EMBEDDED SYSTEM THAT DETECTS ISOMORPHISM AMONG GRAPHS USING NEW HAMMING NUMBER TECHNIQUE

[1]E. Sathesh, [2]Shunmugam Anandaraj
[1,2]Lecturer II
[1,2]School of Computer Science & Information Technology,
[1,2]DMI – St. John the Baptist University, Mangochi, Malawi.

Abstract*—Isomorphism means if two objects are isomorphic there isomorphism exists. It means the objects have same number of vertices and same number of degrees and shapes. It means an isomorphism whose source and destination coincide. This paper shows that detects isomorphism among graphs using new humming number technique. There are lot of approaches are there to detect the isomorphism but there are some disadvantages are there in detection. This paper shows the way to overcome those disadvantages. The new humming number techniques measures the extent to which two links differ from each other. Detection of isomorphism will reduces the number of substitutes to be appraised to suit a specified job. Using this humming number technique embedded system also reduces the number of alternatives.

***Keywords*—Isomorphic, vertices, coincide, extent, substitutes, appraised.**

I. Introduction To Isomorphism

"Basically isomorphism means structural equivalence between two graphs".

Two graphs are isomorphic if there is a one to one correspondence between the links of one chain and those of second chain such that two links of a chain are joined by a joint if and only if the corresponding links of the other chain are joined by a joint. If there is no such correspondence the two chains are said to be "non isomorphic". A graph with a given number of elements or links or members, can be arranged in different ways. Detection of isomorphism reduces number of alternatives to be evaluated to suit a specified job. For chains with lesser number of links, structural similarity can be found by just looking at the possible arrangements. But as the number of links increases, thereby increasing the possible number of arrangements, so the detection of isomorphism becomes very complex.

Even the usage of Franke's diagram leads to a compact notation for chains but does not avoid the difficulty regarding isomorphism detection. In view of this many methods have been developed which are

1. Characteristic polynomial based approaches
2. Code based approaches
3. Distance or path based approaches
4. Hamming number based approaches

I.A) Characteristic Polynomial Approach

Isomorphism can be found by computing the characteristic polynomials of the polynomials of the different linkages. The polynomial can be found by using the following mathematical equation

Polynomial = [XI-D]   where X is a parameter

I=Identity matrix of the linkage of order n.

N = no of links in that linkage

D = distance matrix

By comparing the polynomials of the linkage s we can say that the linkages are isomorphic if the polynomials of the two linkages are the same. Otherwise they are not isomorphic to each other.

I.B) Code Based Approach

To compare a pair of chains for isomorphism their adjacency matrices are both brought to canonical form and comparison is made between codes of chains, the code itself being a suitably decodable function of corresponding matrix. This approach consists of carrying out a number of permutations and combinations of link labels of chain so that the upper triangular adjacency matrix written as a binary sequence becomes maximum or minimum based on which MAXCODE or MINCODE is given.

I.C) Distance/Path Based Approach

This approach defines a linkage path code of graph with n-links in an ordered sequence of numbers. It presents a three step test for isomorphism which in worst case would require comparison of link assortments, planetary or otherwise of the associated graphs and an index based on joint-joint distances in the chains to conclude that pair of chain is isomorphic are not.

DISADVANTAGES OF THESE APPROACHES ARE:

1. As the number of links increases the order of the distance matrix increases, thereby increasing the order of the polynomial.
2. It is not suitable for higher order linkages.
3. It is very difficult to be computed in computer synthesis.
4. These methods give wrong answers in peculiar cases.

To overcome all these difficulties we go for the new hamming number technique.

I.D) HAMMING METHOD

Whether or not each link in a chain is connected directly to any other link is split out in a connectivity matrix C, whose elements either all zero or one. C is of size n*n for a chain of n links and the value of each element in C is decided as follows: If link i is directly connected to link j then the element in the ith row and jth column of C is 1. If the elements ar not connected, the element is zero. The connectivity matrix for the four bar graph will be as shown below.

