which the access is permitted, we have that  $\{\text{grant}\}$  is greatest and  $\{\text{deny}\}$  is least. We can also order by the amount of information we have obtained. In this ordering *conflict* is greatest and *undefined* is least. We write  $\leq_t$  for the truth ordering and  $\leq_k$  for the knowledge (information) ordering.

But if we give formal definitions for the partial order relations ( $\leq_k$  and  $\leq_t$ ), we get following [3]: The definition of  $\leq_k$  is very simple: for any  $x, y \in \text{FOUR}_2$ ,  $x \leq_k y$  iff  $x \subseteq y$ . With the  $\leq_t$  ratio, the situation is not so trivial. If we define for each element of  $\text{FOUR}_2$  its "truth-containing part" and "lie-containing part":  $x^t := \{z \in x \mid z = T\}$ ;  $x^f := \{z \in x \mid z = F\}$ , then we can make sure that:  $x \leq_t y$  iff  $x^t \subseteq y^t$  and  $y^f \subseteq x^f$ . This suggests that truth and lies in Belnap's logic are not completely independent concepts: Belnap's logic suggests that the very meaning of "lie" is not only less true than the meaning of "truth," but also the meaning of "neither truth, not a lie".

Next consider the issue of access control policies containing both "permit rules" and "deny rules". A permit rule returns *true*<sup>(+)</sup>, if the access is granted, and *false*<sup>(+)</sup>, if the access is not granted (but not obligatorily rejectable). Deny rules work symmetrically, i.e. return *true*<sup>(-)</sup>, if the access is denied, and *false*<sup>(-)</sup>, if there is no explicit denial (but is not granted). Much recent work on access control has considered "permit rules". However, the logic of generalized truth values that comes with trilattice *SIXTEEN*<sub>3</sub> and the propositional language  $L_{\text{tr}}$  allows to solve using "deny rules" many problems in the most natural way. Moreover, if the three partial orderings  $\leq_b \leq_f$  and  $\leq_i$  are used to define orderings on valuations, different kinds of non-monotonic inference relations can be defined in terms of minimizing or maximizing truth, falsity, or information.

*Contributions.* We consider equivalence between different presentations of multivalued data, and obtain a parametric expression for an access policy as a combination of "permit rules" and "deny rules". Since the set *SIXTEEN*<sub>3</sub> is the smallest multivalued lattice having two independent orderings: in true and in falsity, it is necessary to consider the presentation of access models based on the *SIXTEEN*<sub>3</sub>. The role of falsity order in consequence relations important for the control of communication was determined distinctly, which allowed to overcome shortcomings of the use of bilattice *FOUR*<sub>2</sub>.

The rest of the paper is organized as follows. Related work and statements from many-valued logic are discussed in Section 2 and 3, respectively. The main results of the paper are placed in Section 3–8. The paper ends with discussions in Section 9.

## 2 Related works

The notion of a bilattice was first introduced into logic and artificial intelligence by M. Ginsberg and M. Fitting [6, 7]. With respect to deductive databases, their ideas were supported by [5].

A very clear and useful introduction into bilattices and application problems in access control is [8]. In most security systems the management of policies is not considered to include neither explicit nor hidden inconsistencies. In real life situation information often come from different contradicting sources. Thus, different sources can provide inconsistent data while deductive reasoning may result in hidden inconsistencies. The idea in Belnap's approach [9] is to develop a logic that is not dependable on inconsistencies.

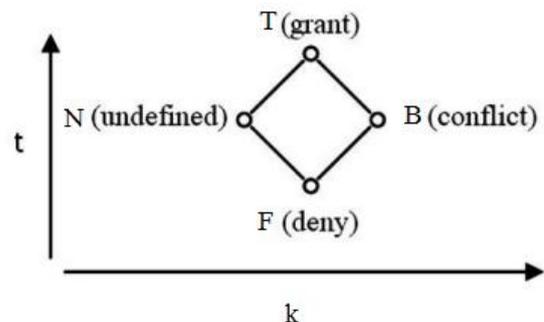
The logic of generalized truth values has been developed in a number of papers by M. Dunn, Y. Shramko, S.P. Odintsov, and H. Wansing, who consider logics defined on trilattices. These investigations have been summarized in [10, 11].

