1 Graph Theory

On to the good stuff! Posets are but one kind of combinatorial structure. A somewhat more complicated structure is the *graph*, which describes relationships among a set of "nodes". In applciation, graphs describe a myriad of different relationships: graphs have been used to explore physical connection (with nodes being cities, relationships being roads), logical connection (with nodes being webpages and relationships being links), or social connection (with nodes being people and relationships being acquaintance or collaboration).

Mathematically, of course, we don't much care what the relationships are, and treat them completely abstractly.

Definition 1. A *graph* is an ordered pair (V, E), where V is a set of objects known as *vertices* or *nodes* and E is a set of unordered pairs of elements of V, which are called *edges*.

The elements of V could be anything: numbers, named constants, even combinatorial objects in their own right! However, under most circumstances, we do not give the vertices highly distinctive names, and it's conventional to refer to the elements of V as $\{v_1, v_2, \ldots, v_n\}$.

Definition 2. If $G = (X, Y)$, then the *vertex-set* X of graph G will be referred to as $V(G)$; in cases where only one graph is under consideration, this may be shortened to simply V; likewise the edge-set Y of graph G will be referred to as $E(G)$, and simply as E when such a reference is unambiguous.

Definition 3. A graph is *finite* if it has a finite number of vertices. The *order* of a finite graph is $|V(G)|$, also denoted $|G|$; the number of edges $|E(G)|$ in a finite graph G is denoted $\|G\|$.

Unless otherwise mentioned, where the term "graph" is used, it denotes "finite graph".

1.1 Edge-vocabulary

Definition 4. Two vertices of a graph G are called *adjacent* (or *adjacent in* G, if the same vertex-set is used for multiple graphs) if $[x, y] \in E(G)$. This may sometimes be denoted $x \sim y$.

Definition 5. An edge e of G is said to be *incident upon* a vertex v if $e = [v, x]$ for some x.

Definition 6. An edge e is said to be a *loop* if $e = [v, v]$ for some vertex v.

Definition 7. A graph G is *simple* if it has no loops.

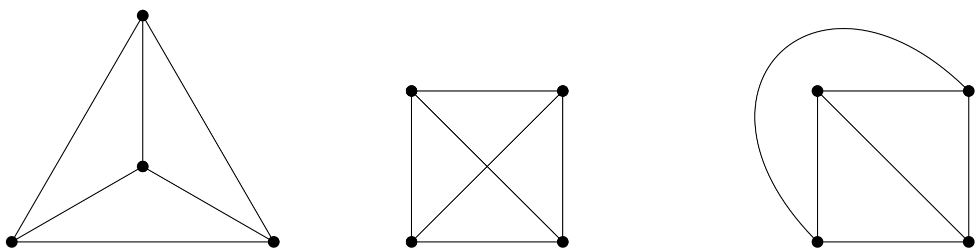
Unless otherwise mentioned, the term "graph" will henceforth denote simple graphs.

1.2 Visualization and Equivalence of Graphs

Graphs are subject to several different representations and visualizations. Graphs are frequently associated according to their essential properties: for instance, if Alex, Bob, Carl, and Diane are all friends, we might have the social conenction graph with vertex-set $\{\text{Alex}, \text{Bob}, \text{Carl}, \text{Diane}\}$ and edge-set $\{[\text{Alex}, \text{Bob}], [\text{Alex}, \text{Carl}], [\text{Alex}, \text{Diane}], [\text{Bob}, textCarl], [\text{Bob}, textDiane], [\text{Carl}, textDiane]\}$. If we were constructing a graph of Southwest Airlines' service in the Louisville environs, we might make our vertex-set the airport-codes $\{\text{SDF}, \text{BHM}, \text{STL}, \text{MDW}\}$, and the edges would be

{[SDF, BHM], [SDF, STL], [SDF, MDW], [BHM, MDW], [BHM, STL], [MDW, STL]}. Lastly, we could construct a nice abstract graph with vertex-set $\{v_1, v_2, v_3, v_4\}$ and edge-set $\{[v_1, v_2], [v_1, v_3], [v_1, v_4], [v_2, v_3], [v_2, v_4], [v$

These three graphs are manifestly different: they each have their own vertex-set and edge-set. And yet, they all describe the same fundamental scenario: a set of four nodes, with interconnections among every node. Likewise, the same structure could be conveyed in several visualizations:



We cut throught the varieties of graph presentation with the following definition.

