Exercises on Weisfeiler–Leman
CSCI 6114 Fall 2021

Joshua A. Grochow

September 28, 2021

1. A partition $V = C_1 \cup \cdots \cup C_k$ is the vertices of a graph G is *equitable* if for each i, j, there is a number d_{ij} (depending only on i, j) such that if $v \in C_i$, then v has exactly d_{ij} neighbors in C_j.

   (a) Show that the trivial partition $C_1 = V$ is equitable iff G is regular.

   (b) Show that a partition is equitable iff the induced subgraph on each C_i is regular, and the induced bipartite graph in between C_i, C_j is biregular (meaning in that subgraph, the vertices in C_i all have the same degree, and the vertices in C_j all have the same degree, but those in C_i need not have the same degree as those in C_j).

2. *Color Refinement* is the following procedure. Given a simple undirected graph G, initially assign all vertices the same color. At each round, each vertex gets a new color, which will be a pair consisting of its old color, together with the multiset of colors of its neighbors. At each stage, the coloring gives a partition of $V(G)$ into its color classes. The process stops when the partition no longer changes, and the resulting coloring is called the *stable coloring*.

   (a) Characterize those graphs for which the initial coloring is the stable coloring.

   (b) How long does it take to compute a stable coloring?

   (c) Show that the partition of $V(G)$ given by a stable coloring is always an equitable partition.

   (d) Show that the stable coloring is always the *coarsest* equitable partition, that is, any other equitable partition refines that given by the stable coloring. In other words, if $V = C_1 \cup \cdots \cup C_k$ is an equitable partition, then under color refinement, for each i, every vertex in C_i receives the same color.

3. The 2-dimensional Weisfeiler–Leman algorithm for Graph Isomorphism (2-WL) works as follows. To decide isomorphism between two graphs G, H, it will assign colors to the pairs of vertices of G iteratively, and then use the result to (hopefully) make isomorphism testing easy.

   (a) Initially, the pair (v_1, v_2) is given one of three colors: red if $v_1 = v_2$, black if $(v_1, v_2) \in E(G)$, and white if $(v_1, v_2) \notin E(G)$.

   (b) At iteration $i+1$, a pair (v_1, v_2) gets a new color, which is itself a tuple of the form (c, M_1, M_2), where c is the previous color of (v_1, v_2) (at iteration i), M_1 is the multiset of colors (from iteration i) of the tuples of the form (u, v_2) (for all $u \in V(G)$), M_2 is the multiset of colors of the tuples of the form (v_1, u).

   (c) At each iteration, the coloring defines a partition of $V(G)^2$. The iteration stops when this partition does not change. This coloring is called the *stable coloring*.

   Each "color" is thus really a nested structure consisting of tuples of colors and multisets of tuples of colors and multisets of tuples of colors and multisets of... If the multiset of colors in the stable coloring of G and the stable coloring of H do not agree, then the algorithm reports that they are not isomorphic. In general, it is possible for the multi-set of colors to agree but still that $G \not\cong H$. We say a graph G is *identified by 2-WL* if for any graph H that is not isomorphic to G, we have that the stable colorings of G and H disagree.

   From each coloring of $V(G)^2$, we can get a coloring of $V(G)$ by assigning to each vertex v the color of the pair (v, v) in $V(G)^2$. Show that the coloring of $V(G)$ one gets this way from the stable coloring of $V(G)^2$ agrees with the stable coloring of $V(G)$ gotten by the color refinement procedure of the previous problem.

4. (a) An $n \times n$ permutation matrix is a matrix with 0-1 entries, with exactly one 1 in each row and exactly one 1 in each column. Show that permutation matrices are orthogonal, that is, if P is a permutation matrix, the $P^{-1} = P^T$.

   (b) For a graph G, let $A(G)$ denote its adjacency matrix, whose (i, j) entry is 1 iff $(i, j) \in E(G)$. Show that two graphs G, H

are isomorphic iff there is a permutation matrix P such that $PA(G) = A(H)P$.

(c) Given two graphs G, H, consider the following system of linear equalities and inequalities, with n^2 variables $P_{i,j}$:

$$\begin{aligned} PA(G) &= A(H)P \\ \sum_j P_{i,j} &= 1 \qquad \forall i \\ \sum_i P_{i,j} &= 1 \qquad \forall j \\ P_{i,j} &\geq 0 \qquad \forall i, j \end{aligned}$$

Show that P is an integer-valued solution to this system of (in)equations iff P is a permutation matrix corresponding to an isomorphism $G \to H$.

(d) Two graphs are called *fractionally isomorphic* if the above system of equations has a rational solution. Give an example of two graphs that are fractionally isomorphic but not isomorphic.

(e) Suppose P is a solution to the above equations. Let G_P denote a directed graph with adjacency matrix P. Show that the strongly connected components of G_P form an equitable partition of G.

(f) (*) Use the preceding to show that two graphs are fractionally isomorphic iff they are indistinguishable by color refinement.

(g) Using the preceding characterization, again give an example of two graphs that are fractionally isomorphic but not isomorphic.

Resources

- Arvind 2016 survey in *Bull. EATCS*
- Sandra Kiefer 2020 survey in *ACM SIGLOG News*
- Fractional isomorphism of graphs was introduced and studied in Ramana, Scheinermann, & Ullman, *Disc. Math.*, 1994; some of the authors wrote a whole book on fractional graph theory (easily found by searching). Turns out fractionally isomorphic graphs share many properties, even if they are not isomorphic.

- (Thanks to Michael Levet for suggesting this and the subsequent resources) Kiefer & McKay ICALP '20 (arXiv:2005.10182 [cs.DM]) show that the number of iterations of color refinement achieves the worst-case bound of $n - 1$.

- Cai, Fürer, Immerman for reading material on (i) how to get our hands on WL, (ii) connections to descriptive complexity (i.e., writing down succinct logical formulas), and (iii) WL (even higher-dimensional) does not solve GI in polynomial time. This is a landmark paper in the field.

- Grohe–Verbitsky, ICALP '06: WL can be efficiently parallelized, and this can yield nice complexity theoretic upper bounds for classes contained within P.