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

The Connectivity Matrix

Matrix 1: Connectivity Matrix

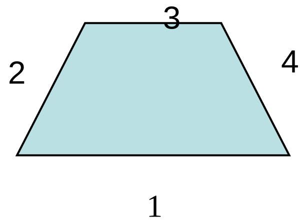

Figure 1: Four-bar Chain

It may be observed that every row represents that particular link and every element in that row indicates whether or not that link is directly connected to the other links. Also, every element of the connectivity matrix informs us merely whether or not two links are directly connected. But the relation between any two links is much more than this single bit of information. How they relate in the above four bar, links 1 and 2 are directly connected. Link 3 is connected to 2 but not to 1. Link 4 is connected to 1 but not to 2. Hence the relationship between links 1 and 2 has to take all these facts in to consideration. This is what the hamming number of two links precisely means. It measures the extent to which two links differ from each other.

The hamming number for any two codes each with n digits has been defined as the total number of bits in which the two codes differ. Applying this definition to the rows i and j of C, we have

hij = S k
s k = 0, if c ik = c jk
= c ik + c jk, otherwise.

The hamming number between any two rows of C of size n can be positive integer from n. If all the digits are different, down to 0 if the two rows are exactly identical. It may be pointed out that here in passing that the original definition by hamming specified that for c ik! = c jk, s k =1. The definition here is s k = c ik + c jk and there is no perceptible effect on account of this slight modification. But its advantage will be evident while considering the secondary hamming string, a concept explained later in this paper. In other words, the hamming number of any two rows is the sum of all the scores for each of the columns of those rows. A score in turn determined as i) the sum of the individual elements if they are unequal and ii) zero if the elements are equal.

The Hamming numbers for all the other pairs of rows are calculated and we obtain the hamming number matrix as H=[h ij]. Thus for the four bar chain of fig 1 the hamming matrix is

$$\begin{matrix} 0 & 4 & 0 & 4 \\ 4 & 0 & 4 & 0 \\ 0 & 4 & 0 & 4 \end{matrix}$$

Hamming

Matrix 2: Hamming Matrix

The hamming matrix is also square, symmetrical and contains zeros all along the leading diagonal However unlike the connectivity matrix it contains digits which could be larger than unity.

- ***Link Hamming*** of all the elements in the ith row of the hamming matrix. Thus the link hamming number of link 1 of fig 1 is 8.
- ***Chain Hamming number :***Chain Hamming number for any chain is the sum of all the link hamming numbers of that chain. It also works out to be the sum of all the elements of the hamming matrix for that chain.
  The chain hamming number for the above chain is 32.
- ***Link hamming string:***Link hamming string for any link i is the string obtained by concatenating a) the link hamming number of i with 2) the frequency of occurrence of all the integers from n down to 0, in the hamming numbers of that row i.
  The link hamming string for link 1 is 8, 2002, implying that the link hamming number is 8 and comprises of two 4s, no 3s, no 2s no 1s and two 0s.
  The link hamming strings for the four links are:

  1: 8, 20002 2:8, 20002 3: 8, 20002 4: 8, 20002
- ***Chain hamming string*** **:**chain hamming string is defined as the concatenation of the (i) chain hamming number and (ii) link hamming strings, these strings placed in decreasing order of magnitude.
  - The chain hamming string for the above chain is defined as
    - 32, 8, 20002, 8, 20002, 8, 20002, 8, 20002.

II. ISOMORPHISM

The chain hamming string is the definitive test for isomorphism among graphs.

This implies that if two chains are known to be isomorphic, their chain hamming strings should be identical and vice versa. Secondly, if two chains are non isomorphic their chain hamming strings differ at some position or the other. The four bar chain in figure 1 can have 24 different sets (factorial 4) of labels for the four links by mere relabelling the links. However all these chains have the same chain hamming string

32, 8, 20002, 8, 20002, 8, 20002, 8, 20002.

These imply that all the 24 chains are isomorphic.

.But when isomorphism is considered for 10 bar linkage as shown in the figure below another phenomenon is observed. The 10 bar linkage and corresponding hamming matrix will be as shown below.

