UPPSALA UNIVERSITET

Föreläsningstackneningar

Grafteori

Rami Abou Zahra

Inlämningsdatum
November 3, 2022

Contents

1. TODO 2
2. Bridges of Köningsberg 3
2.1. Vocabulary 3
3. Simple graphs 8
3.1. Special graphs 10

1. TODO

- Def 3.4, does φ or φ' need to be surjective?

2. Bridges of Köningsberg

This was the birth of graphtheory. The idea here is that the precise location of where the person is does not matter, only the placement of the bridges and mainland. Therefore, we can encode the position by an abstract point (*vertex*) and connect these to *edges* to represent bridges.

2.1. **Vocabulary.**

We therefore obtain the follwing:

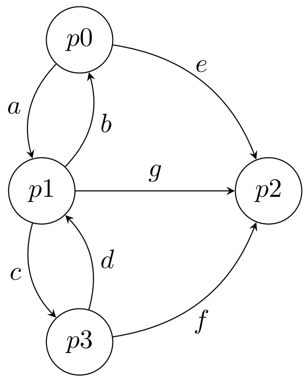

Figure 1.

Definition/Sats 2.1: Multigraph

A *multigraph* G is a tripple $G = (V, E, \iota)$ consisting of:

- A set V of vertices
- A set E of edges
- $\iota : E \to \{A \subseteq V \mid |A| = 1 \text{ or } |A| = 2\}$

Example:
$\iota(c) = \{2, 3\} = \iota(d)$
$\iota(e) = \{1, 4\}$

Anmärkning:
Notice that the graphical view (and the placement of the vertices) is not reflected in the tripple, therefore we can draw the same graph in a completely different manner.

Loops:
This is what happens when $|A| = 1$:

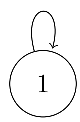

Figure 2.

Parallell edges:
$\iota(e) = \iota(e')$

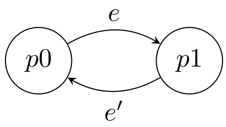

FIGURE 3.

Neighbours/adjacent:

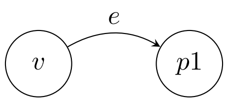

FIGURE 4. v is *incident* to e and a neighbour to w

Anmärkning:
A loop means the vertex belongs to its own Neighbourhood.

Definition/Sats 2.2: Finite graph

We say that a graph G is finite if we have:

$$|V| + |E| < \infty$$

Definition/Sats 2.3: Walk

Let $G = (V, E, \iota)$ be a graph
A *walk* of length k is a sequence $v_0e_1v_1e_2v_2 \cdots e_kv_k$ where as the notation suggests, $e_1, \cdots, e_k$ are edges and $v_0, \cdots, v_k$ are vertices such that $\iota(e_i) = \{v_{i-1}, v_i\}$ for $i = 1, \cdots, k$

Definition/Sats 2.4: Trail

A *trail* is a walk that uses no edges twice. This is something we want in the Bridges of Köningsberg

Definition/Sats 2.5: Path

A *path* is a walk that uses no vertex twice.

Definition/Sats 2.6: Circuit

A *circuit* is a trail where first and last vertex coincide.
Meaning I start somewhere, dont repeat edges, and return at start place

Definition/Sats 2.7: Cycle

A *cycle* is a circuit where the first and last vertices are the only vertices coinciding

Example:
Using the bridges, an example of a trail and a circuit, but not a cycle because vertex 3 is visited twice $1a3g4f2c3b1$
An example of a cycle would be $1a3b1$

Anmärkning:
Every path is a trail.
Every cycle is a circuit.

Definition/Sats 2.8: Eulerian trails

A trail is called *Eulerian* if it uses every edge in the graph

Definition/Sats 2.9: Eulerian circuits

A circuit using every edge is called an *Eulerian circuit*

Anmärkning:
If a graph admits an Eulerian circuit, then the graph is called *simply Eulerian*

Definition/Sats 2.10: Connected vertex

Let $G = (V, E, \iota)$ be a graph
We say that a vertex $v \in V$ is *connected* to a vertex $w \in V$ if there exists walk (or equivalently a trail/path) starting in v and ending in w

If v is connected to w for all $v, w \in V$, then the graph G is *connected*

What we are saying here is that we call vertices that we can walk to connected.

