© 2014 IJIRT | Volume 1 Issue 5 | ISSN : 2349-6002

GRAPH HOMOMORPHISM

Simran Khurana, Angad Kataria
Student,Department Of Information Technology
Dronacharya College Of Engineering,Gurgaon

***Abstract-* this paper aims to give a focused introduction to algebraic graph theory accessible mathematically. We will begin by giving some standard definitions, then expanding our focus to specially study the properties of graph homomorphism. A homomorphism from a graph G to a graph H is a map from V G to V H which takes edges to edges. Homomorphism are a generalisation of graph colourings. A homomorphism from the graph G to the complete graph Kr (with vertices numbered1, 2, . . . , r) is exactly the same as an r-colouring of G (where the colour of a vertex is its image under the homomorphism), since adjacent vertices map to distinct vertices of the complete graph. In particular, we will be discussing the idea of graph colouring by abstracting what it means for a graph to be colourable ,and expanding this abstraction to deal with more complex instances of the graph colouring problem.**

I. INTRODUCTION

Since we are going to be working in the field of graph theory, it is necessary to define precisely what we mean by a graph (for the advanced reader, we will only be working with simple graphs keeping definitions to a minimum, however the ideas presented will translate easily into more complex definitions for graphs). A graph G consists of a set of vertices V (G) and a set of edges E(G) represented by unordered pairs of vertices. It is important to note that a graph may have many different geometric representations, but we just use these as visualization tools and focus on V (G) and E(G) for our analysis. We define vertices x and y to be adjacent if (x; y) 2 E(G), for our purposes we will write x ~ y to ease the notation but let the reader note that this is not an equivalence relation. It comes as no surprise to an algebraist that graphs have sub graphs, we say Y is a sub graph of X if V (Y) V (X) and E(Y) _ E(X). A graph is complete if every vertex is connected to every other vertex, and we denote the complete graph on n vertices by Kn. Finally, we define a graph with no edges to be an independent set of vertices. [1] Now, we will continue our definitions by applying the notions of isomorphisms and homo-morphisms to graphs. An isomorphism between two graphs G and H is a bijective mapping f : G 7! H with the property (x; y) 2 E(G) , (f(x); f(y)) 2 E(H):[1] This simple definition comes as no surprise to those who have worked with isomorphisms in other contexts. We say that a graph isomorphism respects edges, just as group, field, and vector space isomorphisms respect the operations of these structures. We now consider a weakening of this definition to arrive at graph homomorphisms. A homomorphism from a graph G to a graph H is defined as a mapping (not necessarily bijective) h : G 7! H such that
(x; y) 2 E(G)) (h(x); h(y)) 2 E(H)
We say that a graph homomorphism preserves edges, and we will use this definition to guide our further exploration into graph theory and the abstraction of graph colouring.

Example. Consider any graph G with 2 independent vertex sets V1 and V2 that partition
V (G) (a graph with such a partition is called bipartite). Let V (K2) = f1; 2g, the map
f : G 7! K2 defined as
f(v) =
8<
:
1; if v 2 V1
2; if v 2 V2
is a graph homomorphism because no two adjacent vertices in G map to the same vertex in K2, and thus our edges are preserved. A special c homomorphism to K2 of this form is illustrated by where the G is the domain and K2 is the codomain of the homomorphism. Notice that both of the above graphs can be properly coloured with two colours, as we will be returning to this idea momentarily.

II. GRAPH COLOURING

One of the more famous results in graph theory is the 4 colour map theorem which states that any conceivable map can be coloured with 4 colours such that no two regions sharing a border have the same colour, but how could we go about proving such a thing? This next section is devoted to giving a precise definition to a colouring of a graph, as well as giving a precise mathematical meaning to the phrase \this graph is n-colourable." As always, we begin with3 a definition. A proper colouring of a graph G is an assignment of colours to V (G) such that no two adjacent vertices have the same colour. A graph's chromatic number is the minimum number of colours needed to properly colour the graph. We say that a graph is n-colourable if it can be properly coloured with n colours. Here we arrive at our first theorem concerning graph homomorphisms.

