Math 4710/6710 – Graph Theory – Fall 2019

Basic concepts I

* indicates a definition different from the one in the book.

Graphs

Graph G consists of vertex set $V(G)$, edge set $E(G)$, incidence relation ψ_G mapping each edge to unordered pair of vertices.
Convention: $n = |V(G)|$, $m = |E(G)|$.
If $\psi_G(e) = uv$ say e *incident* with u, v.
Parallel edges are incident with same vertices. *Loop* is incident with same vertex twice.
Adjacent vertices u, v: some edge with $\psi_G(e) = uv$. Write $u \sim v$. (Can have $u \sim u$ if loop at u.)
Neighbor of v is u with $u \neq v$, $u \sim v$. (A vertex is never its own neighbor.) $N_G(v)$ or $N(v)$ is set of neighbors of v.
Null graph: $V(G) = E(G) = \emptyset$. Usually assume all graphs nonnull and finite.
Loopless graph has no loops. *Simple* graph has neither loops nor parallel edges.
* *Standard (model) simple graph* has $E(G) \subseteq \binom{V(G)}{2}$, ψ_G defined implicitly as identity map. Usually assume all simple graphs are standard model.
Underlying (standard) simple graph: delete loops, reduce parallel edges to single edge.
Complement $\overline{G}$ of standard simple graph G: $V(\overline{G}) = V(G)$, $E(\overline{G}) = \binom{V(G)}{2} - E(G)$. Note that $\overline{\overline{G}} = G$.

Matrices

Incidence matrix M_G: index rows by vertices, columns by edges, ve-entry is how many times e is incident with v (2 for a loop).
Adjacency matrix A_G: index both rows and columns by vertices. For $u \neq v$, uv-entry is number of edges incident with both u and v. For $u = v$, uu-entry is TWICE number of loops incident with u.

Degrees

Degree $d_G(v)$ or $d(v)$: number of ends of edges incident with v. (Loops count twice. Degree is sum of v's row in M_G, and sum of v's row or v's column in A_G.)
k-regular: all degrees are k. *Cubic* = 3-regular.
Maximum degree is $\Delta(G)$, *minimum degree* is $\delta(G)$, *average degree* is $d(G)$.
Degree sequence of G: list of all degrees, in no particular order. (Technically, multiset of degrees. Often convenient to list in ascending or descending order.)
Degree-Sum Formula: $\sum_{v \in V(G)} d_G(v) = 2|E(G)| = 2m$. *Proof:* Count ends of edges in two ways. ∎ (Hence sum of degrees must be even.)
Corollary: $d(G) = 2m/n$.
Complement degrees: $d_{\overline{G}}(v) = n - 1 - d_G(v)$.

Isomorphisms and automorphisms

Identical graphs G and H have $V(G) = V(H)$, $E(G) = E(H)$, $\psi_G = \psi_H$.
Isomorphism from G to H is pair (θ, φ) where $\theta : V(G) \to V(H)$ is a bijection, $\varphi : E(G) \to E(H)$ is a bijection, and incidence is preserved: $\psi_G(e) = uv \Leftrightarrow \psi_H(\varphi(e)) = \theta(u)\theta(v)$ for all $e \in E(G)$. If isomorphism exists say G and H are *isomorphic*, write $G \cong H$.
For standard simple graphs G and H, just give θ. If $u \sim v$ in $G \Leftrightarrow \theta(u) \sim \theta(v)$ in H for all $u, v \in V(G)$ then we have isomorphism; φ is given by natural extension of θ to pairs of vertices.
For finite graphs, if know $|E(G)| = |E(H)|$ and θ, φ are bijections, enough to check one direction, $\psi_G(e) = uv \Rightarrow \psi_H(\varphi(e)) = \theta(u)\theta(v)$.
Properties of isomorphisms:

(1) $(\mathrm{id}_{V(G)}, \mathrm{id}_{E(G)})$ is an isomorphism of G to itself.
(2) If (θ, φ) is an isomorphism from G to H, then $(\theta^{-1}, \varphi^{-1})$ is an isomorphism from H to G.
(3) If (θ_1, φ_1) is an isomorphism from G to H, and (θ_2, φ_2) is an isomorphism from H to K, then $(\theta_2 \circ \theta_1, \varphi_2 \circ \varphi_1)$ is an isomorphism froM G to K.

For any well-defined class of graphs, (1), (2) and (3) show that isomorphism is reflexive, symmetric, and transitive. Hence it is an equivalence relation.

Labelled graph = graph, as previously defined. *Unlabelled* graph represents equivalence class of graphs, draw picture of graph with no labels on vertices or edges.

Invariant of graphs is property that is same for all isomorphic graphs.

Automorphism of G is isomorphism of G to itself. Properties (1), (2), (3) above, together with associativity of composition, show that set of all automorphisms of G, $\operatorname{Aut} G$, is a group under composition.

Asymmetric graph: only automorphism is identity.

Similar vertices u and v: there is an automorphism mapping u to v; equivalence relation. *Vertex-transitive* graph: all vertices are similar.

Common invariant properties

* *Coboundary* $\delta_G X$ or δX of $X \subseteq V(G)$: set of edges of G with one end in X, other end in $V(G) - X$.

Disconnected if can partition $V(G)$ into two nonempty sets X, Y so no edge from X to Y. Equivalently, there is $X \neq \emptyset, V(G)$ with $\delta X = \emptyset$. *Connected* if not disconnected. (Equivalent condition in terms of paths, later.)

Planar graph: can be drawn in plane without edge crossings. *Embeddable in surface* Σ if can be drawn in Σ without edge crossings.

Cycles and paths

* *Cycle* C_n = connected n-vertex 2-regular (unlabelled) graph.

