Thai Journal of **Math**ematics
Volume 19 Number 2 (2021)
Pages 593–605

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Binary Code Properties of Perfect Matching in Hexagonal Graph

Asekha Khantavchai[1] **and Thiradet Jiarasuksakun**[2,*]

[1] *Department of Informatics Mathematics, Suan Sunandha Rajabhat University, 1 U-Thong nok Road, Dusit, Bangkok, 10300, Thailand*
e-mail : asekha.kh@ssru.ac.th

[2] *Department of Mathematics, King Mongkut's University of Technology Thonburi, 126 Pracha Uthit Rd., Bang Mod, Thung Khru, Bangkok, 10140, Thailand*
e-mail : thiradet.jia@kmutt.ac.th

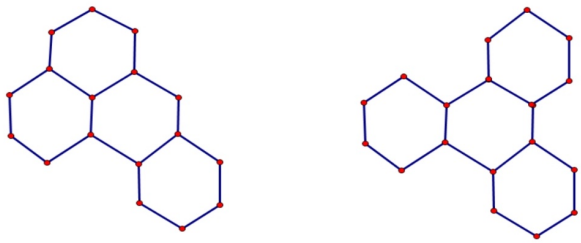
Abstract This paper presents the binary code of perfect matching of hexagonal graph and its properties. According to the results, it can be concluded that the hexagonal graph can be substituted by binary code using Klavzar's algorithm. The properties of the obtained binary code could be used in encoding and decoding. In summary, the perfect matchings of the hexagonal graph can be applied in receiving and delivering data after being replaced by the binary code.

MSC: 94A24; 94B05; 94C15
Keywords: perfect matching; binary code; Klavzar's algorithm; hexagonal graph

Submission date: 13.06.2018 / Acceptance date: 10.10.2018

1. Introduction

This section, we will study about catacondensed hexagonal graphs [1], it is *hexagonal graph* G such that each two adjacent hexagons have exactly one common edge and no vertex is element of three hexagons.



(a) (b)

*Corresponding author.

Figure 1. (a) is not a catacondensed hexagonal graph, while (b) is a catacondensed hexagonal graph.

Another important property of hexagonal graph G is a *resonance graph* of G denoted by $R(G)$. Its vertices are all perfect matching of G and each two vertices are *adjacent* if only three double bond positions in the matchings are different [2].

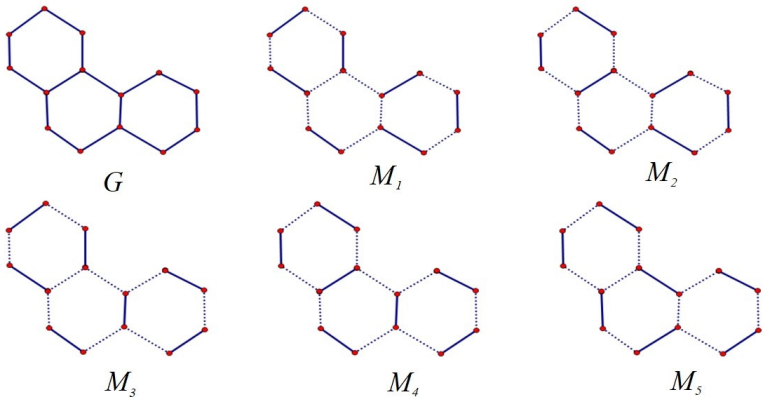


Figure 2. All perfect matchings of G.

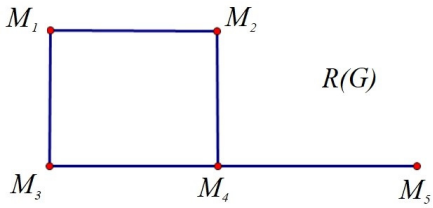


Figure 3. $R(G)$ is a resonance graph of G.

In 2000, Klavzar and Zigert [3] show that every perfect matching of catacondensed hexagonal graphs correspondence to binary code. The method of finding binary code of perfect matching of catacondensed hexagonal graphs. In substitution procedure, types of generating of double bonds in hexagonal graph of each hexagon need to be taken into consideration as the following.

(1) A substitution for one hexagon refers to the following picture.

Figure 4. A substitution for one hexagon.

It can be consistently seen that a hexagonal graph with 1 hexagon exactly has 2 perfect matchings.

Definition 1.1. 1. $\mathcal{B}^n = \{[u]_n = [u_1u_2...u_n]_n \,|u_i \in \{0,1\} \ \forall i = 1,2,3,...n\}$
2. $S(G)|_n$ is the set of binary code substitution of the perfect matchings of hexagonal graph with n hexagons.
3. $\mathcal{B}_k^n$ is the set of codeword in $\mathcal{B}^n$ with each weighs k and all 1- bits are adjacent.

Therefore, $S(G)|_1$ is defined in terms of binary code substitution of the perfect matchings of hexagonal graph with 1 hexagon which accounts for $S(G)|_1 = \{[0]_1, [1]_1\}$.

(2) Hexagonal graph with more than 1 hexagon can be substituted as demonstrated in the following method. Let G be catacondensed hexagonal graphs with n hexagons. We want to find the set of codewords $S(G)|_n$. Given H is n^{th} hexagon of G and e is a common edge between H and $(n-1)^{th}$ hexagon of G and $G' = (G - H) \cup \{e\}$.