Theorem 1. A graph G is r-colourable , there exists a homomorphism from G to Kr
Proof.
Assume G can be properly coloured with r colours labeled (1; 2,......r). We label a generic vertex v of G by vi if it is coloured by i where $1 \leq i \leq r$. Define a mapping h : G -> Kr by h(vi) = ki where ki is the ith vertex in Kr (without loss of generality we can impose an ordering on the vertices of Kr). We claim that this map is a graph homomorphism:
Let a, b £ V (G) and assume a ~ b (a is adjacent to b)
a and b do not have the same colour so a = a i and b = b j where i 6= j.
h(a) = h(a i) = k i 6= kj = h(b j) = h(b) so h(a) 6= h(b)
h(a) _ h(b) (the image of h is a complete graph)
Therefore, a _ b) h(a) _ h(b) showing that h is a homomorphism from G to Kr

Let h : G 7! Kr be a graph homomorphism. For a given y 2 V (Kr) define the set h□1(y) _V (G) to be h□1(y) = fx 2 V (G) j h(x) = yg:
It is important to remember that these sets might be empty. Let a; b 2 h□1(y),) h(a) = h(b) = y (a; b 2 h□1(y))
) h(a) 6_ h(b) (a vertex cannot be adjacent to itself)
) a 6_ b (contra positive of a _ b) h(a) _ h(b)).
So, for every y 2 V (Kr) the corresponding h□1(y) is an independent subset of V (G).
Therefore, every element of h□1(y) can be coloured using the same colour. There are r vertices of Kr, and the sets h□1(y) form a partition of the vertices of G (partition comes from properties of a map, we can't assign one input to two different outputs and every input must map to some element of the co domain), so there are at most r colours needed to properly colour G.
This theorem gives us a concrete mathematical interpretation of coloring a graph, but the above result answers much more than the simple question of whether or not a graph isn-colourable. As a simple consequence of this theorem, the chromatic number of a graph G is the smallest integer r such that there exists a homomorphism from G to Kr. In addition to this small logical step, a number of facts become apparent from the following corollary.
Corollary. A colouring of a graph G is precisely a homomorphism from G to some complete graph.
If we view a homomorphism h : G 7! Kr for some graph G as an assignment of colours to the vertices of G, then h directly tells us how to create this colouring. For any a 2 V (G), if h(a) = ki then we simply assign colour i from a set of r colours to vertex a. This process gives us a unique colouring of G for every h. If we want to know how many proper colourings of G we can create with r colours, we simply count the number of distinct homomorphisms from. G to Kr.

Example. Consider the following coloring of the Petersen graph P represented as a ho-momorphism from P 7! K3 where vertices of P map to their corresponding color in K3.Given our previous discussion, it should be fairly obvious that this map is indeed a graph homomorphism.

Consider the problem of scheduling final exams such that no student is scheduled to take two exams during the same period. We can construct a graph G which has classes as its vertices, where two vertices are adjacent if there is a student taking both classes. We now have a graphical representation G of the current state our classes, and we wish to assign final periods to the vertices of G such that no two adjacent vertices have the same period assignment (otherwise this would constitute a scheduling conict). Finding such a scheduling of our classes is equivalent to finding a proper colouring of G, where each colour represents a different exam period.

Example. If we have 6 final periods with exams already scheduled such that no student must take consecutive exams, we can model this schedule as where each vertex is a final exam period and there exists an edge between vertices if a student can be scheduled to take an exam in both time slots.

Once we have found a graphical representation of a schedule satisfying the constraints of our problem, we must ask if such a scheduling is possible, and if so how can we achieve it. To answer these questions we can again construct a graphical representation G of our current classes (classes are vertices of G while edges represent classes with mutual students) and find a homomorphism from G to our desired schedule represented by a graph H.

III. HOMOMORPHISMS

So, how can we find h : G 7! H given arbitrary graphs G and H as inputs? We will not be explicitly answering this question as it is still an open area of research, and any discussion of an exact algorithm for this problem would be beyond the scope of this paper. Instead, we will now be discussing the mathematics of graph homomorphisms in order to further understand the complexity of this problem. As a result, we will discover special c instances of G and H in which we can easily determine that there is no homomorphism from G to H. However, we must first present a few more definitions.
To ease notation, we write G ! H if there exists a graph homomorphism from G to
H. A clique of a graph G is a sub graph of G which is complete. We say that a graph has clique number n, denoted !(G), if the largest clique of G has n vertices. Just as _(G) is the smallest r such that G ! Kr, !(G) is the smallest n such that K n ! G. [2] We state this as a theorem for consistency.

Theorem 2. For any graph G, !(G) = r , r is the largest integer such that Kr ! G.