Fuzzy logic is a form of multi-valued logic later popularized as logic of vagueness by Zadeh as was reviewed by [12]. It is based on replacing the standard set of Boolean truth values with a different lattice, most often, like in the present paper, the unit interval. Saying that a formula  $\phi$  has truth value  $r \in [0, 1]$  then means that  $\phi$  holds with degree  $r$ , which would apply to typical vague qualifications such as a given person being tall (in contrast to assigning a probability  $p \in [0, 1]$  to  $\phi$ , which would be read as saying that  $\phi$  is either completely true with probability  $p$  or completely false with probability  $1 - p$ , as in "the die under the cup shows a 3 with probability  $p$ ").

## 3 Meet-distributive lattices

Let  $P$  be a non-empty ordered set. If  $\sup\{x, y\}$  and  $\inf\{x, y\}$  exist for all  $x, y \in P$ , then  $P$  is called a *lattice* [13]. A complete lattice is a partially ordered set in which all subsets have both a supremum (join) and an infimum (meet). A lattice  $L$  is meet-distributive if every coatomic interval is Boolean. A bilattice is a set equipped with two partial orderings  $\leq_t$  and  $\leq_k$ .

The four truth values are arranged in a logical lattice [2] on Figure 1. A logical conjunction and logical disjunction are related to the meet operation and to the join operation respectively.



**Fig. 1.** The Belnap bilattice: with synonyms for access control decisions in parentheses.

Let us consider the following bounded lattices: fuzzy data [14, 15, 16]: then  $C[0, 1]$  is the infinite set of real numbers from 0 to 1. For any ground atom  $r(\mathbf{d}) \in H$  the  $p = I(r(\mathbf{d}))$  represents its *plausibility*.

The quantity  $p$  is determined as a closed subinterval [17,18,19]:  $\mathcal{W} = C[0, 1]$  is the set of all closed subintervals. For any ground atom  $r(\mathbf{d}) \in H$  the  $(L, U) = I_{mv}(r(\mathbf{d}))$  represents the lower and upper bounds for an expert's evaluation of  $r(\mathbf{d})$ . The generalized truth-value space has as a base a set  $I = (T, F, t, f)$  containing the initial truth values:  $T$ —an access is granted,  $1, 1, 0, 1$ ;  $F$ —an access is denied,  $0, 0, 0, 1$ ;  $t$ —an access is not rejectable,  $0, 1, 0, 0$ , and  $f$ —an access is rejectable,  $0, 1, 1, 1$ . The power set  $P(I)$  gives sixteen values with the presentation as closed subintervals in Table 1:

**Table 1.** Levels of trilattice  $SIXTEEN_3$ .

	Information
1	0,1 , 0,1
2	1,1 , 0,1    0,0 , 0,1    0,1 , 0,0    0,1 , 1,1
3	1,0 , 0,1    1,1 , 0,0    1,1 , 1,1    0,0 , 0,0 0,0 , 1,1    0,1 , 1,0
4	1,0 , 0,0    1,0 , 1,1    1,1 , 1,0    0,0 , 1,0
5	1,0 , 1,0

For example,  $\{T, t\} = 1,1, 0,1 \oplus_i 0,1, 0,0 = 1,1, 0,0$  (see *Appendix*).

The three partial orderings  $\leq_i, \leq_v, \leq_f$  arrange elements according to the possessed degree of information, truth and falsity, respectively. The  $\leq_f$  is the falsity order: ‘falsity’ decreases if the access is granted. The sixteen truth values are arranged in the falsity meet-distributive lattice on Figure 2. The details relative to the three partial orderings are shown in *Appendix* (Table 4).