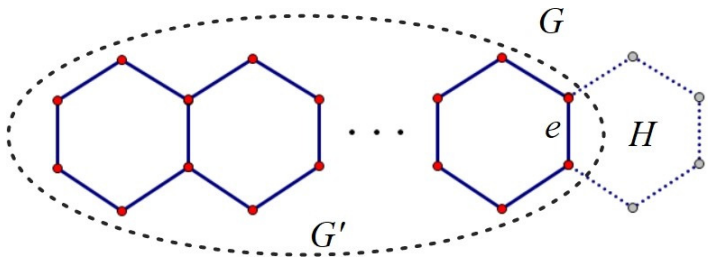


Figure 5. Definition of G' of G.

Thus there are only three possible patterns of double bounds of H.

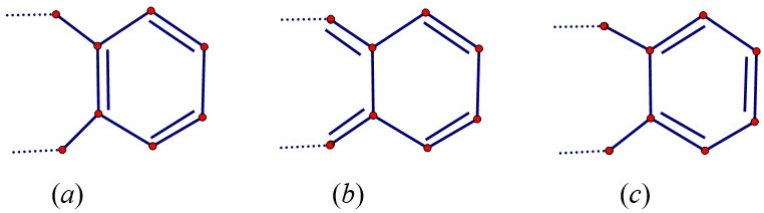
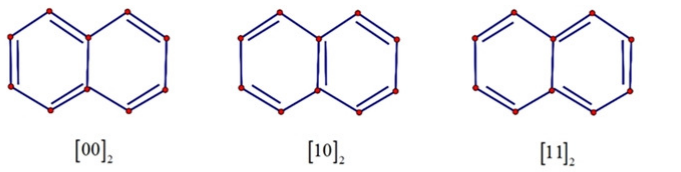
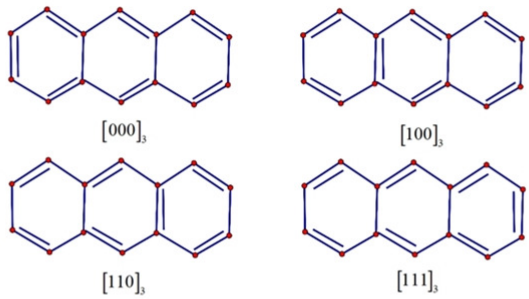


Figure 6. Three possible patterns of double bounds of H.

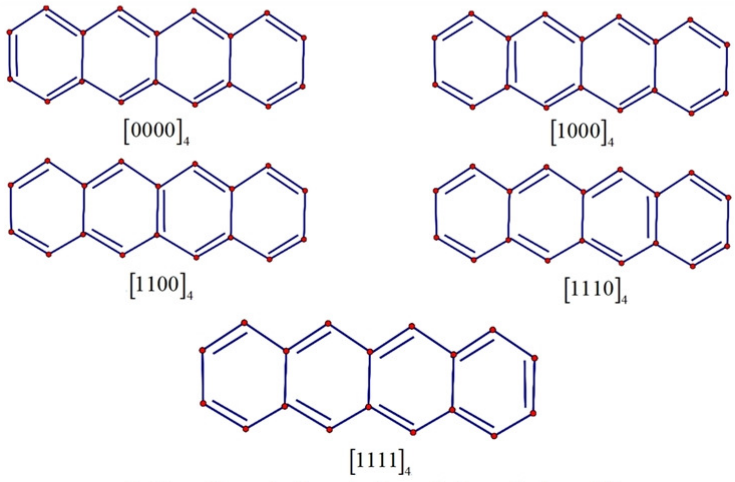
If H is an (a) or (b) form, we will take a bit 0 to $[x]_{n-1} \in S\left(G'\right)|_{n-1}$ for all $[x]_{n-1}$. Hence $[x0]_n \in S(G)|_n$. In another way, if H is an (c) form, we will take a bit 1 to $[x]_{n-1} \in S\left(G'\right)|_{n-1}$ for all $[x]_{n-1}$. Hence $[x1]_n \in S(G)|_n$.



$$S(G)|_2 = \{[00]_2, [10]_2, [11]_2\}.$$



$$S(G)|_3 = \{[000]_3, [100]_3, [110]_3, [111]_3\}.$$



$$S(G)|_4 = \{[0000]_4, [1000]_4, [1100]_4, [1110]_4, [1111]_4\}.$$

Figure 7. Example of binary code of some perfect matching.

It can be seen that it is difficult to substitute the binary for the perfect matchings according to the above-mentioned procedure, and it is difficult in a general case as well. Later that same year, Klavzar [3] developed algorithm in binary code substitution for perfect matchings of linear hexagonal graph. This provides an understanding and an easy converting the binary code in general cases of linear hexagonal graphs. This causes the researcher to use Klavzars algorithm in studying binary code properties, including application in receiving and delivering the information.

Linear hexagonal graph refers to hexagons lining up but not superimposing which can produce straight lines and kinks. The binary code converting can be proceeded hereinafter.

Definition 1.2. G is a linear hexagonal graph with hexagon named $H_1, H_2, H_3, ..., H_n$ respectively. H_i is i^{th} hexagon of G and e_{i-1} is a common edge between H_{i-1} and H_i. Then H_i is **kink** if e_{i-1} and e_i are not parallel, up to isomorphism.

Let $S(G)|_i$ be specified as binary code of linear hexagonal graph with hexagon named $H_1, H_2, H_3, ..., H_i$ respectively. $S(G)|_i$ is calculated from $S(G)|_{i-1}$ which depends on types of generating the double bonds of H_{i-1} and H_{i-2} which includes 6 possible types as follows.