Anmärkning:
v is conncted to v (every vertex is connected to itself)
Moreover, if v is connected to w, then w is connected to v.
v is connected to w and w connected to z, then v is connected to z.

Anmärkning:
Connection is an equivalence relation.

Definition/Sats 2.11: Connected components

Equivelance classes of the equivalence relation are called *connected components.*

Definition/Sats 2.12: Degree of vertex

Let $G = (V, E, \iota)$ be a graph and $v \in V$. The *degree* of v is $\deg(v)$ and is the number of half-edges incident to v.

The reason we do half-edges is because we want loops to count twice (once for exit, and once on entry)

Definition/Sats 2.13: Euler; 1736

A finite connected graph is Eulerian iff all its vertex degrees are even.

Bevis 2.1

In $\Rightarrow$ direction. Any vertex on the circuit needs to have even degree because you need a half-edge to go into the vertex and another one to go out.
Since it is connected, if I visit every vertex I also visit every edge and these come in pairs

In $\Leftarrow$ direction. Assume $G = (V, E\iota)$ is finite, connected, and only has even degrees.
Assume G as no loops (convenience). We dont know if we can build an Eulerian circuit or even if we have a circuit, but we know that there is a trail (since it is connected)
Therefore, consider a trail $J = v_0 e_1 v_1 \cdots e_k v_k$.
Since the graph is finite, then there is a maximum trail, suppose J is a maximum trail (implying max length k). Then we cant possibly extend it, so any edge we see at k must already be on the trail.
What we want to show is that $v_0 = v_k$ because of this. Then we actually have a circuit.
Therefore, assume there are $2s \quad (s \in \mathbb{N})$ edges incident to v_k. We know there is an even number of edges (because we excluded loops).
If we look at our trail $v_0 e_1 v_1 \cdots v_{i-1} e_i \underbrace{v_i}_{v_k} e_{i+1} v_{i+1}$
Then e_i and e_{i+1} are incident to $v_k = v_i$, but so is v_k. But v_k only has one edge, therefore e_1 has to be incident to $v_k = v_0$
We have now shown we have a trail, we show it is Eulerian.
Assume for a contradiction that it is not Eulerian. This means that there are parts not in our trail.
There is $e \in E$ with endpoints $\iota(e) = \{v, w\}$ s.t e is not on J but one of v, w is.
WLOG v is on J. Say $v = v_j$ for some j.
Consider $wev_j e_{j+1} \cdots e_k \underbrace{v_k}_{v_0} e_1 v_1 e_2 v_2 \cdots e_j v_j$, we claim that this is a trail. Notice here that we have length $k + 1$, which is longer than k. Contradiction. $\square$

Anmärkning:
A useful proof-tool in graphtheory is setting up a situation where we fix a maxlength and argue the contrary.

Anmärkning:
Notice how $\Rightarrow$ was "obvious", we call this *TONCAS* - The Obvious Necessary Conditions Are Sufficient

Anmärkning:
If we have loops, we can simply traverse these loops and add them to our trail. This will not affect the proof.

Corollary:
A finite connected graph admits an Eulerian trail iff either 0 or 2 of its vertex degrees are odd

We can show this by retracing this back to the previous theorem. If we have 0 odd degrees, then the theorem holds.
If we have 2 vertices of odd degree, then we can draw an additional edge between v, w. This means that both of the vertices that had odd degrees have gotten their degrees bumped up by one, so they know have even degree, which implies the theorem (is an Eulerian circuit), so it visits all the edges (and especially the new edge). Then we can remove the new edge from the Eulerian circuit, which gives an Eulerian trail in the original graph.

If we look at the statement of the corollary, it leaves a graph. What happens if it has 1 odd vertex degree? We are gonna show that this is impossible.

Definition/Sats 2.14: Handshake lemma

Let $G = (V, E, \iota)$ be a finite graph.
Then

$$2\,|E| = \sum_{v \in V} \deg(v)$$

In particular, G has even number of vertices of odd degree. (odd+odd = even, even + even = even)

Bevis 2.2: Handshake lemma

We use a trick from combinatorics (double counting). We identify a quantity and count it in 2 different ways.