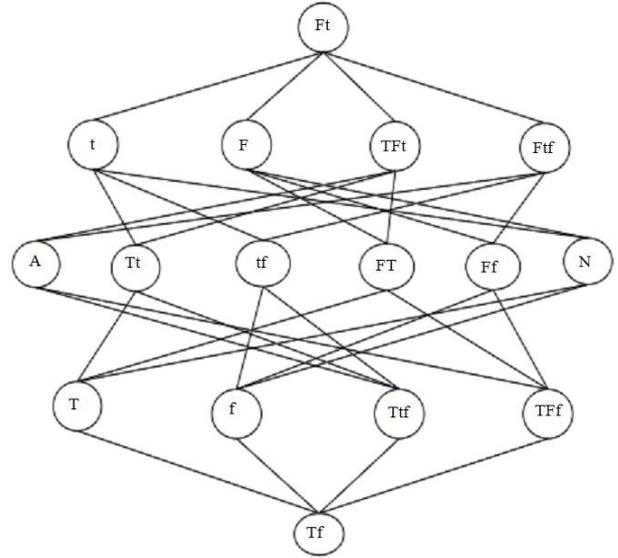
**Table 2.** Policy language: the  $rp$  are access predicates, and  $b \in \{F, T\}$ .

$p ::= ([x,y],[z,v])$	Policy
$b \text{ if } rp$	Basic policy [2]
$([1-y, 1-x], [1-v, 1-z])$	Logical negation
$([\min \{x, x_1\}, \min \{y, y_1\}], [\max \{z, z_1\}, \max \{v, v_1\}])$	Logical meet
$([1-x, 1-y], [1-z, 1-v])$	Falsity negation
$([\max \{x, x_1\}, \max \{y, y_1\}], [\max \{z, z_1\}, \max \{v, v_1\}])$	Disjunction; $p \oplus q$

#### 4 Policies

We take responses of an access control system to be the values  $([x,y],[z,v])$  of the sixteen generalized truth values space, which we write as  $SIXTEEN_3$ . These values

and their three orderings form a distributive, interlaced bilattice.



**Fig. 2.** Falsity meet-distributive lattice.

Consequence relations  $\models_i$  and  $\models_f$  are extensions of implication to  $SIXTEEN_3$ . For each logical order a separate entailment relation between any sentences  $A, B \in L_{if}$  was defined in [20]:

$$A \models_i B \text{ iff } \forall v^{16} v^{16} A \leq_i v^{16} B, \text{ and}$$

$$A \models_f B \text{ iff } \forall v^{16} v^{16} B \leq_f v^{16} A.$$

There it was proved that both of these restrictions coincide in fact with Belnap's four-valued logic.

Expression  $a \supset b$  with  $a, b \in SIXTEEN_3$  yields  $b$  if  $a$  is lessor equal to  $T$  in the falsity ordering and yields  $T$  otherwise. Table 2 gives the syntax of access control policy language. Informally, an expression for every access  $a ::= \{(r(\mathbf{d}), I(r(\mathbf{d}))) \mid r(\mathbf{d}) \in A, \text{ where } A \subseteq H\}$  is interpreted as a mapping from accesses to elements of  $L_{if}$ . If policy  $p$  produces result  $\alpha$  on access  $a$  and policy  $q$  produces result  $\beta$  on  $a$ , then  $p \wedge q$  produces result  $\alpha \wedge \beta$  on  $a$ . The other Belnap operators are similarly interpreted on policies.

The intuition behind these policy operators is that:

- $\neg p$  denies an access iff grants it (and vice versa)
- $p \wedge q$  grants an access iff both  $p$  and  $q$  grant it, and denies an access iff at least one of  $p$  and  $q$  denies it
- $p \supset q$  grants an access iff  $p$  does not grant it or  $q$  does grant it, and  $p \supset q$  denies an access if  $p$  grants it and  $q$  denies it
- $p \oplus q$  grants (resp. denies) an access iff  $p$  or  $q$  grants (resp. denies) it.

The only operator in the policy language not derived from Belnap logic is the basic expression  $b \text{ if } rp$ , where  $b$  ranges over  $\{F, T\}$ , and  $rp$  is an access predicate. Informally, the  $b \text{ if } rp$  ‘rule’ gives result  $b$  for an access satisfying  $rp$  and gives result  $\perp$  otherwise. The idea is that  $rp$  defines the domain of

accesses covered by the rule, so that  $\perp$  is produced on accesses outside of the rule's domain.