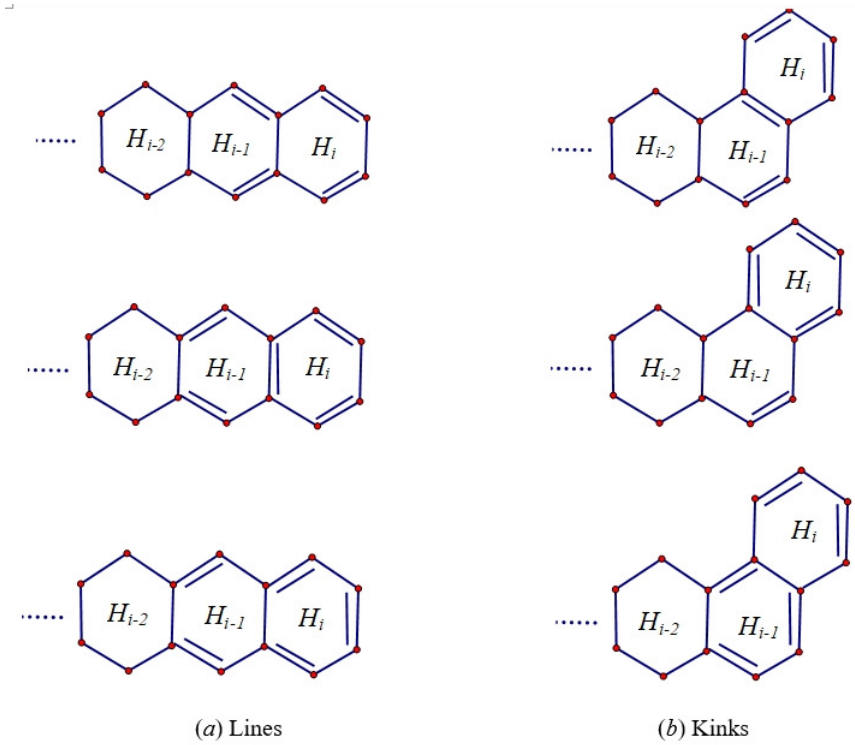


Figure 8. Possible types of H_{i-2}, H_{i-1} and H_i.

According to the above picture, it can be concluded that the principles of generating binary code are [3]:

(1) In case of straight lines, $S(G)|_i$ is calculated by adding the final bit to a member of $S(G)|_{i-1}$. That is $S(G)|_i$ composed of $[u_1u_2u_3...u_{i-1}0]_i$ when $[u_1u_2u_3...u_{i-1}]_{i-1}$ is a binary code in $S(G)|_{i-1}$, and of $[u_1u_2u_3...u_{i-1}1]_i$ when binary code $[u_1u_2u_3...u_{i-1}]_{i-1}$ ends with 1.

(2) In case of kink linear hexagonal graph, $S(G)|_i$ consisted of $[u_1u_2u_3...u_{i-1}0]_i$ when $[u_1u_2u_3...u_{i-1}]_{i-1}$ is a binary code in $S(G)|_{i-1}$, and of $[u_1u_2u_3...u_{i-1}1]_i$ when binary code $[u_1u_2u_3...u_{i-1}]_{i-1}$ ends with 0.

Example 1.3. We will consider G, which is linear hexagonal graph with connected kinks. By Klavzar's algorithm, we have

$$S(G)|_2 = \{[00]_2, [10]_2, [11]_2\}$$
$$S(G)|_3 = \{[000]_3, [100]_3, [110]_3, [001]_3, [101]_3\}$$
$$S(G)|_4 = \{[0000]_4, [1000]_4, [1100]_4, [0010]_4, [1010]_4, [0001]_4, [1001]_4, [1101]_4\}$$
$$S(G)|_5 = \left\{\begin{array}{l}[00000]_5, [10000]_5, [11000]_5, [00100]_5, [10100]_5, [00010]_5, [10010]_5, \\ [11010]_5, [00001]_5, [10001]_5, [11001]_5, [00101]_5, [10101]_5\end{array}\right\}$$
$$S(G)|_6 = \left\{\begin{array}{l}[000000]_6, [100000]_6, [110000]_6, [001000]_6, [101000]_6, [000100]_6, \\ [100100]_6, [110100]_6, [000010]_6, [100010]_6, [110010]_6, [001010]_6, \\ [101010]_6, [000001]_6, [100001]_6, [110001]_6, [001001]_6, [101001]_6, \\ [000101]_6, [100101]_6, [110101]_6\end{array}\right\}$$

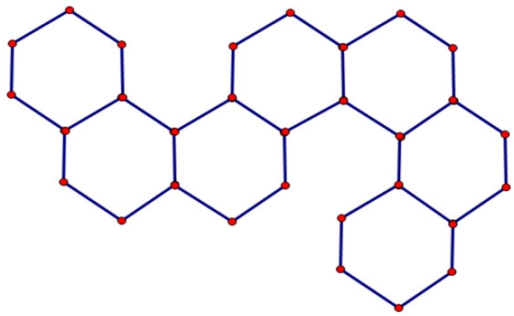


Figure 9. Example of a linear hexagonal graph with connected kinks.

This solution makes an easier method to find a resonance graph $R(G)$ of catacondensed hexagonal graphs G. To find edges of $R(G)$, we suppose that G has n hexagons then we will consider perfect matching $u, v \in V(R(G))$ such that correspondence to codeword $u, v \in \mathcal{B}^n$. uv is an edge in $R(G)$ if and only if $d(u, v) = 1$. However, here we will study properties of binary code of perfect matching of catacondensed hexagonal graphs G for receiving and delivering information. We hope that future study will make new method to receiving and delivering information by hydrocarbon melecules.