We double count half-edges. Every edge gives 2 half-edges, so we $2\,|E|$ half-edges. On the other hand, every vertex gives $\deg(v)$ half-edges $\Rightarrow \sum_{v \in V} \deg(V)$ half-edges.
It does not matter how I count them, therefore these quantities have to be the same. $\square$

Anmärkning:
We can also use induction to show the Handshake lemma.
Start with 0 edges on V, which implies all the degrees are 0. Then add edges 1 by 1. And whenever you add an edge, the RHS increases by 2.

What happens if we have 4 vertices of odd degree?
We can partition $E = E_1 \cup E_2$ such that E_1 is a edge set of a trail and so is E_2 (but $E_1 \cap E_2 = \emptyset$)

3. Simple graphs

The idea of simple graphs is to forbid parallell edges and loops. Here, we dont care how things are connected, but which things that *are*.

For example, we can encode the game *Towers of Hanoi* as a simple graph by letting n disks be stacked on $3 pegs$
We can therefore encode a game state by a string of length n

Definition/Sats 3.1: Simple graph

A *simple graph* is a multigraph without parallell edges or loops
An equivalent definition, it is a pair $G = (V, E)$ where $E \subseteq \mathcal{P}_2(V)$

By $\mathcal{P}_2$ we mean the powersets of size 2:

$$\mathcal{P}_2(V) = \{A \subseteq V \mid |A| = 2\}$$

Anmärkning:
Our ι is gone! This is because by not having parallell edges and loops, then $\iota : E \to \mathcal{P}_2(V)$ is injective and we can identify the output of ι with its input, and that is what the definition of a simple graph is

Anmärkning:
Every graph is a multigraph

Lemma 3.1

Any simple graph on n vertices has at most $\binom{n}{2}$ edges

Bevis 3.1

The edge set $E \subseteq \mathcal{P}_2(V)$ and $|\mathcal{P}_2(V)| = \binom{n}{2}$ □

Anmärkning:
This implies that simple graphs are finite. In multigraphs, we could put arbitrary edges between vertices, but here it is not accepted.

Our vertex set is arbitrary, it doesnt matter if $V = \{1, 2, 3, 4\}$ or $V = \{a, b, c, d\}$, we need to set up a notion of "sameness" in graphs taking into account that the vertex set is arbitrary.

Definition/Sats 3.2: Labelled graph

A *labelled graph* is a graph with a fixed vertex set, commonly $V = \{1, 2, \cdots, n\}$ if V is finite.

Lemma 3.2

There are $2^{\binom{n}{2}}$ labelled graphs on n vertices.

Bevis 3.2

Since $V = \{1, 2, \cdots, n\}$ is fixed, two graphs (V, E) and (V, E') coincide iff $E = E'$

Conversely, any subset of $\mathcal{P}_2(V)$ defines an edge set. We are essentially looking for $\mathcal{P}(\mathcal{P}_2(V))$, and the cardinality of this is $2^{|\mathcal{P}_2(V)|} = 2^{\binom{n}{2}}$ □

Definition/Sats 3.3: Morphism

Let $G = (V, E)$ and $G' = (V', E')$ be simple graphs.

A *morphism*

$$\varphi : G \to G'$$

is a map

$$\varphi : V \to V'$$

such that $\{v, w\} \in E \Rightarrow \{\varphi(v), \varphi(w)\} \in E'$

Example:
See example 15

Anmärkning:
Graph-morphisms do not need to be injective/surjective, nor do they need to exist

Graph-morphisms preserve edges between graphs, thats their whole point

Definition/Sats 3.4: Identity morphism

For every simple graph G, theere is an identity morphism $id_G : G \to G$ where $id_G : V \to V$ is the identity map

For simple graphs G, G', G'' and morphisms

$$\varphi : G \to G'$$
$$\varphi' : G' \to G''$$

There is a morphisms $\varphi' \circ \varphi : G \to G''$, given by the map $\varphi' \circ \varphi : V \to V''$

Definition/Sats 3.5: Isomorphism

Two graphs G, G' are *isomorphic* if there is a bijective morphism $\varphi : V \to V'$ and $\{v, w\} \in E \Leftrightarrow \{\varphi(v), \varphi(w)\} \in E'$

Another way of saying this there is $\varphi : G \to G'$ and a $\psi : G' \to G$ such that $\varphi \circ \psi = id_{G'}$ and $\psi \circ \varphi = id_G$

Anmärkning:
Isomorphic graphs are not necessarily the same if they are labelled. We need to make sure the degree of each vertice coincide, and that we dont lose any edges.