TEN-BAR LINKAGE HAMMING MATRIX

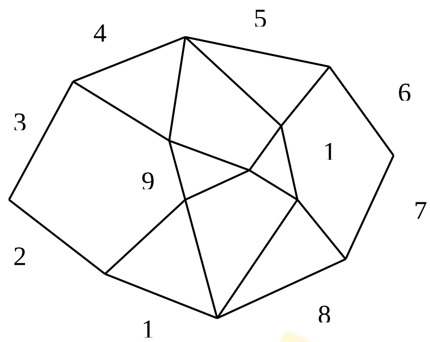

Figure 2: Ten-bar Linkage

0	5	3	4	6	5	3	6	6	2
5	0	4	3	5	4	4	3	3	5
3	4	0	5	3	4	4	5	3	5
4	3	5	0	6	3	5	6	6	2
6	5	3	6	0	5	3	4	2	6
5	4	4	3	5	0	4	3	5	3
3	4	4	5	3	4	0	5	5	3
6	3	5	6	4	3	5	0	2	6
6	3	3	6	2	5	5	2	0	6
2	5	5	2	6	3	3	6	6	

Matrix 3: Ten-bar Linkage Matrix

An interesting phenomenon has been observed while computing the hamming numbers for ten bar chains. All chains belonging to any one family have common chain hamming number. The families and the chain hamming numbers are as shown

Chain Family	Serial Numbers	Family Population	Chain Hamming No.
0064	1-50	50	380
0145	51-145	95	376
0226	146-202	57	372
1036	203-217	15	368
0307	218-220	3	368
1117	221-228	8	364
2008	229,230	2	256

Table 1: Chain Family Table

One remarkable feature is that not even one out of the total 230 families has disobeyed the rule that the chain hamming numbers of all members of a family are equal. *But the chains 1 and 40 have shown different chain hamming strings though they are isomorphic, so to overcome this we go for secondary hamming technique.*

III. Secondary Hamming Technique

In the process of checking isomorphism, two pairs 1 and 40 have same chain hamming strings although they are different. This can be overcome by secondary hamming technique that is similar to primary hamming technique.

1. Form the adjacency matrix.
2. Apply primary hamming technique to adjacency matrix, to find the primary hamming matrix.
3. Compare the row to row equivalence of the hamming matrix. The elements that occupy identical positions in the respective rows only should be compared.
4. Add the elements, which are different in the rows being compared.
5. Form the sum matrix, which is the required secondary hamming matrix..

0	40	80	48	76	76	76	76	62	26
40	0	48	80	76	76	76	76	62	26
80	48	0	40	76	76	76	76	26	62
48	80	40	0	76	76	76	76	26	62
76	76	76	76	0	40	56	24	74	74
76	76	76	76	40	0	24	56	74	74
76	76	76	76	56	24	0	40	74	74
76	76	76	76	24	56	40	0	74	74
62	62	26	26	74	74	74	74	0	76
26	26	62	62	74	74	74	74	76	0

Matrix 4: Adjacency Matrix

0	80	80	8	76	76	76	76	78	10
80	0	8	80	76	76	76	76	10	78
80	8	0	80	76	76	76	76	10	78
8	80	80	0	76	76	76	76	78	10
76	76	76	76	0	56	56	8	74	74
76	76	76	76	56	0	8	56	74	74
76	76	76	76	56	8	0	56	74	74
76	76	76	76	8	56	56	0	74	74
78	10	10	78	74	74	74	74	0	76
10	78	78	10	74	74	74	74	76	0

Matrix 5: Secondary Hamming Matrix

These are the secondary hamming matrices for chains 1 and 40.
The chain secondary hamming string for chain no 1 is:
5624, 4*572(4*76, 2*74, 1*56, 1*40, 1*24, 1*0), 4*560(1*80, 4*76, 1*62, 1*48, 1*40, 1*26, 1*0), 2*548(1*76, 4*74, 2*62, 2*26, 1*0)
The chain secondary hamming string for chain no 40 is:
5624, 4*572(4*76, 2*74, 2*56, 1*8, 1*0), 4*560(2*80, 1*78, 4*76, 1*10, 1*8, 1*0), 2*548(2*78, 1*76, 4*74, 1*10, 1*0)
the two strings differ and hence chains nos 1 and 40 are not isomorphic.
Now we will go on to the design of embedded system that detects isomorphism among two graphs using new hamming number technique with an introduction to isomorphism.

