

Article

# The Structure of the Block Code Generated by a *BL*-Algebra

Hashem Bordbar 

Centre for Information Technologies and Applied Mathematics, University of Nova Gorica,  
5000 Nova Gorica, Slovenia; hashem.bordbar@ung.si; Tel.: +386-070-614-786

**Abstract:** Inspired by the concept of *BL*-algebra as an important part of the ordered algebra, in this paper we investigate the binary block code generated by an arbitrary *BL*-algebra and study related properties. For this goal, we initiate the study of the *BL*-function on a nonempty set  $P$  based on *BL*-algebra  $L$ , and by using that,  $l$ -functions and  $l$ -subsets are introduced for the arbitrary element  $l$  of a *BL*-algebra. In addition, by the mean of the  $l$ -functions and  $l$ -subsets, an equivalence relation on the *BL*-algebra  $L$  is introduced, and using that, the structure of the code generated by an arbitrary *BL*-algebra is considered. Some related properties (such as the length and the linearity) of the generated code and examples are provided. Moreover, as the main result, we define a new order on the generated code  $C$  based on the *BL*-algebra  $L$ , and show that the structures of the *BL*-algebra with its order and the correspondence generated code with the defined order are the same.

**Keywords:** *BL*-function; *BL*-code; binary linear block codes; coding theory; *BL*-algebra

## 1. Introduction

*Hajek*, in 1998, introduced *BL*-algebra in order to provide a general framework for formalizing statements of a fuzzy nature [1]. *BL*-algebra is the algebraic structure arising from the continuous triangular norms, and it has certain logical axioms similarly to Boolean algebras or *MV*-algebras from classical logic or Lukasiewicz logic, respectively. In addition, every *MV*-algebra is a *BL*-algebra, whereas the converse is not always true. Thus, the class of *MV*-algebra is a subset of the class of *BL*-algebra, and this is the main reason that we selected *BL*-algebra to investigate the code generated by it. Moreover, *Hajek* showed that every *BL*-algebra with an involutory complement is *MV*-algebra.

In the twentieth century, there is a problem in engineering about the transmutation of information. Shannon [2] in 1948 and Hamming [3] in 1950 provided some frameworks to solve the problem. Their idea was developed and, as a consequence, the electronic information could be transmitted throughout the noisy channel and stored by minimum errors, and coding theory was born. Because the electronic information is a string of zeros and ones, it uses a finite field as the alphabet set. Thus, coding theory can use different areas of mathematics such as linear algebra, finite geometry, lattices, and combinatorics, especially when the alphabet set is generalized to different types of fields. Coding theory can be viewed not only as a part of computer science and engineering but also as a part of pure mathematics, as the mathematicians were interested in the fundamental aspects of this concept.

Application of coding theory in ordered algebraic structures was initiated by Jun and Song in 2011 [4]. They introduced the notion of BCK-valued functions and established binary block codes by using the notion of BCK-valued functions. After that, in 2014, Borumand et al. [5] and in 2015, Flaut [6] presented some relationships between *BCK*-algebras and the related binary block codes. They proved that every *BCK*-algebra determines a binary block code. Gilani [7] studied some properties of the codes generated by *BCK*-functions in an arbitrary *BCK*-algebras. Details about the fundamental relations in an arbitrary *BCK*-algebra and related generated code, namely *BCK*-code, was investigated by Bordbar in [8]. During the last few years, binary block codes generated by different



**Citation:** Bordbar, H. The Structure of the Block Code Generated by a *BL*-Algebra. *Mathematics* **2022**, *10*, 692. <https://doi.org/10.3390/math10050692>

Academic Editor: Patrick Solé

Received: 27 January 2022

Accepted: 21 February 2022

Published: 23 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

types of ordered algebraic structures were studied, for instance, codewords in a binary block code generated by a *UP*-valued function investigated by Chinram and Iampan in 2021 [9], and in 2015, Mostafa et al. [10] applied coding theory to *KU*-algebras and gave some connection between binary block codes and *KU*-algebras.

In this paper, we investigate the code generated by a *BL*-algebra. Our motivation is to study the properties of a code generated by one ordered algebraic structure. We begin with a discussion of the ordered algebraic structure known as *BL*-algebra as it is an extended algebraic structure and has some other ordered structures as its subsets, such as *MV*-algebra. Moreover, by using the order in a *BL*-algebra, we define a new order in a generated code and give the code an algebraic ordered structure. The defined order among the codewords can be useful in decoding and will be our future work. For our goal, in Section 3 we define a *BL*-function on an arbitrary nonempty set *P* based on *BL*-algebra *L* and by using the *BL*-function, *l*-functions of it and *l*-subsets of *P* for  $l \in L$  are investigated. In addition, properties of *BL*-function and its *l*-functions and *l*-subsets of *P* that we need for generated code are considered. In Section 4, a binary equivalence relation  $\approx$  defines on a *BL*-algebra *L*, and by using this relation and *BL*-function, the code *C* based on *L* is generated. Finally, we study an order among the codewords of *C* that gives the code *C* an ordered algebraic structure. In Theorem 3, we show that the *BL*-algebra *L* with its order and the code *C* based on *L* with respect to defined order have the same structures.