2. The Properties of Binary Code of Linear Hexagonal Graph

According to algorithm above, binary code of linear hexagonal graph can be produced more easily and its the properties can be studied in general. With the algorithm, $S(G)|_i$ could be calculated because the members of $S(G)|_{i-1}$ are known. Consequently, a well-known proving technique, mathematical induction is employed to help in the study of binary code of linear hexagonal graph. However, in order to prevent any confusion, we will define some notations in this section as follows.

1. If $[x]_{n-1} \in \mathcal{B}^{n-1}$ and $y \in \{0,1\}$, then $[xy]_n \in \mathcal{B}^n$ are binary codes of word length n. It is added bit y to binary code $[x]_{n-1}$.

2. If $[u]_n \in \mathcal{B}^n$ and $k \in \{0,1\}$, then $k \cdot [u]_n \in \mathcal{B}^n$. It is a scalar multiplication.

Theorem 2.1. *Let $S(G)|_n$ be the set of binary code of straight line (without kink) of linear hexagonal graph. Then $|S(G)|_n| = n+1$ minimum weight $w(S(G)|_n) = 0$, min $d(S(G)|_n) = 1$ and max $d(S(G)|_n) = n$.*

Proof. Let $S(G)|_n$ be the set of binary code of straight line (without kink) of linear hexagonal graph. By mathematical induction, it is obvious that $|S(G)|_n| = n+1$. It can be seen as follows: $[0]_n = [000...0]_n \in S(G)|_n$, $[100...0]_n \in S(G)|_n$ and $[111...1]_n \in S(G)|_n \ \forall n \in N$. Thus $w(S(G)|_n) = w([0]_n) = 0$, $min\ d(S(G)|_n) = d([000...0]_n, [100...0]_n) = 1$ and $max\ d(S(G)|_n) = d([000...0]_n, [111...1]_n) = n$. ∎

Theorem 2.2. *Let $S(G)|_n$ be the set of binary code of straight line (without kink) of linear hexagonal graph. $S(G)|_n - \{[0]_n\}$ is a basis of $\mathcal{B}^n$.*

Proof. Let $\mathcal{B}^n = \{[u]_n = [u_1u_2...u_n]_n \,|u_i \in \{0,1\} \ \forall i = 1,2,3,...n\}$ and by Klavzar's algorithm we can write

$$S(G)|_n = \Big\{ [0]_n, [x_10]_n, [x_20]_n, [x_30]_n, ..., [x_{n-1}0]_n, [111...1]_n \,| \\ [x_i]_{n-1} \in S(G)|_{n-1} - \{[0]_{n-1}\} \Big\}$$

By mathematical induction, we assume that $S(G)|_k - \{[0]_k\}$ is a basis of $\mathcal{B}^n \ \forall k < n$.

(1) Suppose that $a_1 \cdot [x_10]_n + a_2 \cdot [x_20]_n + ... + a_{n-1} \cdot [x_{n-1}0]_n + a_n \cdot [111...1]_n = [0]_n$, where $a_i \in \mathcal{B}$. Then $a_1 \cdot [x_10]_n + a_2 \cdot [x_20]_n + ... + a_{n-1} \cdot [x_{n-1}0]_n + [a_na_na_n...a_n]_n = [0]_n$. Therefore, the last bit $0 + a_n = 0$, then $a_n = 0$. We have $a_1 \cdot [x_10]_n + a_2 \cdot [x_20]_n + ... + a_{n-1} \cdot [x_{n-1}0]_n + [000...0]_n = [0]_n$ and $a_1 \cdot [x_1]_{n-1} + a_2 \cdot [x_2]_{n-1} + ... + a_{n-1} \cdot [x_{n-1}]_{n-1} = [0]_{n-1}$. Since $S(G)|_{n-1} - \{[0]_{n-1}\}$ is linearly independent, then $a_i = 0 \ \forall i$. And $a_n = 0$, thus $S(G)|_n - \{[0]_n\}$ is linearly independent.

(2) Suppose that $[u]_n = [u_1u_2...u_n]_n \in \mathcal{B}^n$. By mathematical induction, we have $S(G)|_{n-1} - \{[0]_{n-1}\}$ span $\mathcal{B}^{n-1}$ and there are $a_1, a_2, ..., a_{n-1} \in \mathcal{B}$ such that $a_1 \cdot [x_1]_{n-1} + a_2 \cdot [x_2]_{n-1} + ... + a_{n-1} \cdot [x_{n-1} = 111...1]_{n-1} = [u_1u_2...u_{n-1}]_{n-1}$....(*) We know that $u_i + u_i = 0 \ \forall u_i \in \mathcal{B}$, then $u_n \cdot [111...10]_n + u_n \cdot [111...11]_n = [000...0u_n]_n$...(**)

By (*) and (**), there are $a_1, a_2, ..., a_{n-2}, a_{n-1} + u_n, u_n \in \mathcal{B}$ such that
$a_1 \cdot [x_10]_n + a_2 \cdot [x_20]_n + ... + a_{n-2} \cdot [x_{n-2}0]_n + (a_{n-1} + u_n) \cdot [111...10]_n + u_n \cdot [111...11]_n = (a_1 \cdot [x_10]_n + a_2 \cdot [x_20]_n + ... + a_{n-2} \cdot [x_{n-2}0]_n + a_{n-1} \cdot [111...10]_n)$
$+ (u_n \cdot [111...10]_n + u_n \cdot [111...11]_n) = [u_1u_2...u_{n-1}0]_n + [000...0u_n]_n = [u_1u_2...u_n]_n = [u]_n$. Consequently, $S(G)|_n - \{[0]_n\}$ span $\mathcal{B}^n$.
From (1) and (2), we can conclusion that $S(G)|_n - \{[0]_n\}$ is basis of $\mathcal{B}^n$. ∎