IV. Embedded System

"Embedded Systems are computer systems, which are wholly contained within some other device, used to control, monitor or assist the operation of equipment, machinery or plant".

Characteristics of Embedded systems are:
Single-functioned:
An embedded system usually executes a specific program repeatedly. For example, a pager is always a pager. In contrast, a desktop system executes variety programs, like spreadsheets, word processors, and video games, with new programs added frequently.

Tightly constrained:
All computing systems have constraints on design metrics, but those on embedded systems can be especially tight. A design metric is a measure of systems often must cost just a few dollars, must be sized to fit on a single chip, must perform fast enough to process data in real time, and must consume minimum power to extend battery life or prevent the necessity of a cooling fan.

Reactive and real time:
Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without delay. For example, a car's cruise controller continually monitors and
For creating a data path the following four steps are followed:
1. We first create a register for any declared variable i.e., x and y, treating output port as an implicit variable, we create a register d and connect it to the output port. These are shown in light grey rectangles.
2. We create a functional unit for each arithmetic operation in the state diagram. We perform two subtractions, one comparison for less than, one comparison for inequality, so we use two subtractions and two comparators as shown as white rectangles.
3. We connect the ports, registers, and functional units .For each write to a variable in the state diagram, we draw a connection from writes source to the variables register. For arithmetic and logical operations we connect the sources to an input of the operations corresponding functional unit. When more than one source is connected to a register, we add approximately sized multiplexer shown in dark rectangles.
4. Finally we create a unique identifier for each control input and output of the data path components such as x_sel and x_neq_y in the diagram.Now our data path is created and will be as shown in the figure.

V. Creating FSM And Optimizing FSM

Now our FSMD is modified in to FSM .it has same states and transitions as the fsmd We replace complex actions and conditions by Boolean ones, making use of our data path .we replace every variable write by actions that set the select signals of the multiplexer in front of the variable's register such that the write's source passes through, and we assert the load signal of that register. We replace every logical operation in a condition by the corresponding functional unit control output. Any signal not explicitly assigned in a state is implicitly assigned to zero. State minimization in which all the states that have same input combinations and same outputs are merged together to get a single state.

VI. Creating Controller

We then complete the controller design by implementing the fsm using our sequential design techniques. The state table for controller design will have seven inputs of the controller and so we will get 128 rows in the table. Since it is tedious to solve such large tables using our k-maps and other techniques we use CAD (computer aided design) tools and the final controller will be as shown below.

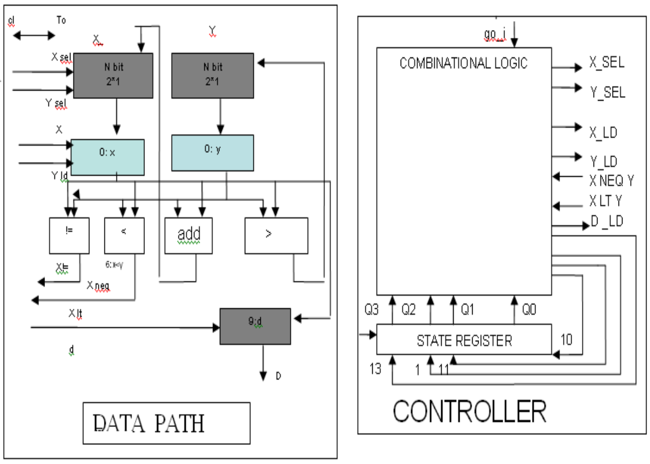

Figure 3: CAD Tools and Controller

VII. Fantasies Embedded In Real Time

VII. a)In the driving seat:

Embedded systems can be used to implement features ranging from adjustment of the suspension to suit road conditions and the octane content in the fuel to antilock braking systems (ABS) and security systems. Embedded systems can also make driver-less vehicle control a reality.. One such technology is Adaptive Cruise Control (ACC) from Ford. ACC allows cars to keep safe distances from other vehicles on busy highways. The driver can set the speed of his car and the distance between his car and others.