## 2. Preliminaries

A *BL*-algebra is a structure  $\mathcal{L} := (L, \wedge, \vee, \odot, \rightarrow, 0, 1)$  such that  $(L, \wedge, \vee, 0, 1)$  is a bounded lattice,  $(L, \odot, 1)$  is an abelian monoid, i.e.,  $\odot$  is commutative and associative, and the following conditions hold for all  $x, y, z \in L$ :

- (BL1)  $x \odot 1 = x$ ,
- (BL2)  $x \odot y \leq z$  if and only if  $x \leq y \rightarrow z$ ,
- (BL3)  $x \wedge y = x \odot (x \rightarrow y)$ ,
- (BL4)  $(x \rightarrow y) \vee (y \rightarrow x) = 1$ .

Every *BL*-algebra  $\mathcal{L}$  satisfies the following assertions

$$x = 1 \rightarrow x, x \rightarrow x = 1, x \rightarrow 1 = 1, \tag{1}$$

$$x \leq y \Leftrightarrow x \rightarrow y = 1, \tag{2}$$

$$x \leq y \rightarrow x, \tag{3}$$

$$x \rightarrow (y \rightarrow z) = y \rightarrow (x \rightarrow z), \tag{4}$$

$$x \leq y \Rightarrow z \rightarrow x \leq z \rightarrow y, y \rightarrow z \leq x \rightarrow z \tag{5}$$

for all  $x, y, z \in L$ .

For more information about *BL*-algebra, please refer to [1,11].

The alphabets used in coding theory are finite fields with *q* elements, *GF*(*q*). We say that a code is *q* – ary if its codewords are defined over the *q* – ary alphabet *GF*(*q*). The most commonly used alphabets are binary fields, *GF*(2*m*). This article focuses on codes with the familiar alphabet *GF*(2), which are known as *binary codes*.

Let *c* be a codeword. Then the *Hamming weight*  $w(c)$  of a codeword *c* is the number of nonzero components in the codeword. The *Hamming distance* between two codewords  $d(c_1, c_2)$  is the number of places in which the codewords *c*<sub>1</sub> and *c*<sub>2</sub> differ. In other words,  $d(c_1, c_2)$  is the Hamming weight of the vector *c*<sub>1</sub> – *c*<sub>2</sub>, representing the component-wise difference of the vectors *c*<sub>1</sub> and *c*<sub>2</sub>. The minimum (Hamming) distance of a code *C* is the minimum distance between any two codewords in the code *C*, that is,

$$d(C) = \min\{d(x, y) \mid x \neq y, x, y \in C\}.$$

The notation (*n*, *M*, *d*) is used to represent a code with code length *n*, a total of *M* codewords, and minimum distance *d*. One of the major goals of coding theory is to develop

codes that strike a balance between having small  $n$  (for fast transmission of messages), large  $M$  (to enable transmission of a wide variety of messages), and large  $d$  (to detect many errors).

### 3. BL-Functions on a Nonempty Set $P$

In this section, the notions of BL-functions on a nonempty set  $P$  based on a BL-algebra  $L$ ,  $l$ -functions, and  $l$ -subsets of  $P$  for an arbitrary element  $l \in L$ , will be introduced. Some of the properties connected with  $l$ -subsets of  $P$  and  $l$ -functions of a BL-function will be investigated. Throughout this section, unless stated otherwise,  $\mathcal{L} := (L, \wedge, \vee, \odot, \rightarrow, 0, 1)$  denotes a BL-algebra. In addition, in this paper, we use the set  $L$  for our definitions as a BL-algebra  $\mathcal{L} = (L, \wedge, \vee, \odot, \rightarrow, 0, 1)$ .

**Definition 1.** Let  $P$  be a nonempty set and  $L$  be a BL-algebra. A mapping  $\phi : P \rightarrow L$  is called a BL-function on  $P$  based on  $L$  and denoted by  $\phi_L$ . If there is no confusion of  $L$ , we use  $\phi$  instead of  $\phi_L$ . Moreover, for a BL-function  $\phi$  on  $P$  and  $l \in L$ , define  $\phi_l : P \rightarrow \{0, 1\}$  for each  $p \in P$  as follows:

$$\phi_l(p) = \begin{cases} 1, & \text{if and only if } \phi(p) \rightarrow l = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The function  $\phi_l$  is called a  $l$ -function of  $\phi$ .

**Definition 2.** Let  $P$  be a nonempty set and  $L$  be a BL-algebra. For a BL-function  $\phi : P \rightarrow L$  on  $P$  and each  $l \in L$ , the set  $P_l$  defined by

$$P_l := \{p \in P \mid \phi(p) \rightarrow l = 1\}, \tag{7}$$

is called a  $l$ -subset of  $P$ .