3. Generating Matrix

We now use the meterial from previos section to create a martrix which is used in the encoding process for linear code $\mathcal{B}^n$. By theorem 2.2, generating matrices of $\mathcal{B}^n$ are as follows:
Generating matrix [4] of $\mathcal{B}^3$ is

$$G_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Generating matrix of $\mathcal{B}^4$ is

$$G_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

We conclude that generating matrix of $\mathcal{B}^n$ is

$$G_n = \begin{bmatrix} 1 & 0 & 0 & . & . & . & 0 \\ 1 & 1 & 0 & & & & . \\ 1 & 1 & 1 & & & & . \\ . & & & . & & & . \\ . & & & & . & & . \\ . & & & & & . & 0 \\ 1 & 1 & 1 & . & . & . & 1 \end{bmatrix}.$$

or $G_n = [a_{ij}]_{n\times n}$ such that $a_{ij} = 1$ when $i \geq j$ and $a_{ij} = 0$ when $i < j$. By this generating matrix, it is easy to find that binary code $S(G)|_n$ is the set of binary code of straight line (without kink) of linear hexagonal graph.

4. Encoding and Decoding

In this section, application of the binary code of linear hexagonal graph in receiving and delivering data is demonstrated, it can be used in reality because $S(G)|_n - \{[0]_n\}$ is illustrated as the main base of $\mathcal{B}^n$. That is why only some perfect matchings are selected to generate effective linear codes in receiving and delivering the information. In fact, all perfect matchings should not have been selected to be used in receiving and delivering the information because delivering mistakes cannot be examined by using the properties of the binary code. For better illustration, the researcher exemplifies how to put the binary code of perfect matching of linear hexagonal graph to create Hamming codes in order to use in receiving and delivering information. The properties of the Hamming codes are linear code and 1-error correcting code. This means even though there is a mistake during receiving and delivering information by hydrocarbon compound such as electron moving that causes a change in position of double bonds, the binary code generated could still correct that mistake.

Hydrocarbon compounds can be used in receiving and delivering information. Moreover, it is expected that if this study is further conducted, receiving and delivering information might be more effective.

In applications of hydrocarbon graphs in a transmission of data, it should be beep in mind that a vast volume of hydrocarbon molecules may be used or transmitted in the current transmission of data. Thus, reducing the use of hydrocarbon molecules will be considerably beneficial to the transmission of data. To illustrate, a the 16-bit codeword in $\mathcal{B}^{16}$ is generated with $S(G)|_{16}$ which is a set of binary codes in hydrocarbon graphs with 16 hexagons. In transmitting this codeword above, it allows us to use at most 16 words of binary codes in the graphs which equal/represent 16 hydrocarbon molecules, while the transmission of data in reality requires a higher number of codewords and that results in a higher number of hydrocarbon molecules than expected.

From this reason, this section proposes a theorem for reducing hydrocarbon molecular numbers. We intend to reduce the use of such molecules from the highest number of

molecules equal to members of $S(G)|_n - \{[0]_n\}$ to one molecule. In this concept, rather than transmitting hydrocarbon molecules which act as generators to form a codeword, we modified one generator by inserting atoms of certain elements into any hexagon or inserting compounds between 2 hexagons, both of which can represent a codeword without using multiple generators.

Transmitting data "101" by using members of $S(G)|_3 - \{[0]_3\}$, where G is straight line (without kink) of linear hexagonal graph.

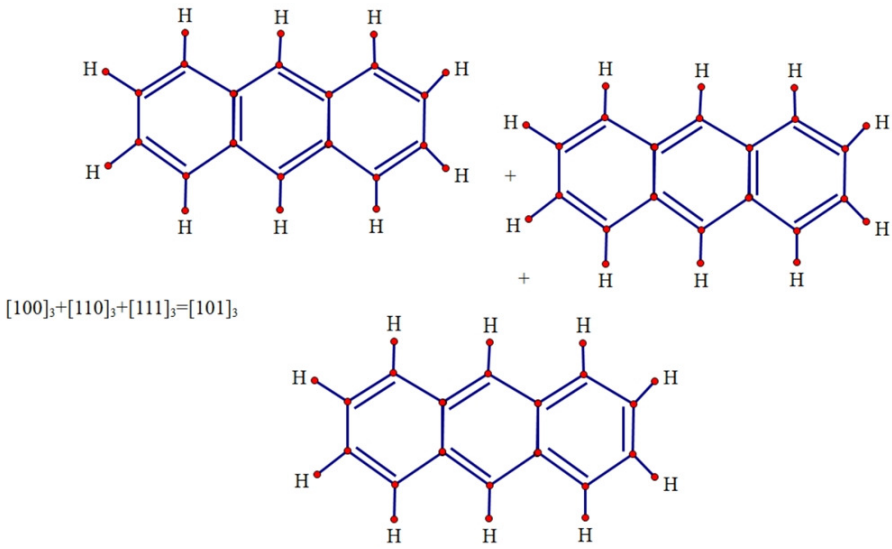


Figure 10. Transmitting data by using members of $S(G)|_3 - \{[0]_3\}$.

Transmitting data "101" by inserting atoms of certain elements into any hexagon or inserting compounds between 2 hexagons.

(1) Compressing data by adding chlorine atom in place of hydrogen atom to a selected hexagon to add 1 to a bit of a codeword.

$$[101]_3 = [100]_3 + [001]_3$$

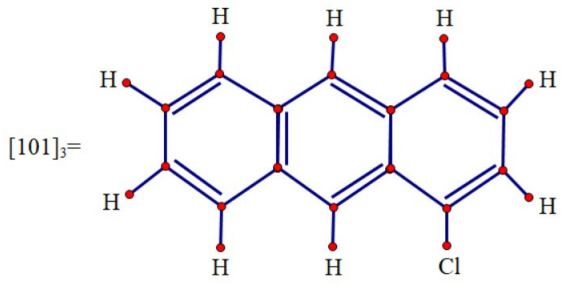


Figure 11. Diagram shows data compression by modification of atoms in compounds.

