Graph isomorphism/connectivity
Cardinality

Margaret M. Fleck

16 November 2009

This lecture finishes our coverage of basic graph concepts: isomorphism, paths, and connectivity. It shows the basic ideas without covering every possible permutation of these ideas e.g. for different types of graphs.

The second half is filled by a brief discussion of cardinality, an interesting topic but which doesn't have an obvious fixed place in the syllabus. It's covered at the very end of section 2.4 in Rosen.

1 Announcements

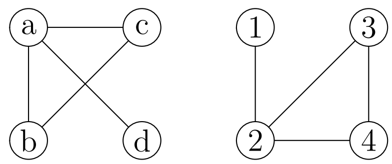
Discussion sections will happen this week, despite the strike. However, we are working with only part of the normal course staff. So if the strike doesn't finish quickly, I may need to make various adjustments to the course after break. The most likely is that the final homework might end up being self-graded.

2 Isomorphism

Two graphs are called isomorphic if they have the same structure, even though their vertices and edges might have been positioned differently in 3D or in a 2D picture. Let's look at how this idea is made formal.

Recall that a simple graph is an undirected graph with no self-loops (no edges from a vertex bad to itself) and no multi-edges (multiple edges between the same pair of vertices). Suppose that $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are simple graphs. Then G_1 and G_2 are isomorphic if there is a bijection $f : V_1 \to V_2$ such that vertices a and b are joined by an edge if and only if $f(a)$ and $f(b)$ are joined by an edge.

For example, consider the following two graphs:

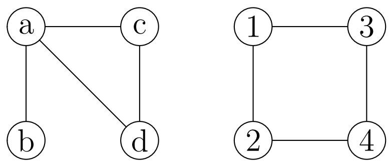


These two graphs are isomorphic. We can prove this by defining the function f so that it maps 1 to d, 2 to a, 3 to c, and 4 to b. The reader can then verify that edges exist in the left graph if and only if the corresponding edges exist in the right graph.

To prove that two graphs are not isomorphic, we could walk through all possible functions mapping the vertices of one to the vertices of the other. However, that's a huge number of functions for graphs of any interesting size. An exponential number, in fact. Instead, a better technique for many examples is to notice that a number of graph properties are "invariant," i.e. preserved by isomorphism.

- The two graphs must have the same number of vertices and the same number of edges.
- For any vertex degree k, the two graphs must have the same number of vertices of degree k.
- Any subgraph of the first graph must have a matching subgraph somewhere in the second graph. (We would normally choose a small subgraph.)

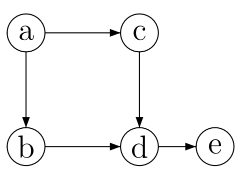
We can prove that two graphs are not isomorphic by exhibiting one example of a property that is supposed to be invariant but, in fact, differs between the two graphs. For example, in the following picture, the lefthand graph has a vertex of degree 3, but the righthand graph has no vertices of degree 3, so they can't be isomorphic.

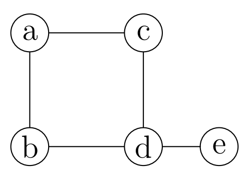


3 Paths

In an undirected graph, a path of length k from vertex a to vertex b is a sequence of edges that connect end-to-end $(v_1, v_2), (v_2, v_3), (v_3, v_4), \ldots, (v_{k-1}, v_k)$, where $v_1 = a$, $v_2 = b$. A single vertex a is considered to be a path of length zero from a to a. This definition works for a directed graph as well. Just beware that your path must follow the arrows on the edges.

For example, in the lefthand graph, there is a path from a to d: $(a, c), (c, d), (d, e)$. But there isn't a path from e back to a, because the arrows go the wrong way. In the righthand undirected graph, both paths exist.



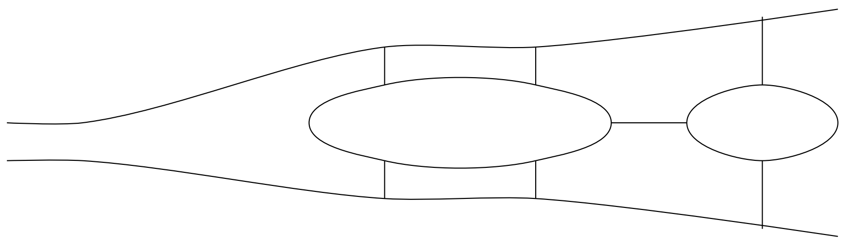


Three special types of paths are important to know about

- A circuit is a path that ends at the same node where it started.
- A path is simple if no edge occurs more than once in the path.

- An Euler circuit of a graph G is a simple circuit that contains every edge in the graph.

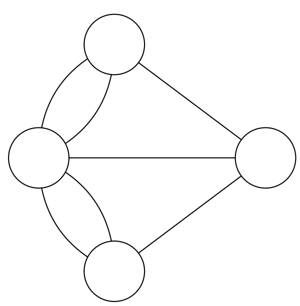
Fascination with Euler circuits dates back to the 18th century. At that time, the city of Königberg, in Prussia, had a set of bridges that looked roughly as follows:



Folks in the town wondered whether it was possible to take a walk in which you crossed each bridge exactly once, coming back to the same place you started. This is the kind of thing that starts long debates late at night in pubs, or keeps people amused during boring church services. Leonard Euler was the one who explained clearly why this isn't possible.