Definition 8. Graphs G and H are *isomorphic* if there is a bijective map $\phi : V(G) \to V(H)$ such that $[a, b] \in E(G)$ if and only if $[\phi(a), \phi(b)] \in E(H)$.

Graph isomorphism is a relation, and can be pretty easily shown to be an equivalence relation. Thus, we may collect graphs into *isomorphism classes*: the above unlabeled pictures can be thought of as representative of their entire isomorphism class: that is, the set of *all* graphs with 4 vertices and all those vertices interconnected. When we speak of "a graph", we're more often than not referring to the entire isomorphism class, and thus not worrying overmuch about what the vertices are labeled.

Question 1: *How many different simple graphs are there on one vertex, on two vertices, or on three vertices?*

Answer 1: *If we interpret the word "graph" to describe a vertex set and edge-set, the answer is of course "infinitely many" for all of the above (since we could use any of an infinitude of different objects to serve as vertices), but if we interpret it as meaning "isomorphism classes of graphs", as we usually shall, the answers are more interesting.*

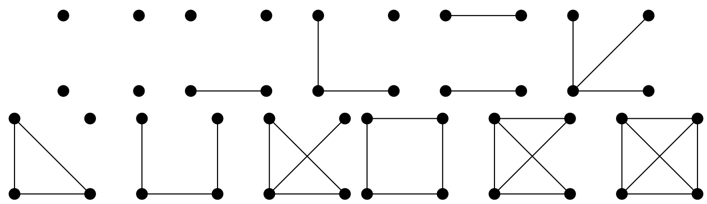
Consider a simple graph on one vertex, which we shall call v_1. *There are no edges one could put in this graph, so there's really only one graph possible: the graph with a single vertex and no edges.*

On two vertices v_1 *and* v_2, *we have the potential for an edge. We now have two distinct graphs on two vertices: the graph with no edges, and the graph with one edge.*

On three vertices v_1, v_2, *and* v_3, *we have three possible edges. Every graph with three vertices and one edge can be shown to be isomorphic under rearrangement of the vertices; likewise every graph with three vertices and two edges, or three vertices and three edges. Thus, there are three distinct graphs on three vertices.*

Question 2: *How many different simple graphs are there on four vertices?*

Answer 2: *Here we have to be a bit cleverer! Among four labeled vertices* v_1, v_2, v_3, v_4 *there are 6 potential edges, so we could lay down or not lay down edges in each of 6 positions, for a total of* $2^6 = 64$ *configurations. But this is massively overcounting: several of the configurations are in the same isomorphism class – for instance, all six of the graphs consisting of a single edge are isomorphic. If we did a brute-force enumeration of these graphs, we could produce a full classification:*



However, one needn't use brute force: this is in fact a brilliant place to use Burnside's Lemma!

Let us consider S to be the set of possible graphs — not isomorphism classes, but actual graphs — that can be built on the set of vertices $\{v_1, v_2, v_3, v_4\}$. There are $\binom{4}{2} = 6$ unordered pairs of distinct vertices, so there are 6 potential edges in a simple graph on these four vertices. Since each edge can be present or absent, each of these 6 potential edges can be chosen in one of two ways. Thus $|S| = 2^6 = 64$, which is far more than the number of isomorphism classes, since if two graphs can be mapped onto each other via permutation of the vertices, they are isomorphic. Thus, the isomorphism classes of the graphs are exactly the equivalence classes of the graphs under the permutation group S_4. So $|S/S_4|$ is the number of isomorphism classes of 4-vertex graphs.