(2) Compressing data by inserting carbon atoms and hydrogen atoms between hexagonal chains to add 1 to two adjacent bits.

$$[101]_3 = [110]_3 + [011]_3$$

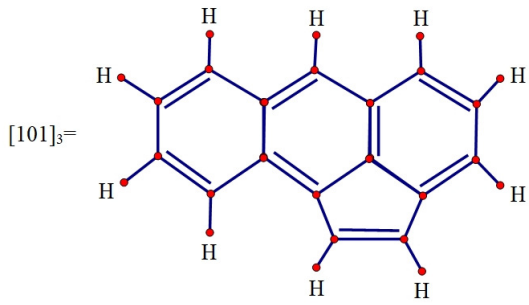


Figure 12. Diagram shows data compression by modification of atoms in compounds.

Thus, it is necessary to know whether members of $\mathcal{B}^n$ can generate such codewords. First, we would like to explain our additional lemma which is used as an instrument for proving our fundamental theorem.

Lemma 4.1. *Let $\mathcal{B}_2^n$ be the set of codeword in $\mathcal{B}^n$ with each weighs 2 and two 1-bits are adjacent.*

$$\mathcal{B}_2^n = \{[y_1]_n = [1100...0]_n, [y_2]_n = [0110...0]_n, ..., [y_{n-1}]_n = [00...011]_n\}.$$

We have,

(1) $\sum_{i\in I}[y_i]_n = [100...01]_n$ *with* $I = \{1, 2, 3, ..., n-1\}$,
(2) $\sum_{i\in I}[y_i]_n = [010...01]_n$ *with* $I = \{2, 3, ..., n-1\}$,
(3) $\sum_{i\in I}[y_i]_n = [0010...01]_n$ *with* $I = \{3, ..., n-1\}$,
.
.
.
(n-1) $\sum_{i\in I}[y_i]_n = [00...011]_n$ *with* $I = \{n-1\}$.

Proof. By rules of binary addition, it is obvious that
(1) $\sum_{i\in I}[y_i]_n = [100...01]_n$ with $I = \{1, 2, 3, ..., n-1\}$,
(2) $\sum_{i\in I}[y_i]_n = [010...01]_n$ with $I = \{2, 3, ..., n-1\}$,
(3) $\sum_{i\in I}[y_i]_n = [0010...01]_n$ with $I = \{3, ..., n-1\}$,
.
.
.
(n-1) $\sum_{i\in I}[y_i]_n = [00...011]_n$ with $I = \{n-1\}$. ∎

Theorem 4.2. $\forall [u]_n \in \mathcal{B}^n, \exists [x]_n \in S(G)|_n$ *such that* $[u]_n = [x]_n + [v]_n$ *for some* $[v]_n \in span(\mathcal{B}_1^n)$, *where* $\mathcal{B}_1^n$ *is the set of binary code in* $\mathcal{B}^n$ *with weight 1.*

Proof. Assume that $[u]_n \in \mathcal{B}^n$ such that $[u]_n = [u_1u_2u_3...u_n]_n$; $u_i \in \{0, 1\}$. We have $[u]_n = [u_1u_2u_3...u_n]_n = [u_100...0]_n + [0u_20...0]_n + [00u_3...0]_n + ... + [000...u_n]_n$. Since

$u_i \in \{0,1\}$, $[u]_n = [0]_n + \sum_{i\in I}[y_i]_n$ with $I = \{i \in \{1,2,3,...,n\} \,|\, u_i \neq 0\}$. We have $\forall i \in I, [y_i]_n \in \mathcal{B}_1^n$. Then $[u]_n = [0]_n + [v]_n$ for some $[v]_n \in span\,(\mathcal{B}_1^n)$. We know that $[0]_n \in S\,(G)\,|_n$, thus $\forall\,[u]_n \in \mathcal{B}^n$, $\exists\,[x]_n \in S\,(G)\,|_n$ such that $[u]_n = [x]_n + [v]_n$ for some $[v]_n \in span\,(\mathcal{B}_1^n)$, where $\mathcal{B}_1^n$ is the set of binary code in $\mathcal{B}^n$ with weight 1. ∎

Theorem 4.3. *$\forall\,[u]_n \in \mathcal{B}^n, \exists\,[x]_n \in S\,(G)\,|_n$ such that $[u]_n = [x]_n + [v]_n$ for some $[v]_n \in span\,(\mathcal{B}_2^n)$, where $\mathcal{B}_2^n$ is the set of binary code in $\mathcal{B}^n$ with weight 2 and 1- bits are adjacent.*

Proof. We will use mathematical induction on $n \in N, n \geq 2$. Let $p(n)$ be the statement that $\forall\,[u]_n \in \mathcal{B}^n, \exists\,[x]_n \in S\,(G)\,|_n$ such that $[u]_n = [x]_n + [v]_n$ for some $[v]_n \in span\,(\mathcal{B}_2^n)$.

(1) Consider $p(2) : S\,(G)\,|_2 = \{[00]_2\,, [10]_2\,, [11]_2\}$, we have $[00]_2 = [00]_2 + ([11]_2 + [11]_2)$, $[10]_2 = [10]_2 + ([11]_2 + [11]_2)$, $[01]_2 = [10]_2 + [11]_2$ and $[11]_2 = [00]_2 + [11]_2$. Thus $\forall\,[u]_2 \in \mathcal{B}^2, \exists\,[x]_2 \in S\,(G)\,|_2$ such that $[u]_2 = [x]_2 + [v]_2$ for some $[v]_2 \in span\,\left(\mathcal{B}_2^2\right)$.