**Example 1.** Let  $L = \{0, a, b, 1\}$  be a set with the following Cayley tables:

$\odot$	0	a	b	1
0	0	0	0	0
a	0	a	a	a
b	0	a	b	b
1	0	a	b	1

$\rightarrow$	0	a	b	1
0	1	1	1	1
a	0	1	1	1
b	0	a	1	1
1	0	a	b	1

Then  $\mathcal{L} = (L, \wedge, \vee, \odot, \rightarrow, 0, 1)$  is a BL-algebra (see [12]), where  $x \wedge y = \min\{x, y\}$  and  $x \vee y = \max\{x, y\}$ .

(1) For a set  $P = \{p_1, p_2, p_3\}$ , the function  $\phi : P \rightarrow L$  defined by

$$\phi(p_1) = a, \phi(p_2) = b \text{ and } \phi(p_3) = 1,$$

is a BL-function on  $P$ , and the  $l$ -subsets of  $P$  for each  $l \in L$  are as follows:

$$P_0 = \emptyset, P_a = \{p_1\}, P_b = \{p_1, p_2\} \text{ and } P_1 = P.$$

In addition, for each  $l \in L$ , the  $l$ -functions of  $\phi$  are as shown in the following table:

$\phi_l$	$p_1$	$p_2$	$p_3$
$\phi_0$	0	0	0
$\phi_a$	1	0	0
$\phi_b$	1	1	0
$\phi_1$	1	1	1

(2) Let  $Q = \{q_1, q_2, q_3, q_4\}$  and define the function  $\psi : Q \rightarrow L$  by

$$\psi(q_1) = a, \psi(q_2) = b, \psi(q_3) = 1 \text{ and } \psi(q_4) = 0.$$

Then  $\psi$  is a BL-function on  $Q$ . For each  $l \in L$ , the  $l$ -subsets of  $Q$  are as follows:

$$Q_0 = \{q_4\}, Q_a = \{q_1, q_4\}, Q_b = \{q_1, q_2, q_4\} \text{ and } Q_1 = Q.$$

In addition, the  $l$ -functions of  $\psi$  for each  $l \in L$  are as shown in the following table:

$\psi_l$	$q_1$	$q_2$	$q_3$	$q_4$
$\psi_0$	0	0	0	1
$\psi_a$	1	0	0	1
$\psi_b$	1	1	0	1
$\psi_1$	1	1	1	1

The following proposition shows the relationship between BL-function  $\phi_L$  on  $P$  and its  $l$ -functions and  $l$ -subsets of  $P$  for  $l \in L$ .

**Proposition 1.** Let  $\phi : P \rightarrow L$  be a BL-function on a nonempty set  $P$  based on  $L$ , where  $L$  is a BL-algebra. Then the function  $\phi$  can be described by its  $l$ -functions and  $l$ -subsets of  $P$ , for  $l \in L$ , as the infimum of the following sets:

$$(\forall p \in P)(\phi(p) = \inf\{l \in L \mid p \in P_l\}), \tag{8}$$

in other words,

$$(\forall p \in P)(\phi(p) = \inf\{l \in L \mid \phi_l(p) = 1\}). \tag{9}$$

**Proof.** Let  $p \in P$  be an arbitrary element and  $\phi(p) = l$ . Then using (1),

$$\phi(p) \rightarrow l = l \rightarrow l = 1.$$

Thus  $\phi_l(p) = 1$ , which means that  $p \in P_l$ . Assume that  $p \in P_{l'}$ , for  $l' \in L$ . Then  $1 = \phi(p) \rightarrow l' = l \rightarrow l'$ . By using (2), we conclude that  $l \leq l'$ . Because  $l' \in \{l \in L \mid p \in P_l\}$ , it follows that

$$\phi(p) = \inf\{l \in L \mid p \in P_l\}.$$

The equality (9) is a direct conclusion of (6) and (7).  $\square$

**Corollary 1.** For a BL-algebra  $L$ , if  $\phi : P \rightarrow L$  is a BL-function on  $P$  based on  $L$ , then for  $p \in P$ ,

$$\phi(p) = \inf\{\phi_l(p) \leftrightarrow l \mid l \in L\},$$

where

$$\phi_l(p) \leftrightarrow l = \begin{cases} l & \text{if } p \in P_l, \\ 1 & \text{otherwise.} \end{cases}$$