## 5 Queries

Properties about policies and their relationships can be expressed as *queries*, propositional formulas in which the atomic propositions concern the truth or falsity ordering among policies. Table 3 gives the syntax for queries.

**Table 3.** Query language:  $p$  and  $q$  range over policy expressions.

$\phi, \psi ::=$	Query
$p \leq q$	Truth ordering
$p \leq_f q$	Falsity ordering
$rp_c(a,b) = (\neg a \vee b)$	Confidentiality request
$rp_i(a,b) = (a \wedge \neg b)$	Integrity request

In [4], we defined expressions for basic policies in a natural way, based on material implication, as a rigorous expression of the information flow operator [21], as logical formulas for some predicate query symbols:

- confidentialityrequest(Bell–LaPadula),
- integrityrequest(Biba),

where  $a$  – the class of confidentiality/integrity of subject, and  $b$  – the class of object ( $a, b \in \{0,1\}$ ).

## 6 Example for analysis

Access control refers to mechanism by which the access of principals to resources is regulated, as can be seen in the following example of the access referring to the transport resources.

Generally, access control is implemented by an authorization service, which includes an authorization decision function (ADF) for deciding whether a user request to access a resource should be permitted or not. The output of ADF is usually determined by evaluating the request with respect to *authorization state*. The *authorization policy*, which is implemented by the ADF, is to authorize a request if it is listed in the authorization state [22].

A group of students of School of Space and Information Technologies in Krasnoyarsk is planning a trip to see the Pillars Nature Sanctuary (Stolby). Normally, it takes about one hour to get to the Right Bank across the Yenisei River by car, but the students know that no-one of locals needs to be told that Kopylovski Bridge Expressway (KB) is “the longest parking lot”.

The students consult a traffic service, which integrates information from several independent information sources to provide traffic advisory along

various travel routes. They have a combination of several sources. Let us assume that these sources are:

- weather forecast (rain, snow, fog);
- social activity;
- police activity;
- road reparation.

The service uses the following rules to generate advisories:

If the weather is bad, and there is road reparation along the route, the probability of a delay is 0.9.

If there is road reparation, and social activities along the route, the likelihood of a delay is 0.8.

If there is road reparation, and police activities along the route, the likelihood of a delay is 0.99.

These rules are expressed as levels of evaluation, as  $[0.9, 1]$ ,  $[0.8, 1]$  and  $[0.99, 1]$ . The service generates advisories expressed as the likelihood of delays along the routes of interest (*auth state*). Students do not want to miss the excursion due to traffic, but they also have conference deadlines and so do not want to leave too early.

They decide that if the advisory says that the likelihood of delays is between 0.2 and 0.4, then they add one extra hour to the trip time. If the likelihood is between 0.4 and 0.6, then they add two hours, and if the likelihood is over 0.6 then they take a river-train (*auth policy*).

In Section 8, we will return to this example and show that our approach improves the quality of the advisory and could have helped the students avoid unnecessary trouble.

## 7 Combining answers

We can learn levels for compound events, which include basic objects, having the values for basic objects.

### 7.1 Case 1: Independence

Since in the model an event can be true, false, or unknown, (i.e., we are modeling different factors independently) we have:

$$Cl_1(A \otimes_t B) = ([xx_1, yy_1], [1 - (1-z)(1-z_1), 1 - (1-v)(1-v_1)]).$$

$$Cl_1(A \oplus_t B) = ([1 - (1-x)(1-x_1), 1 - (1-y)(1-y_1)], [zz_1, vv_1]). \quad (1)$$

### 7.2 Case 2: Positive correlation

Two events  $A$  and  $B$  are positively correlated if they overlap as much as possible. One can obtain that the solution can be shown to be

$$Cl_2(A \otimes_t B) = ([\min\{x, x_1\}, \min\{y, y_1\}], [\max\{z, z_1\}, \max\{v, v_1\}]).$$

$$Cl_2(A \oplus_t B) = ([\max\{x, x_1\}, \max\{y, y_1\}], [\min\{z, z_1\}, \min\{v, v_1\}]) \quad (2)$$