Proof. If G has a clique of size n labeled C, then any isomorphism from Kn to C inducesa homomorphism from Kn to G (the fact that C is isomorphic to Kn should be obvious,considering they are both complete graphs on n vertices). Conversely, if Kn ! G then eachvertex of Kn must have a unique image in G, where all of these images must be connectedin order to preserve the edges of Kn. These images must then form a clique of size n in G.Given this necessary and su_cient condition, it directly follows that !(G) = r , Kr ! Gwhere r is maximal.

We have already de_ned the chromatic number of a graph G, denoted _(G), as the minimum number of colors needed to properly color G. The following theorem describes a useful relationship between both _(G) and _(H), and !(G) and !(H) respectively, where H is the codomain of a homomorphism from G.

Theorem 3. If G ! H then !(G) _ !(H) and _(G) _ _(H).

Proof. The compositions of homomorphisms is again a homomorphism, i.e. G ! H ! Kimplies G ! K. Given this fact,
K!(G) ! G ! H implies K!(G) ! H, so !(H) is at least !(G),
and G ! H ! K_(H) implies G ! K_(H), so _(G) is at most _(H).
Even with these simple results, we are beginning to make signi_cant process in determining if, given graphs G and H, there is no homomorphism from G to H. We can compute!(G) and _(G) for any graph G, although these computations might be di_cult. Comparing our results for G and H we can immediately tell if G does not map homomorphically to H. We will _nish our discussion of graph homomorphisms with a brief introduction to the concept of homomorphism classes. If G H and H G we say that G _ H . As one might suspect, '_' produces an equivalence relation. The relation is symmetric because the identity map from G to G is an isomorphism and thus a homomorphism, transitive because composition of homomorphisms is again a homomorphism, and reexive by nature of our de_nition. This equivalence relation forms equivalence classes of graphs, which we can use to solve our decision problem.
Since we are now dealing with equivalence classes, it is natural to ask if there are canonical representatives of these classes. We de_ne a core of a graph G to be the element of [G](homomorphism equivalence class of G) with the least number of vertices. Given thisde_nition, we get some interesting results involving these cores.

1) G is a core , every endomorphism of G is an automorphism.
2) Any graph (or homomorphism equivalence class) has a unique core unto isomorphism.

These results are certainly nontrivial, but we will not discuss the proofs here. gives su_cient proofs of these facts and a discussion of the topic of cores and homomorphism equivalence classes in the context of classifying graphs, but these results are

not crucial to our discussion. Instead, let us examine why the above results might prove useful. Since the number of vertices of the core of a graph G must be less than or equal to
jV (G)j, we assume it is more computationaly simple to _nd homomorphism from this core to a desired H than using our original G. If we denote the core of G by C(H), it should be clear that G ! H if and only if C(G) ! H. Therefore, we can re_ne our search for G ! H 10 for all graphs homomorphically equivalent to G. This is useful because cores have speci_c properties that they do not share with generic graphs (fact 1 from above for example).

These results do not explicitly help us solve the problem of determining whether G ! H because we are still faced with the problem of _nding a core. This new problem may be just as complex as our original, and if we solve it we still have not fully answered our question. If we could easily _nd a core, we could then determine if H _ G (cores of homomorphism classes are unique unto isomorphism) but even then we would be faced with an isomorphism problem, which is again nontrivial. However, deciding whether or not G ! H is true may still be more challenging than deciding if C(G) ! C(H) is true, and the preceding are equiv-alent.

REFERENCES

[1] Chris Godsil, Gordon Royle, Algebraic Graph Theory. Graduate Texts in Mathematics,2001. ISBN: 0 387 95241 1

[2] Peter J. Cameron Graph homomorphisms Combinatorics Study Group Notes, September2006 , Paper (Link)

[3] Norman Biggs, Algebraic Graph Theory. Cambridge Mathematical Library, 2nd edition,1993. ISBN: 0 521 45897 8

[4] Ronald C. Read, Robin J. Wilson, An Atlas of Graphs. Oxford Science
Publications,1998. ISBN: 0 19 853289

[5] Martin Dyer, Catherine Greenhill The complexity of counting graph homomorphisms ,Paper (Link)

[6] William Stein, Graph Theory in Sage ,Talk (Link)

[7] Rob Beezer, Chris Godsil Explorations in Algebraic Graph Theory with Sage:
, Practice Problems (Link)