(2) Given that $\forall\,[u]_k \in \mathcal{B}^k, \exists\,[x]_k \in S\,(G)\,|_k$ such that $[u]_k = [x]_k + [v]_k$ for some $[v]_k \in span\,\left(\mathcal{B}_2^k\right), \forall k < n$. Let $[u]_n \in \mathcal{B}^n$ and $[u]_n = [u_1u_2u_3...u_n]_n$; $u_i \in \{0,1\}$.
We know that $[u_1u_2u_3...u_{n-1}]_{n-1} \in \mathcal{B}^{n-1}$. By inductive hypothesis, we have $\exists\,[x]_{n-1} = [x_1x_2x_3...x_{n-1}]_{n-1} \in S\,(G)\,|_{n-1}$ such that $[u_1u_2u_3...u_{n-1}]_{n-1} = [x]_{n-1} + [v]_{n-1}$ for some $[v]_{n-1} \in span\,\left(\mathcal{B}_2^{n-1}\right)$. Since $[u]_n \in \mathcal{B}^n$, then $u_n = 0$ or $u_n = 1$.

Case I. $u_n = 0$, we have $[x_1x_2x_3...x_{n-1}0]_n \in S\,(G)\,|_n$ and $[u_1u_2u_3...u_{n-1}0]_n = [x_1x_2x_3...x_{n-1}0]_n + [v0]_n$ for some $[v0]_n \in span\,(\mathcal{B}_2^n)$.

Case II. $u_n = 1$, we categorize to several cases according to members in $S\,(G)\,|_n$,

(II.1) If $[x_1x_2x_3...x_{n-1}]_{n-1} = [000...00]_{n-1} \in S\,(G)\,|_{n-1}$. By definition of $S\,(G)\,|_n$, we have $[100...000]_n \in S\,(G)\,|_n$. From Lemma 4.1, we know that $[100...01]_n$ $\in span\,(\mathcal{B}_2^n)$. Thus $[x_1x_2x_3...x_{n-1}1]_n = [000...01]_n = [100...00]_n + [100...01]_n$ which is

$$\begin{aligned}[u_1u_2u_3...u_{n-1}u_n]_n &= [u_1u_2u_3...u_{n-1}1]_n \\ &= [x_1x_2x_3...x_{n-1}1]_n + [v0]_n, \text{ where } [v0]_n \in span\,(\mathcal{B}_2^n) \\ &= [000...01]_n + [v0]_n \\ &= ([100...00]_n + [100...01]_n) + [v0]_n \\ &= [100...00]_n + ([100...01]_n + [v0]_n),\end{aligned}$$

where $[100...01]_n + [v0]_n \in span\,(\mathcal{B}_2^n)$.

(II.2) If $[x_1x_2x_3...x_{n-1}]_{n-1} = [100...00]_{n-1}$. By definition of $S\,(G)\,|_n$, we have $[000...00]_n$ $\in S\,(G)\,|_n$. From Lemma 4.1, we know that $[100...01]_n \in span\,(\mathcal{B}_2^n)$. Thus $[x_1x_2x_3...x_{n-1}1]_n = [100...01]_n = [000...00]_n + [100...01]_n$ which is

$$\begin{aligned}[u_1u_2u_3...u_{n-1}u_n]_n &= [u_1u_2u_3...u_{n-1}1]_n \\ &= [x_1x_2x_3...x_{n-1}1]_n + [v0]_n, \text{ where } [v0]_n \in span\,(\mathcal{B}_2^n) \\ &= [100...01]_n + [v0]_n \\ &= ([000...00]_n + [100...01]_n) + [v0]_n \\ &= [000...00]_n + ([100...01]_n + [v0]_n),\end{aligned}$$

where $[100...01]_n + [v0]_n \in span\,(\mathcal{B}_2^n)$.

(II.3) If $[x_1x_2x_3...x_{n-1}]_{n-1} = [110...00]_{n-1}$. By definition of $S\,(G)\,|_n$, we have $[1110...00]_n$ $\in S\,(G)\,|_n$. From Lemma 4.1, we know that $[0010...01]_n \in span\,(\mathcal{B}_2^n)$. Thus $[x_1x_2x_3...x_{n-1}1]_n = [110...01]_n = [1110...00]_n + [0010...01]_n$ which is

$[u_1u_2u_3...u_{n-1}u_n]_n = [u_1u_2u_3...u_{n-1}1]_n$
$= [x_1x_2x_3...x_{n-1}1]_n + [v0]_n$, where $[v0]_n \in span\,(\mathcal{B}_2^n)$
$= [110...01]_n + [v0]_n$
$= ([1110...00]_n + [0010...01]_n) + [v0]_n$
$= [1110...00]_n + ([0010...01]_n + [v0]_n)$,
where $[0010...01]_n + [v0]_n \in span\,(\mathcal{B}_2^n)$.