**Proposition 2.** Let  $L$  be a BL-algebra and  $\phi : P \rightarrow L$  be a BL-function on a nonempty set  $P$  based on  $L$ . Then for elements  $l_1, l_2 \in L$  we have the following assertion,

$$l_1 \leq l_2 \Rightarrow P_{l_1} \subseteq P_{l_2} \tag{10}$$

**Proof.** Assume that  $l_1, l_2 \in L$  are arbitrary elements such that  $l_1 \leq l_2$ . Hence  $l_1 \rightarrow l_2 = 1$ . Moreover let  $x \in P_{l_1}$ . Then  $\phi(x) \rightarrow l_1 = 1$ , which means that  $\phi(x) \leq l_1$ . By using (5) we have

$$l_1 \rightarrow l_2 \leq \phi(x) \rightarrow l_2.$$

Thus using (1), we conclude that

$$(l_1 \rightarrow l_2) \rightarrow (\phi(x) \rightarrow l_2) = 1 \rightarrow (\phi(x) \rightarrow l_2) = \phi(x) \rightarrow l_2 = 1.$$

Therefore,  $x \in P_{l_2}$ , that is  $P_{l_1} \subseteq P_{l_2}$ .  $\square$

**Theorem 1.** Let  $\phi : P \rightarrow L$  be a BL-function on  $P$ . Then

- (i)  $(\forall p_1, p_2 \in P) (\phi(p_1) \neq \phi(p_2) \Leftrightarrow P_{\phi(p_1)} \neq P_{\phi(p_2)})$ .
- (ii)  $(\forall l \in L) (\forall p \in P) (p \in P_l \Leftrightarrow P_{\phi(p)} \subseteq P_l)$ .

**Proof.** For BL-function  $\phi : P \rightarrow L$ , let  $p_1, p_2 \in P$  be such that  $P_{\phi(p_1)} \neq P_{\phi(p_2)}$ . Moreover, suppose that  $\phi(p_1) = \phi(p_2)$ . If  $x \in P_{\phi(p_1)}$ , then

$$\phi(x) \rightarrow \phi(p_1) = \phi(x) \rightarrow \phi(p_2) = 1,$$

which means that  $x \in P_{\phi(p_2)}$ . Thus  $P_{\phi(p_1)} \subseteq P_{\phi(p_2)}$ . Similarly,  $P_{\phi(p_2)} \subseteq P_{\phi(p_1)}$ , and this is a contradiction. Therefore, for all  $p_1, p_2 \in P$ , if  $P_{\phi(p_1)} \neq P_{\phi(p_2)}$ , then  $\phi(p_1) \neq \phi(p_2)$ . This proves the  $(\Leftarrow)$ .

In order to prove  $(\Rightarrow)$ , suppose that  $p_1, p_2 \in P$  such that  $\phi(p_1) \neq \phi(p_2)$ . Then clearly  $\phi(p_1) \rightarrow \phi(p_2) \neq 1$  or  $\phi(p_2) \rightarrow \phi(p_1) \neq 1$ . Hence,

$$\begin{aligned} P_{\phi(p_1)} &= \{x \in P \mid \phi(x) \rightarrow \phi(p_1) = 1\} \\ &\neq \{x \in P \mid \phi(x) \rightarrow \phi(p_2) = 1\} \\ &= P_{\phi(p_2)}. \end{aligned}$$

Therefore,  $P_{\phi(p_1)} \neq P_{\phi(p_2)}$ .

(ii) Let  $l \in L$  and  $p \in P$  be such that  $p \in P_l$ . Then  $\phi(p) \rightarrow l = 1$ , and using Proposition 2,

$$P_{\phi(p)} \subseteq P_l.$$

Conversely, suppose that  $P_{\phi(p)} \subseteq P_l$  for  $l \in L$  and  $p \in P$ . Because  $\phi(p) \rightarrow \phi(p) = 1$ , we conclude that  $p \in P_{\phi(p)}$ . Therefore,  $p \in P_l$  and the proof is complete.  $\square$

Theorem 1 part (ii) shows that the converse of Proposition 2 is true. Thus, we have the following corollary.

**Corollary 2.** Let  $\phi : P \rightarrow L$  be a BL-function on a nonempty set  $P$  based on  $L$ , where  $L$  is a BL-algebra. Then

$$(\forall p_1, p_2 \in A) (\phi(p_1) \rightarrow \phi(p_2) = 1 \Leftrightarrow P_{\phi(p_1)} \subseteq P_{\phi(p_2)}). \tag{11}$$

**Proposition 3.** Let  $\phi : P \rightarrow L$  be a BL-function on a nonempty set  $P$  and  $M \subseteq L$ . Put

$$\alpha = \inf\{m \mid m \in M\}.$$

Then

$$P_\alpha = \cap\{P_m \mid m \in M\} \tag{12}$$

**Proof.** Note that there exists the infimum of  $M$  in  $L$  for any  $M \subseteq L$ . Thus, for the infimum element  $\alpha$  of  $M$  we have

$$\begin{aligned} x \in P_\alpha &\Leftrightarrow \phi(x) \rightarrow \alpha = 1 \\ &\Leftrightarrow (\forall m \in M) (\phi(x) \rightarrow m = 1) \\ &\Leftrightarrow (\forall m \in M) (x \in P_m) \\ &\Leftrightarrow x \in \cap\{P_m \mid m \in M\}. \end{aligned}$$

$\square$

For a BL-algebra  $L$  and a BL-function  $\phi : P \rightarrow L$  on a nonempty set  $P$ , define the sets  $P_L$  and  $\phi_L$  as follows:

$$P_L := \{P_l \mid l \in L\}, \phi_L := \{\phi_l \mid l \in L\}.$$

Then we have the following corollary.