## 8 Example for analysis (continued)

Returning to the example in Section 7, suppose that our information sources predict 50% chance of bad weather, parades with 50% certainty, roadwork along the Kopylovski Bridge with certainty 80%, and police activity with the probability of 40%. This information is expressed in this way:

- roadwork (RW):([0.8, 0.8], [0.2, 0.2]);
- social\_act (SA): ([0.5, 0.5], [0.5, 0.5]);
- bad\_weather (BW): ([0.5, 0.5], [0.5, 0.5]);
- police\_act (PA): ([0.4, 0.4], [0.6, 0.6]).

The traffic service fetches the above information from four different information sources and integrates them independently using these rules as in Eq. (1). Here, the grant and rejection are mutually complementary:

- delay, if there are roadworks and bad weather, for example:

$$Cl_1(RW \otimes_i BW) = ([0.9 \times 0.8 \times 0.5, 0.8 \times 0.5], [0.6, 0.64]).$$

Eventually, one can obtain

$$Cl_1(RW \otimes_i BW) = ([0.36, 0.4], [0.6, 0.64]) \quad (3)$$

- delay, if there are roadworks and the marathon:

$$Cl_1(RW \otimes_i SA) = ([0.32, 0.4], [0.6, 0.68]) \quad (4)$$

If there are road works and traffic accident:

$$Cl_1(RW \otimes_i PA) = ([0.32, 0.32], [0.68, 0.68]) \quad (5)$$

Using the second rule of Eq. (1) one can obtain from Eqs. (5) and (6):

$$Cl_{tot} [(RW \otimes_i BW) \oplus_i (RW \otimes_i SA)] = ([0.56, 0.64], [0.36, 0.44]). \quad (6)$$

Adding Eq. (7) one has finally:

$$([0.7, 0.76], [0.24, 0.3]) \quad (7)$$

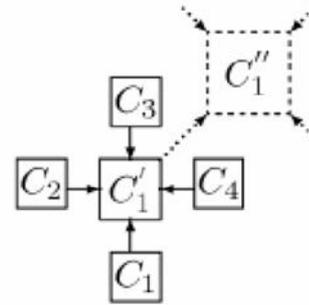
When correlation is not considered, the evaluation of delay(KB) is [0.36, 1], which means that the available information predicts traffic delay with certainty 0.36 and smooth traffic with certainty [0, 0.64]. Considering the possibilities of parades and accidents, it is reasonable to up the expectation of delays.

In contrast, our method computes the confidence level for traffic delays to be [0.7, 0.76], which is narrower than the certainty factor according belief logic programming theory [0.63, 1] [23]. Thus, we see that the theory can predict better expert's evaluation of the combined information.

## 9 Discussions

In fact, Belnap's interpretation suits perfectly well when applied to a single computer. In addition, it presupposes that this computer receives information from classical sources, i.e., from sources, which can operate exclusively with classical truth values (T, F). And it

appears that Belnap's interpretation cannot adequately be applied when we deal not just with one computer but with several interconnected computers, i.e., a computer network. Indeed, consider four Belnap computers ( $C_1, C_2, C_3, C_4$ ) connected to some central computer ( $C'_1$ ), a server, to which they are supposed to supply information (Figure 3). Incidentally, it is interesting to observe that if we wish to extend our network and connect our server to some "higher" computer ( $C''_1$ ), then generalized truth values of the third order (the set  $P(SIXTEEN_3)$ ) comes into question (and so on).



**Fig. 3.** Computernetwork.

Apparently, we should stop at the 16-digit logic, for example, such as  $SIXTEEN_3$ . If we look at the set  $SIXTEEN_3$  and consider its algebraic structure, it turns out that within the framework of this set it is possible to make an effective distinction between the increase in the truth of the facts and the decrease of their falsity, and therefore one can determine the order in truth and the falsity order as two different and mutually independent orders[3]. Since the set  $SIXTEEN_3$  is the smallest multivalued lattice having this property, it is necessary to refer to the presentation of access models based on the  $SIXTEEN_3$ .