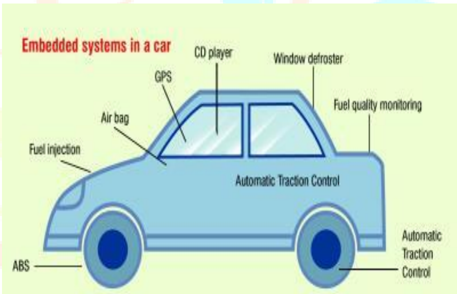

Figure 4: Antilock Braking System

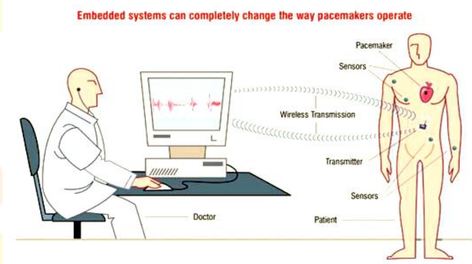

Figure 5: Adaptive Cruise Control

VII.b)The doctor will see you now:

Embedded technology advances are pointing towards the use of pacemakers that can be transplanted in or near the heart itself. The pacemaker will be able to monitor parameters like blood pressure, blood flow, pulse rate, temperature, etc, using micro sensors planted in various parts of the body. This capability will enable the pacemaker to automatically vary its operation to suit the changing body conditions.

VIII. Applications In Daily Life

Embedded systems are found in a variety of common electronic devices such as consumer electronics (cell phones, pagers, digital cameras, camcorders, videocassettes recorders, portable video games, calculators, and personal digital assistants) and many more.

IX. Results

Unless and until a new approach is supported by results of practical experiments it is not acceptable by simply saying that it will take minimum time. So we have an optimized code written in C for determining whether the two graphs are isomorphic or not using this technique.

The results are very interesting and it took only 4 sec for small graphs and maximum of 10 sec for very bigger graphs. Also other techniques may fail in some special cases but this technique never fails.

Conclusion

1. Most other methods use a characteristic polynomial in some form or the other and so the number of terms available for one to one comparisons of two chains is at best equal to one. On the other hand the number of terms available in this method is at best (n+1)2 and at worst (n+1).
2. This method is extremely simple and reliable both in formulation stage and in the execution stage which makes the computer computation very easy and reliable than other methods and *takes less time than any other one.*

3. Also the embedded system designed in this paper is very useful in the field of Robotics where the usage of isomorphism is very frequent and this can used to know the different arrangements .

REFERENCES

[1] Philip Koopman, "Better Embedded System Software",Drumnadrochit Education LLC, 2010
[2] Michael Barr, "Programming embedded systems in C and C++", Shroff Publishers & Distributors, 2004
[3] Christopher Hallinan,"Embedded Linux Primer", Pearson Education, 2010
[4] David Kleidermacher, Mike Kleidermacher, "Embedded Systems Security", Elsevier, 2012
[5] Tim Wilmshurst, "Designing Embedded Systems with PIC Microcontrollers",Newnes, 2006
[6] Jonathan W. Valvano, "Embedded Microcomputer Systems", Thomson, 2007
[7] AhmetBindal, "Electronics for Embedded Systems", Springer International Publishing, 2017
[8] Rob Toulson, Tim Wilmshurst, "Fast and Effective Embedded Systems Design", Elsevier, 2012
[9] Jack G. Ganssle, "The Art of Programming Embedded Systems", Academic Press, 1992
[10] Tammy Noergaard, "Embedded Systems Architecture", Newnes, 2013