**Corollary 3.** *If  $\phi : P \rightarrow L$  is a BL-function on a nonempty set  $P$ , then*

- (i)  $P_{\inf\{l \mid l \in L\}} = \cap\{P_l \mid l \in L\}$ ,
- (ii) for  $l_1, l_2 \in L$ , we have  $P_{l_1} \cap P_{l_2} \in P_L$ .

**Proposition 4.** *Let  $\phi : P \rightarrow L$  be a BL-function on a nonempty set  $P$ . Then  $P$  is represented by the union of  $P_l$  for  $l \in L$ , that is,*

$$P = \cup\{P_l \mid l \in L\}. \tag{13}$$

**Proof.** Obviously,  $\cup\{P_l \mid l \in L\} \subseteq P$ . Let  $p \in P$  and  $l \in L$  be such that  $\phi(p) = l$ . Then by using the definition of  $l$ -subset of  $P$ , we have  $p \in P_l$ . Thus,

$$p \in \cup\{P_l \mid l \in L\},$$

which means that  $P \subseteq \cup\{P_l \mid l \in L\}$ . Therefore,  $P = \cup\{P_l \mid l \in L\}$ .  $\square$

**Proposition 5.** *Let  $\phi : P \rightarrow L$  be a BL-function on a nonempty set  $P$  and  $p \in P$ . Then*

$$\cap\{P_l \mid p \in P_l\} \in P_L. \tag{14}$$

**Proof.** Remember that by using (6) and (7), we conclude that for any  $p \in P$ ,

$$p \in P_l \Leftrightarrow \phi_l(p) = 1.$$

It follows from Proposition 3 that

$$\cap\{P_l \mid p \in P_l\} = \cap\{P_l \mid \phi_l(p) = 1\} = P_{\inf\{l \mid p \in P_l\}} \in P_L.$$

$\square$

Let  $\phi : P \rightarrow L$  be a BL-function on a nonempty set  $P$  and  $\approx$  be a binary relation on  $L$  defined by

$$(\forall l_1, l_2 \in L)(l_1 \approx l_2 \Leftrightarrow P_{l_1} = P_{l_2}). \tag{15}$$

The binary relation  $\approx$  is an equivalence relation on  $L$ . Moreover, for an arbitrary element  $l \in L$ , define the sets  $\phi(P)$  and  $\{l\}_\leq$  as follows:

$$\phi(P) := \{l \in L \mid \phi(p) = l \text{ for some } p \in P\}$$

$$\{l\}_\leq := \{x \in L \mid x \leq l\} = \{x \in L \mid x \rightarrow l = 1\}.$$

The relationships between an equivalence relation  $\approx$  and the sets  $\phi(P)$  and  $\{l\}_\leq$  are described in the following theorem.

**Theorem 2.** *For a BL-function  $\phi : P \rightarrow L$  on a nonempty set  $P$  and the elements  $l_1, l_2 \in L$ , we have the following assertion:*

$$l_1 \approx l_2 \Leftrightarrow \phi(P) \cap \{l_1\}_\leq = \phi(P) \cap \{l_2\}_\leq. \tag{16}$$

**Proof.** Suppose that  $l_1, l_2 \in L$ . Then

$$\begin{aligned} l_1 \approx l_2 &\Leftrightarrow P_{l_1} = P_{l_2} \\ &\Leftrightarrow (\forall p \in P)(\phi(p) \rightarrow l_1 = 1 \Leftrightarrow \phi(p) \rightarrow l_2 = 1) \\ &\Leftrightarrow \{p \in P \mid \phi(p) \in \{l_1\}_\leq\} = \{p \in P \mid \phi(p) \in \{l_2\}_\leq\} \\ &\Leftrightarrow \phi(P) \cap \{l_1\}_\leq = \phi(P) \cap \{l_2\}_\leq. \end{aligned}$$

$\square$

#### 4. Code Generated by a BL-Algebra

The relation  $\approx$  on  $L$  that is defined in (15) is an equivalence relation on  $L$ . Thus, it provides the partition of  $L$ . For any  $l \in L$ , let  $[l]$  denotes an equivalence class containing  $l$ , which means that

$$[l] := \{k \in L \mid l \approx k\}.$$

In what follows, a binary block code of length  $n$  will be made from an arbitrary finite BL-algebra. In this method  $n$  is a natural number; this helps us to generate a binary block code of the desired length  $n$ .