*Future work.* Our most recent work [4] shows that it is possible to obtain expressions for basic policies, based on the information flow operator [21], as logical formulas for some request predicate symbols:

- confidentiality request (Bell–LaPadula),
- integrity request (Biba). We intend to continue our preliminary work in this area.

In our current approach, we hope to develop an implementation of given framework to avoid the immediate resolution of conflicts by the choice of access decisions based on a set  $I = (T, F, t, f)$ . Thirdly, we mean to improve the language description of authorization state using several logical techniques.

In conclusion, we extend a formal framework for authorization policies based on trilattice  $SIXTEEN_3$ ; all possible complex patterns for base policies are supported in the presence of conflict or lack of information; and authorization state and policy specifications are cleanly separated. Policy composition is achieved through combination modes (independence, positive correlation, etc.), which are familiar in the theory of deductive databases. They are formally analyzable and functionally complete for policy coordination. The parameterized

method not only helps on analysis of policies, but these types and their analysis can also certify in the future investigations important run-time behavior of policy evaluation.

## Appendix

### A.1 The algebra of confidence levels

We shall establish the structure and properties of trilattices here, which is used in previous sections.

*Definition 1.* Denote by  $C[0,1]$  the set of all closed subintervals over  $[0,1]$ . Consider the set  $C_c =_{\text{def}} C[0,1] \times C[0,1]$ . We denote the elements of  $C_c$  as  $([x, y][z, v])$ . Define the following orders on this set. Let  $([x, y][z, v]), ([x_1, y_1][z_1, v_1])$  be any two elements of  $C_c$ .

$([x, y][z, v]) \leq_t ([x_1, y_1][z_1, v_1])$  iff  $x \leq x_1, y \leq y_1$  and  $z_1 \leq z, v_1 \leq v$   
 $([x, y][z, v]) \leq_f ([x_1, y_1][z_1, v_1])$  iff  $x \geq x_1, y \geq y_1$  and  $z \geq z_1, v \geq v_1$   
 $([x, y][z, v]) \leq_i ([x_1, y_1][z_1, v_1])$  iff  $x \leq x_1, y_1 \leq y$  and  $z \leq z_1, v_1 \leq v$

*Definition 2.* Let  $\langle C_c, \leq_t, \leq_f, \leq_i \rangle$  be as defined in Definition 1. Then the meet and join corresponding to the truth, falsity and knowledge (information) orders are defined as follows. The symbols  $\otimes$  and  $\oplus$  denote meet and join, and the subscripts  $t, f,$  and  $i$  represent truth, falsity, and information (knowledge), respectively.

1.  $([x, y][z, v]) \otimes_t ([x_1, y_1][z_1, v_1]) =$   
 $([\min\{x, x_1\}, \min\{y, y_1\}], [\max\{z, z_1\}, \max\{v, v_1\}])$
2.  $([x, y][z, v]) \oplus_f ([x_1, y_1][z_1, v_1]) =$   
 $([\max\{x, x_1\}, \max\{y, y_1\}], [\min\{z, z_1\}, \min\{v, v_1\}])$
3.  $([x, y][z, v]) \otimes_i ([x_1, y_1][z_1, v_1]) =$   
 $([\min\{x, x_1\}, \min\{y, y_1\}], [\min\{z, z_1\}, \min\{v, v_1\}])$
4.  $([x, y][z, v]) \oplus_j ([x_1, y_1][z_1, v_1]) =$   
 $([\max\{x, x_1\}, \max\{y, y_1\}], [\max\{z, z_1\}, \max\{v, v_1\}])$
5.  $([x, y][z, v]) \otimes_f ([x_1, y_1][z_1, v_1]) =$   
 $([\min\{x, x_1\}, \max\{y, y_1\}], [\min\{z, z_1\}, \max\{v, v_1\}])$
6.  $([x, y][z, v]) \oplus_t ([x_1, y_1][z_1, v_1]) =$   
 $([\max\{x, x_1\}, \min\{y, y_1\}], [\max\{z, z_1\}, \min\{v, v_1\}])$

The top and bottom elements with respect to the various orders are as follows.