The permutation group S_4 has $4! = 24$ elements, but we need not investigate each of them individually when applying Burnside's Lemma. The elements of S_4 can be classified according to their cycle-index: we have the identity permutation, which is unique; the single-swap, which can be constructed in any of $\binom{4}{2} = 6$ ways; the double swap, which can be constructed in any of $\frac{1}{2}\binom{4}{2}\binom{2}{2} = 3$ ways; the 3-cycle, which can be constructed in $2!\binom{4}{3} = 8$ ways; and the 4-cycle, which can be constructed in $3! = 6$ ways. Note that $1 + 6 + 3 + 8 + 6 = 24$, so we are satisfied that all permutations are accounted for.

The invariants under the identity are of course all of S, so $\text{Inv}(e) = 2^6 = 64$.

Considering a representative swap, for instance $(12)(3)(4)$, we see that edges v_1v_3 and v_2v_3 are swapped, as are v_1v_4 and v_2v_4, so two pairs of edges are swapped, and the other two edges are fixed. Thus, an invariant of $(12)(3)(4)$ is determined by choosing existence/nonexistence for each of four entities: the two fixed edges, and the two swapped pairs, so $\text{Inv}((12)(3)(4)) = 2^4 = 16$, and a similar result holds for every other swap.

Now let us consider a representative double-swap, e.g. $(12)(34)$. Edges v_1v_3 and v_2v_4 are swapped, as are v_1v_4 and v_2v_3, so two pairs of edges are swapped, and the other two edges are fixed. Thus, as above, there are four entities in question, so $\text{Inv}((12)(34)) = 2^4 = 16$.

Looking now at a three-cycle, for instance $(123)(4)$, we shall see that edges v_1v_2, v_2v_3, and v_3v_1 are cycled, as are v_1v_4, v_2v_4, and v_3v4, so in order to be invariant, existence/nonexistence must be chosen uniformly on each of these two cycles, so $\text{Inv}((123)(4)) = 2^2 = 4$

Likewise in the four-cycle (1234), edges v_1v_2, v_2v_3, v_3v_4, and v_4v_1 are cycled, while v_1v_3 and v_2v_4 are swapped, so in order to be invariant, existence/nonexistence must be chosen uniformly on each of these two cycles, so $\text{Inv}((123)(4)) = 2^2 = 4$.

Thus, using Burnside's Lemma:

$$|S/S_4| = \frac{2^6 + 6 \cdot 2^4 + 3 \cdot 2^4 + 8 \cdot 2^2 + 6 \cdot 2^2}{24} = 11$$

As an interesting curiosity, Pólya enumeration can also give us some insight here. A useful subquantification of the graphs with 4 vertices would be the number with each quantity of edges. So we can think of this much like our coloring of geometrical features, associating existence of an edge with the algebraic quantity x and nonexistence with 1. Then, building on the analysis seen above, we have a simple pattern enumeration:

$V(G)$ cycle index	Multiplicity	Representative π	Cycle index on $E(G)$	Inv(π)	Pattern Inventory
x_1^4	1	(1)(2)(3)(4)	x_1^6	2^6	$(1+x)^6$
$x_2x_1^2$	6	(12)(3)(4)	$x_1^2x_2^2$	2^4	$(1+x)^2(1+x^2)^2$
x_2^2	3	(12)(34)	$x_1^2x_2^2$	2^4	$(1+x)^2(1+x^2)^2$
x_1x_3	8	(123)(4)	x_3^2	2^2	$(1+x^3)^2$
x_4	6	(1234)	x_2x_4	2^2	$(1+x^2)(1+x^4)$

and Pólya enumeration would give the pattern inventory:

$$\frac{(1+x)^6+9(1+x)^2(1+x^2)^2+8(1+x^3)^2+6(1+x^2)(1+x^4)}{24} = 1+x+2x^2+3x^3+2x^4+x^5+x^6$$

which indeed classifies the 11 graphs of order 4 by number of edges.