For  $n \in \mathbb{N}$ , let  $P = \{p_1, p_2, \dots, p_n\}$  and  $L$  be a finite BL-algebra. Every BL-function  $\phi : P \rightarrow L$  on  $P$  determines a binary block code  $C$  of length  $n$  in the following way:

Let  $l \in L$ . Then for  $[l]$  the correspondence codeword is  $c_l = c_1c_2 \cdots c_n$  such that for  $1 \leq i \leq n$

$$c_i = \phi_l(p_i) \tag{17}$$

where  $p_i \in P$ . We called  $C$  a BL-code based on  $L$  and denoted by  $C_L$ . If there is no confusion of  $L$ , we use  $C$  instead of  $C_L$ .

During our study of block code generated by an arbitrary BL-algebra, three parameters are important. The first parameter is the code length  $n$ . In the BL-code based on  $L$ , we can make a code of the desired length  $n$ . This can be helpful as we can choose the length in different situations. The second parameter that we consider is the total number of codewords. In this kind of code, the total number of codewords is equal to the total number of distinct equivalence classes of  $\approx$  relation. The third parameter is the distance between pairs of codewords in a code. In the following examples, these notations will be explained much more.

**Example 2.** Let  $L = \{0, a, b, 1\}$  be a set with Cayley tables as follows:

$\odot$	0	a	b	1	$\rightarrow$	0	a	b	1
0	0	0	0	0	0	1	1	1	1
a	0	0	a	a	a	a	1	1	1
b	0	a	b	b	b	b	0	a	1
1	0	a	b	1	1	1	0	a	b

Then  $\mathcal{L} := (L, \wedge, \vee, \odot, \rightarrow, 0, 1)$  is a BL-algebra. For a set  $P = \{p_1, p_2, p_3, p_4\}$ , let  $\phi : P \rightarrow L$  be a BL-function on  $P$  given by

$$\phi(p_1) = a, \phi(p_2) = 0, \phi(p_3) = 1 \text{ and } \phi(p_4) = b$$

Then the  $l$ -subsets of  $P$  are

$$P_0 = \{p_2\}, P_a = \{p_1, p_2\}, P_b = \{p_1, p_2, p_4\} \text{ and } P_1 = P.$$

In addition, the  $l$ -functions of  $\phi$  are

$\phi_p$	p <sub>1</sub>	p <sub>2</sub>	p <sub>3</sub>	p <sub>4</sub>
$\phi_0$	0	1	0	0
$\phi_a$	1	1	0	0
$\phi_b$	1	1	0	1
$\phi_1$	1	1	1	1

Clearly, we have four different equivalence classes, which are  $[0], [a], [b]$ , and  $[1]$ . Thus the total number of codewords is 4 ( $M = 4$ ). By using (17), we conclude that

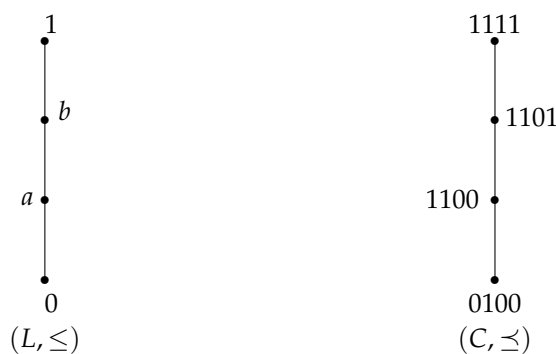
$$c_0 = 0100, c_a = 1100, c_b = 1101, c_1 = 1111.$$

Thus the binary block code  $C$  of length  $n = 4$  and 4 codewords is  $C = \{0100, 1100, 1101, 1111\}$ . Besides, the minimum distance of  $C$  is 1 ( $d(C) = 1$ ). It is clear that the code  $C$  is not a linear code because  $0000 \notin C$ .

Let  $c_l = l_1l_2 \cdots l_n$  and  $c_k = k_1k_2 \cdots k_n$  be two codewords belonging to a binary block-code  $C$  of length  $n$ . Define an order relationship  $\preceq$  on the set of codewords belonging to a binary block-code  $C$  of length  $n$  as follows:

$$c_l \preceq c_k \Leftrightarrow l_i \leq k_i \text{ for } i = 1, 2, \dots, n. \tag{18}$$

By using (2) for the BL-algebra  $L$  and (18) for the BL-code  $C$  based on  $L$ , we conclude that the graphs of  $L$  concerning the order  $\leq$  and the code  $C$  with respect to the order  $\preceq$  have the same structures. For instance, in Example 2, we have



Therefore, we can have the following theorem.

**Theorem 3.** Let  $\mathcal{L} := (L, \wedge, \vee, \odot, \rightarrow, 0, 1)$  be a finite BL-algebra and  $|L| = n$ , where  $n \in \mathbb{N}$ . Then  $L$  determines a block-code  $C$  of length  $n$  (namely BL-code) such that the graph of  $L$  with respect to its order  $\leq$  and the graph of BL-code  $C$  with respect to the order  $\preceq$  have the same structure.