Definition/Sats 3.6

The number g_n of non-isomorphic simple graphs on n vertices satisfies the following:

- $g_n = \frac{2^{\binom{n}{2}}}{n!}\left(1 + \frac{n^2 - n}{2^{n-1}} + \frac{8n!}{(n-4)!} \cdot \frac{(3n-7)(3n-9)}{2^{2n}} + \mathcal{O}\left(\frac{n^5}{2^{5n/2}}\right)\right)$

In particular, g_n behaves asymptotically as $2^{\binom{n}{2}}/n!$ in the same way that the probability distribution Hyp becomes Bin for large populations. When we make lots of graphs, eventually, the number of graphs that are isomorphic are so small they dont matter in the grand scheme.

3.1. **Special graphs.**

Some graphs are so special that they are given special names:

- The complete graphs on n vertices, denoted by K_n. All $\binom{n}{2}$ edges are present (every vertex is a neighbour of everything else)
- The path graph of length l, denoted by P_l is just a regular path as a graph
- The cycle graph on n vertices, denoted by C_n $(n \geq 3)$
- The complete bipartite graphs, denoted $K_{a,b}$. Here, V is partitioned as the disjoint union $V = V_a \cup V_b$. This means $|V| = a + b$. There are no edges between two vertices in the same set, but all possible edges are between the two sets.
  Notice that $K_{a,b} \cong K_{b,a}$
- The complete r-partite graphs $K_{a_1,\cdots,a_r}$ has a vertex set $V = \bigcup_{i=1}^{r} V_{a_i}$ such that $|V_{a_i}| = a_i$
  We say that two vertices are neighbours iff they are in different sets.

Lemma 3.3

The complete r-partite graph $K_{a_1,\cdots,a_r}$ on n vertices (sum of all $a_i = n$) has

$$|E| = \frac{1}{2}(n^2 - a_1^2 - \cdots - a_r^2)$$

Bevis 3.3

A vertex in set V_{a_i} has $n - a_i$ neighbours
By the Handshake lemma, $2\,|E| = \sum_{v\in V} \deg(v) = \sum_{i=1}^{r} a_i(n - a_i) = n\sum$

$$2\,|E| = \sum_{v\in V} \deg(v) = \sum_{i=1}^{r} a_i(n-a_i) = n\underbrace{\sum_{i=1}^{r} a_i}_{=n} - \sum_{i=1}^{r} a_i^2$$

$$n^2 - a_1^2 - \cdots - a_r^2$$

$\square$

Anmärkning:
$K_{a,b}$ has $\frac{1}{2}(n^2 - a^2 - b^2) = ab$ edges and $K_n = K_{1,\cdots,1}$ has $\frac{1}{2}(n^2 - n) = \binom{n}{2}$ edges

Definition/Sats 3.7: Subgraph

Let $G = (V, E)$ be a simple graph.
A simple graph $H = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$

Definition/Sats 3.8: Induced subgraph

An *induced* subgraph, is a subgraph $H = (V', E')$ of G, such that $E' = \{\{x, y\} \in E \mid x, y, \in V'\}$

Denoted by $H = G[V']$

Definition/Sats 3.9: Edge-induced subgraph

An *edge-induced* subgraph is a subgraph $H = (V', E')$ such that $V' = \{v \in V \mid v \text{ is incident to some } e \in E'\}$

Denoted by $H = G < E' >$

Definition/Sats 3.10: Spanning subgraph

A subgraph $H = (V', E')$ of G is a *spanning* subgraph if $V' = V$

Anmärkning:
There is a way to extend this into multigraphs, but you need to find a way to take care of ι