(II.4) If $[x_1x_2x_3...x_{n-1}]_{n-1} = [1110...00]_{n-1}$. By definition of $S\,(G)\,|_n$, we have $[11110...00]_n \in S\,(G)\,|_n$. From Lemma 4.1, we know that $[00010...01]_n \in span\,(\mathcal{B}_2^n)$. Thus $[x_1x_2x_3...x_{n-1}1]_n = [1110...01]_n = [11110...00]_n + [00010...01]_n$ which is
$[u_1u_2u_3...u_{n-1}u_n]_n = [u_1u_2u_3...u_{n-1}1]_n$
$= [x_1x_2x_3...x_{n-1}1]_n + [v0]_n$, where $[v0]_n \in span\,(\mathcal{B}_2^n)$
$= [1110...01]_n + [v0]_n$
$= ([11110...00]_n + [00010...01]_n) + [v0]_n$
$= [11110...00]_n + ([00010...01]_n + [v0]_n)$,
where $[00010...01]_n + [v0]_n \in span\,(\mathcal{B}_2^n)$.

.
.
.

(II.n-1) If $[x_1x_2x_3...x_{n-1}]_{n-1} = [1111...10]_{n-1}$. By definition of $S\,(G)\,|_n$, we have $[1111...10]_{n-1} \in S\,(G)\,|_n$. From Lemma 4.1, we know that $[000...011]_n \in span\,(\mathcal{B}_2^n)$. Thus $[x_1x_2x_3...x_{n-1}1]_n = [111...101]_n = [1111...10]_n + [0000...011]_n$ which is
$[u_1u_2u_3...u_{n-1}u_n]_n = [u_1u_2u_3...u_{n-1}1]_n$
$= [x_1x_2x_3...x_{n-1}1]_n + [v0]_n$, where $[v0]_n \in span\,(\mathcal{B}_2^n)$
$= [111...101]_n + [v0]_n$
$= ([1111...10]_n + [0000...011]_n) + [v0]_n$
$= [1111...10]_n + ([0000...011]_n + [v0]_n)$,
where $[0000...011]_n + [v0]_n \in span\,(\mathcal{B}_2^n)$.

(II.n) If $[x_1x_2x_3...x_{n-1}]_{n-1} = [1111...11]_{n-1}$. By definition of $S\,(G)\,|_n$, we have $[1111...11]_{n-1} \in S\,(G)\,|_n$. Thus $[x_1x_2x_3...x_{n-1}1]_n = [1111...111]_n \in S\,(G)\,|_n$ which is
$[u_1u_2u_3...u_{n-1}u_n]_n = [u_1u_2u_3...u_{n-1}1]_n$
$= [x_1x_2x_3...x_{n-1}1]_n + [v0]_n$, where $[v0]_n \in span\,(\mathcal{B}_2^n)$
$= [111...111]_n + [v0]_n$
where $[v0]_n \in span\,(\mathcal{B}_2^n)$.

Conclude that $\exists\,[x]_n \in S\,(G)\,|_n$ such that $[u]_n = [x]_n + [v]_n$ for some $[v]_n \in span\,(\mathcal{B}_2^n)$. By mathematical induction, it can be concluded that $\forall\,[u]_n \in \mathcal{B}^n, \exists\,[x]_n \in S\,(G)\,|_n$ such that $[u]_n = [x]_n + [v]_n$ for some $[v]_n \in span\,(\mathcal{B}_2^n)$, where $\mathcal{B}_2^n$ is the set of binary code in $\mathcal{B}^n$ with weight 2 and 1- bits are adjacent and $n \in N, n \geq 2$. ∎

5. Conclusion

According to the results, it can be concluded that the hexagonal graph can be substituted by binary code using Klavzar's algorithm. The properties of the obtained binary code could be used in encoding and decoding. In summary, the perfect matchings of the hexagonal graph can be applied in receiving and delivering data after being replaced by the binary code.

This study discovers that edge independent sets or perfect matchings could be converted into binary codes. Given that Klavzar's study[**?**]eveloped algorithm for binary

conversion, we could collect, at ease, data of perfect matchings of hydrocarbon graphs in the form of binary codes. Moreover, we are aware that the basic use of binary codes of perfect matchings in hydrocarbon graphs lay in communication. Hence the present study seeks to investigate properties of such binary codes based on coding theory and finds that they could be used as a tool for the transmission of data. Nevertheless, if there is a need to use hydrocarbon molecules to transmit data or information, that will lead to a number of research problems in many fields, such as engineering, information technology, and chemistry. Hence, this study establishes theorems 4.2 and 4.3 to facilitate real applications.

It is hoped that the results of this study would encourage more research on hydrocarbon graphs in relation to both graph theory and coding theory, which would promote a clearer understanding on hydrocarbon molecules. It could also provide practical results, which would serve as a reminder that mathematics is a basic component of any development.

6. Suggestions for Further Study

Researchers who would like to learn more about independent set of hydrocarbon graph can do in both chemistry and mathematics perspectives. There are 2 interesting issues be further explored.

1. Seeking for more efficient methods than the one of Klavzar to convert perfect matchings of hexagonal graph into binary code.
2. The study of the results of this reseach might be applicable for receiving and delivering information by utilizing hydrocarbon compounds so that it might be used effectively in reality.

References

[1] A. Balaban, F. Harary, Chemical graphs–V: Enumeration and proposed nomenclature of benzenoid cata-condensed polycyclic aromatic hydrocarbons, Tetrahedron, 24 (6) (1968) 2505–2516.

[2] S. Klavžar, V. Aleksander, P. Žiger, On resonance graphs of catacondensed hexagonal graphs: structure, coding, and Hamilton path algorithm, Match Communications in Mathematical and in Computer Chemistry 49 (2003) 99–116.

[3] S. Klavžar, A. Vesel, P. Zigert, I. Gutman, Binary coding of Kekul structures of catacondensed benzenoid hydrocarbons, Computers and Chemistry 25 (6) (2001) 569–575.

[4] J.H. Lint, Introduction to Coding Theory, Springer-Verlag, New York, 1999.