**Proof.** Let  $\mathcal{L} := (L, \wedge, \vee, \odot, \rightarrow, 0, 1)$  be a finite BL-algebra and  $L = \{l_1, l_2, \dots, l_n\}$ . Moreover, suppose that  $P = L$ . Then  $P$  is a nonempty set and  $\phi : L \rightarrow L$  defined by  $\phi(l_i) = l_i$ , for  $1 \leq i \leq n$  is a BL-function on  $L$  based on  $L$ . Suppose that  $\frac{L}{\approx}$  be a set of all equivalence classes of the elements of  $L$  regarding the equivalence relation  $\approx$  defined in (15). That is,

$$\frac{L}{\approx} = \{[l] \mid l \in L\}.$$

Define the mapping  $\psi : \frac{L}{\approx} \rightarrow C$  by

$$\psi([l_i]) = c_{l_i}, \tag{19}$$

whereby using (17), we have

$$c_{l_i} = \phi_{l_i}(l_1)\phi_{l_i}(l_2) \cdots \phi_{l_i}(l_n),$$

for  $1 \leq i \leq n$ .



Moreover, let  $[l_i] = [l_j]$ , for  $i, j \in \{1, 2, \dots, n\}$ , and  $i \neq j$ . Then using (7) and (15), we conclude that

$$\begin{aligned}
 [l_i] = [l_j] &\Leftrightarrow L_{l_i} = L_{l_j} \\
 &\Leftrightarrow \{l \in L \mid \phi(l) \rightarrow l_i = 1\} = \{l \in L \mid \phi(l) \rightarrow l_j = 1\} \\
 &\Leftrightarrow \{l \in L \mid \phi_{l_i}(l) = 1\} = \{l \in L \mid \phi_{l_j}(l) = 1\} \\
 &\Leftrightarrow (\forall l \in L) \left( \phi_{l_i}(l) = 1 \Leftrightarrow \phi_{l_j}(l) = 1 \right) \\
 &\Leftrightarrow (\forall l \in L) \left( \phi_{l_i}(l) = 0 \Leftrightarrow \phi_{l_j}(l) = 0 \right) \\
 &\Leftrightarrow c_{l_i} = c_{l_j}.
 \end{aligned}$$

This means that the function  $\psi$  in (19) is well defined and the inverse implications show that the function  $\psi$  is one-to-one.

Now suppose that  $l_i, l_j \in L$  are such that  $l_i \leq l_j$ , for  $1 \leq i, j \leq n$ . Then Proposition 2 shows that  $L_{l_i} \subseteq L_{l_j}$ . If  $l \in L$  and  $l \in L_{l_i}$ , then  $\phi_{l_i}(l) = 1$ . Because  $L_{l_i} \subseteq L_{l_j}$ ,  $l \in L_{l_j}$  and  $\phi_{l_j}(l) = 1$ , therefore  $\phi_{l_i}(l) \leq \phi_{l_j}(l)$ . Thus, in this case  $c_{l_i} \leq c_{l_j}$ .

If  $l \in L$  and  $l \notin L_{l_i}$ , then  $\phi_{l_i}(l) = 0$  and we have two opportunities. The first one is  $l \in L_{l_j}$ , which means that  $\phi_{l_j}(l) = 1$ , and the second one is  $l \notin L_{l_j}$ , which means that  $\phi_{l_j}(l) = 0$ . In both case, we have  $\phi_{l_i}(l) \leq \phi_{l_j}(l)$ . Hence,  $c_{l_i} \leq c_{l_j}$ . Therefore, if  $l_i \leq l_j$ , then  $\psi(l_i) \leq \psi(l_j)$ , that is,  $\psi$  preserves the order. Therefore, the figures of  $(L, \leq)$  and  $(C, \leq)$  have the same structures.  $\square$

**Example 3.** Let  $L = \{0, a, b, c, d, 1\}$  be a set with the following Cayley tables.

$\odot$	1	a	b	c	d	0	$\rightarrow$	1	a	b	c	d	0
1	1	a	b	c	d	0	1	1	a	b	c	d	0
a	a	b	b	d	0	0	a	1	1	a	c	c	d
b	b	b	b	0	0	0	b	1	1	1	c	c	c
c	c	d	0	c	d	0	c	1	a	b	1	a	b
d	d	0	0	d	0	0	d	1	1	a	1	1	a
0	0	0	0	0	0	0	0	1	1	1	1	1	1

Then  $\mathcal{L} = (L, \wedge, \vee, \odot, \rightarrow, 0, 1)$  is a BL-algebra. For a set  $P = \{p_i \mid i = 1, 2, \dots, 6\}$ , let  $\phi : P \rightarrow L$  be a BL-function on a set  $P$  given by

$$\phi(p_1) = 0, \phi(p_2) = a, \phi(p_3) = b, \phi(p_4) = c, \phi(p_5) = d, \text{ and } \phi(p_6) = 1.$$