Specifically, an Euler circuit is possible exactly when each node has even degree. In order to complete the circuit, you have to leave each node that you enter. If the node has odd degree, you will eventually enter a node but have no unused edge to go out on.

For our specific example, the corresponding graph looks as follows. Since all of the nodes have odd degree, there's no possibility of an Euler circuit.

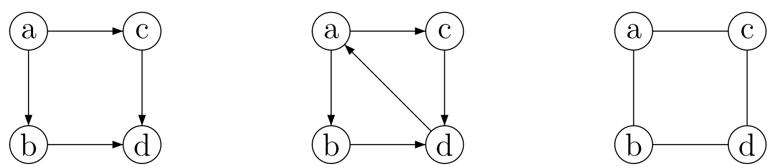


4 Connectivity

An undirected graph G is connected if there is a path between every pair of nodes in G. That is, for any nodes a and b in G, there is a path from a to b. If G is directed, we have two notions of connectivity:

- G is strongly connected if there is a (directed) path between any pair of vertices in G.
- G is weakly connected if its underlying undirected path (i.e. remove the directions from all the edges) is connected.

If a and b are two nodes in G, strong connectivity requires that there be a path from a to b and also a path from b to a. So the lefthand graph in the following picture is not connected, because there's no way to get from d to a. This graph is weakly connected, because its underlying undirected graph (on the right in the figure) is connected. The directed graph in the middle, with an extra edge from d to a, is strongly connected.



5 The rationals and the reals

You're familiar with three basic sets of numbers: the integers, the rationals, and the reals. The integers are obviously discrete, in that there's a big gap between successive pairs of integers.

To a first approximation, the rational numbers and the real numbers seem pretty similar. The rationals are dense in the reals: if I pick any real number x and a distance δ, there is always a rational number within distance δ of x. Between any two real numbers, there is always a rational number.

We know that the reals and the rationals are different sets, because we've shown that a few special numbers are not rational, e.g. π and $\sqrt{2}$. However, these irrational numbers seem like isolated cases. In fact, this intuition is entirely wrong: the vast majority of real numbers are irrational and the rationals are quite a small subset of the reals.

6 Completeness

One big difference between the two sets is that the reals have a so-called "completeness" property. It states that any subset of the reals with an upper bound has a smallest upper bound. (And similarly for lower bounds.) So if I have a sequence of reals that converges, the limit it converges to is also a real number. This isn't true for the rationals. We can make a series of rational numbers that converge π (for example) such as

$$3, 3.1, 3.14, 3.141, 3.1415, 3.14159, 3.141592, 3.1415926, 3.14159265$$

But there is no rational number equal to π.

In fact, the reals are set up precisely to make completeness work. One way to construct the reals is to construct all convergent sequences of rationals and add new points to represent the limits of these sequences. Most of the machinery of calculus depends on the existence of these extra points.

7 Cardinality

Furthermore, although the rationals and the reals both contain infinitely many points, we can show that the reals have "more" points. To do this, we need to definite what it means for two sets to have the same cardinality, i.e. the same mathematical size.

> Definition: Two sets A and B have the same cardinality if and only if there is a bijection from A to B.

Finite sets have the same cardinality exactly when they have the same number of elements in the usual sense. But not all infinite sets have the same cardinality.

> An infinite set A is *countable* or *countably infinite* if there is a bijection from $\mathbb{Z}^+$ to A.

The full set of integers is countable, because we can map the natural numbers onto the integers using the function f where $f(n) = \frac{n}{2}$ when n is even and $f(n) = \frac{-n+1}{2}$ when n is odd.

The positive rationals are countable because we can put them into an ordered list. [show picture, which is on Rosen p. 159]. Remember that we saw a formula for such a function in homework 4. It's not hard to extend this construction to include the negative rationals.

However, we can show that the reals are not countable. Specifically, we'll show that there's no bijection from the positive integers to the real interval $[0, 1]$ using a construction called "diagonalization" developed by Georg Cantor.

If the numbers in $[0, 1]$ were countable, we could put them into a list a_1, a_2, and so forth. Let's write out a table of the decimal expansions of the numbers on this list. [see picture p. 160 of Rosen.] Now, examine the digits along the diagonal of this table: a_{11}, a_{22}, etc. Suppose we construct a new number b whose kth digit b_k is 4 when a_{kk} is 5, and 5 otherwise. Then b won't match any of the numbers in our table, so our table wasn't a complete list of all the numbers in $[0, 1]$.

So, the reals are larger than the integers.

8 Uncomputability

A practical consequence of this difference in size is that there are mathematical functions that can't be computed by any program. First, consider the functions $f : \mathbb{N} \to D$ (where D is the decimal digits). Each function corresponds to the decimal expansion of some real number in $[0, 1]$. So even this limited set of functions is uncountable.

However, suppose we fix an alphabet for writing our programs (e.g. 8-bit ASCII). Since each individual program is finite in length, we can put all possible programs into a (very long) ordered list. For any fixed character length k, there are only a finite set of possible programs. So, we can write down all programs by first first writing down all the 1-character programs, then all the 2-character programs, and so forth. In other words, there's a bijection between the integers and the total set of programs.

But this means that the number of functions is uncountable, whereas the number of programs is only countably infinite. So there must be mathematical functions that we can't compute with any (finite-length) program.