**Table 4.** Bounds relative to the three partial orderings.

Top value	Bottom value
$T_t = 1, 1, 0, 0$	$\perp_t = 0, 0, 1, 1$
$T_f = 0, 0, 0, 0$	$\perp_f = 1, 1, 1, 1$
$T_i = 1, 0, 1, 0$	$\perp_i = 0, 1, 0, 1$

$T_t$  corresponds to  $\{T, t\}$ ,  $\perp_t$  corresponds to  $\{F, f\}$ .

$T_f$  corresponds to  $\{F, t\}$ ,  $\perp_f$  corresponds to  $\{T, f\}$ .

$T_i$  corresponds to  $\{TF, tf\}$ ,  $\perp_i$  corresponds to  $\{\}$ .

## References

1. S. Older, Shiu-Kai Chin, IEEE Security & Privacy, **10(6)**, 74 (2012)
2. G. Bruns, M. Huth, 2008 21<sup>st</sup> IEEE Computer Security Foundations Symposium(CSF), 163 (2008)
3. Y. Shramko, H. Wansing, JPL, **34**, 121 (2005)
4. M.M. Kucherov, N.A. Bogulskaya, ACM International Conference on Cryptography, Security and Privacy (ICCCSP '17), 131 (2017)
5. L. V. S. Lakshmanan, N. Shiri, IEEE Trans. on Knowledge and Data Engineering, **13(4)**, 554 (2001)
6. M. Ginsberg, Computational Intelligence **4**, 256 (1988)
7. M. Fitting, Bilattices are nice things, in *Self-reference*, (CSLI Lecture Notes, CSLI, Stanford, 53, 2006)
8. G. Bruns, D.S. Dantas, and M. Huth. In *Proc. of the Fifth Workshop on Formal Methods in Security Engineering: From Specifications to Code*, 12 (2007)
9. N.J. Belnap, In *Contemporary Aspects of Philosophy* (Proceedings of the Oxford International Symposia, Oxford, GB, 30, 1975)
10. Y. Shramko, J. M. Dunn and T. Takenaka, JLC, **11(6)**, 761 (2001)
11. Y. Shramko, and H. Wansing, *Truth and Falsehood. An Inquiry into Generalized Logical Values* (Springer, Dordrecht, 2011)
12. P. Wild, L. Schröder, D. Pattinson, and B. König, *arXiv: 1802.00478v2*, 5Feb (2018)
13. G. Grätzer, *Lattice Theory: Foundation* (Springer Science & Business Media, Berlin/Heidelberg, Germany, 1998)
14. P. Hájek, *Metamathematics of fuzzy logic* (Trends in Logic, 4, Kluwer Academic Publishers, Dordrecht, 1998)
15. M. Zemankova and A. Kandel, Information Sciences, **37**, 107 (1985)
16. D. Dubois and H. Prade, In *Possibility Theory: An Approach to Computerized Processing of Uncertainty* (New York and London: Plenum Press, **6**, 217, 1989)
17. M. Kifer and A. Li, 2<sup>nd</sup> Intl. Conference on Database Theory, Bruges, Belgium: Springer Verlag, LNCS **326**, 102 (1988)
18. R. T. Ng and V. S. Subrahmanian, Information and Computation, **101(2)**, 150 (1992)
19. V. S. Lakshmanan, N. Leone, R. Ross, and V. S. Subrahmanian, ACM Trans. Database Systems, 419 (1997)
20. S. P. Odintsov, H. Wansing, Stud Logica, **103**, 91 (2015)
21. D. E. Denning, CACM, **19(5)**, 236 (1976)
22. J. Crampton, M. Huth, NordSec'10 Proceedings of the 15th Nordic conference on Information Security Technology for Applications, 155 (2010)
23. H. Wan, M. Kifer, LPNMR **2009**, 316 (2010)