Then

$\phi_p$	$p_1$	$p_2$	$p_3$	$p_4$	$p_5$	$p_6$
$\phi_0$	1	0	0	0	0	0
$\phi_a$	1	1	1	0	1	0
$\phi_b$	1	0	1	0	0	0
$\phi_c$	1	0	0	1	1	0
$\phi_d$	1	0	0	0	1	0
$\phi_1$	1	1	1	1	1	1

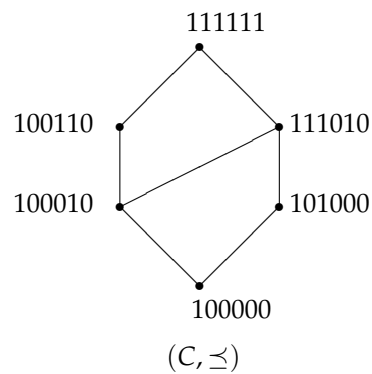
Thus, using the (17) we have

$$c_0 = 100000, c_a = 111010, c_b = 101000, c_c = 100110, c_d = 100010, c_1 = 111111.$$

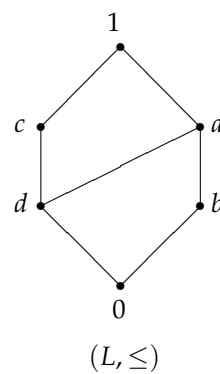
Finally, the generated binary block code  $C$  based on  $L$  is:

$$C = \{100000, 111010, 101000, 100110, 100010, 111111\}.$$

Moreover, the graph of  $C$  using the order (18) is as follows:



which is the same with the graph of  $L$  and the order (2), that is,



**5. Conclusions**

In this paper, we have studied the code generated by a  $BL$ -algebra as one of the important classes of ordered algebra. For this goal, the notion of  $BL$ -function on a nonempty set  $P$  based on  $BL$ -algebra  $L$  was introduced, and for  $l \in L$ ,  $l$ -fuctions of a  $BL$ -fuction and  $l$ -subsets of  $P$  were studied. After investigating some results concerning the  $BL$ -functions, our study has focused on a binary equivalence relation  $\approx$  on  $L$ , and using this relation we define the code  $C$  based on  $L$ . Finally, we have defined an order between the codewords of  $C$ , which gives the code  $C$  the ordered structure. Moreover, the graph of  $C$  with its order and the graph of  $L$  have the same structures.

The results related to  $BL$ -code  $C$  show that, in general, this code is not linear. For our future work, we will concentrate on some conditions that make this binary block code a linear code. Moreover, using the notations and ideas of this article, the order that we defined between the codewords of the code  $C$  based on  $BL$ -algebra  $L$  can help us to find a new algorithm for decoding the ciphertext. In our future research, we focus on this part of the decoding algorithm.

**Author Contributions:** Conceptualization, H.B.; methodology, H.B.; investigation, H.B.; writing—original draft preparation, H.B.; writing—review and editing, H.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** No funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

1. Hájek, P. *Metamathematics of Fuzzy Logic*; Kluwer Academic Publishers: Dordrecht, The Netherlands, 1998.
2. Shannon, C.E. A Mathematical Theory of Communications. *Bell Syst. Tech. J.* **1948**, *27*, 379–423 [[CrossRef](#)]
3. Hamming, R.W. Error detecting and error-correcting codes. *Bell Syst. Tech. J.* **1950**, *29*, 147–160. [[CrossRef](#)]
4. Jun, Y.B.; Song, S.Z. Codes based on BCK-algebras. *Inf. Sci.* **2011**, *181*, 5102–5109. [[CrossRef](#)]
5. Saeid, A.B.; Fatemidokht, H.; Flaut, C.; Rafsanjani, M.K. On codes based on BCK-algebras. *J. Intell. Fuzzy Syst.* **2015**, *29*, 2133–2137. [[CrossRef](#)]
6. Flaut, C. BCK-algebras arising from block codes. *J. Intell. Fuzzy Syst.* **2015**, *28*, 1829–1833. [[CrossRef](#)]
7. Gilani, A. Codes generated by BCK-functions in BCK-algebras. In Proceedings of the 4th International Conference on Combinatorics, Cryptography, Computer Science and Computing, Tehran City, Iran, 20–21 November 2019.
8. Bordbar, H. BCK codes. *Adv. Comp. Int.* **2022**, *2*, 4. [[CrossRef](#)]
9. Chinram, R.; Iampan, A. Codewords generated by UP-valued functions. *AIMS Math.* **2021**, *6*, 4771–4785. [[CrossRef](#)]
10. Mostafa, M.; Youssef, B.; Jad, H.A. Coding theory applied to KU-algebras. *J. New Theory* **2015**, *6*, 43–53.
11. Busneag, D.; Piciu, D. On the lattice of deductive systems of a BL-algebra. *Cent. Eur. Sci. J.* **2003**, *2*, 221–237.
12. Hedayati, H.; Jun, Y.B. Chain conditions on fuzzy positive implicative filters of BL-algebras. *Hacet. J. Math. Stat.* **2011**, *40*, 819–828.