* *Path* $P_n = C_n$ with one edge deleted.

Independent sets and cliques

Independent or *stable* set $S \subseteq V(G)$: no loops on S and vertices of S are pairwise nonadjacent. *Independence number* $\alpha(G)$ = maximum number of vertices in an independent set.

Clique $S \subseteq V(G)$: any two distinct vertices of S are adjacent. *Clique number* $\omega(G)$ = maximum number of vertices in a clique. (*Clique* also used to refer to subgraph.)

For simple graphs $\alpha(\overline{G}) = \omega(G)$, $\omega(\overline{G}) = \alpha(G)$.

Complete graph K_n: simple n-vertex, any two distinct vertices are adjacent. ($E(K_n) = \binom{V(K_n)}{2}$), whole vertex set is clique.)

Trivial graph is K_1.

* *Supercomplete graph* is not necessarily simple, but any two distinct vertices are adjacent.

Empty or * *edgeless graph* $\overline{K_n}$: n vertices, no edges.

Bipartite graph: can partition $V(G)$ into X, Y (may be empty) so that every edge has one end in X, other end in Y. Write $G = G[X, Y]$. (So X, Y are independent.)

Complete bipartite graph $K_{m,n}$: simple $G[X, Y]$, $|X| = m$, $|Y| = n$, every $x \in X$ adjacent to every $y \in Y$.

k-partite graph: can partition $V(G)$ into $V_1, V_2, \ldots, V_k$ (sets may be empty) so that each V_i is independent.

Complete k-partite graph $K_{n_1,n_2,\ldots,n_k}$: k-partite simple graph with $|V_1| = n_1$, $|V_2| = n_2$, etc., any two vertices adjacent unless in the same V_i.

Subgraphs

Subgraph F of G, $F \subseteq G$: F is a graph, $V(F) \subseteq V(G)$, $E(F) \subseteq E(G)$, and $\psi_F = \psi_G|_{E(F)}$.

Proper subgraph, $F \subset G$: $F \subseteq G$, $F \neq G$.

$G - v$: delete vertex v and all incident edges. Repeat: $G - S$, $S \subset V(G)$.

$G - e$: delete edge e (do not delete any vertices). Repeat: $G - T$, $T \subseteq E(G)$.

G/e (*G contract e*): delete e, identify ends of e if distinct. Repeat: G/T, $T \subseteq E(G)$.

$G/\{u, v\}$: identify vertices u and v.

$G + e$: add edge e with known incidences. Repeat: $G + T$, T set of edges.

Disjoint subgraphs: no common vertices. *Edge-disjoint* subgraphs: no common edges.

* *Component* is a maximal connected subgraph (maximal under subgraph ordering).

Spanning subgraph F of G: $V(F) = V(G)$. Spanning path or cycle usually called *hamilton* path or cycle. Spanning k-regular subgraph usually called *k-factor*.

Induced subgraph $G[S]$ for $S \subseteq V(G)$: vertex set is S, all edges of G with both ends in S.
Edge-induced subgraph $G[T]$ for $T \subseteq E(G)$: edge set is T, vertex set is all ends of all edges in T.

Graphs from other graphs

Line graph $L(G)$ of simple graph G: $V(L(G)) = E(G)$, make e and f adjacent in $L(G)$ if they are incident with a common vertex in G.
* *Union and intersection* of graphs G, H can be defined if they are *consistent*: incidence function agrees for any edges in $E(G) \cap E(H)$. Take union/intersection of vertex and edge sets, and inherit incidence from G or H (or both).
Cartesian product $G \square H$ of simple G, H:
$V(G \square H) = V(G) \times V(H) = \{(u, v) \mid u \in V(G), v \in V(H)\}$;
$E(G \square H) = \{(u_1, v)(u_2, v) \mid u_1u_2 \in E(G), v \in V(H)\} \cup \{(u, v_1)(u, v_2) \mid u \in V(G), v_1v_2 \in E(H)\}$
(Sometimes also denoted $G \times H$.)
$m \times n$ *grid* $= P_m \square P_n$.
Join $G \vee H$ of disjoint G, H: join every vertex of G to every vertex of H by one edge. (Sometimes also denoted $G + H$.) If simple, $\overline{G \vee H} = \overline{G} \cup \overline{H}$.

Moving around

Walk in G: alternating sequence of vertices and edges $W = v_0e_1v_1e_2v_2 \ldots v_{\ell-1}e_\ell v_\ell$ where $\psi_G(e_i) = v_{i-1}v_i$ for each i. (In simple graph can just write $W = v_0v_1v_2 \ldots v_{\ell-1}v_\ell$.) *Length* is ℓ. *Initial vertex* v_0, *terminal* or * *final vertex* v_ℓ, *ends* v_0 and v_ℓ, *internal vertices* $v_1, v_2, \ldots, v_{\ell-1}$.
Reverse of walk W: $W^{-1} = v_\ell e_\ell v_{\ell-1} \ldots v_1e_1v_0$.
uv-walk has initial vertex u, final vertex v.
Closed walk has initial vertex = final vertex.
Trail is walk with no repeated edges.
* *Path* is walk with no repeated vertices. (So defines subgraph that is path graph.)
* *Cycle* is closed walk with no repeated vertices except that initial vertex = final vertex, no repeated edges, and at least one edge. (So defines subgraph that is cycle graph.) In simple graph write $(v_0v_1v_2 \ldots v_{\ell-1})$.
* *Reachability relation* R_G: uR_Gv if there is a uv-walk in G. $R_G(u) = \{v \in V(G) \mid uR_Gv\}$. This is an equivalence relation.

The *distance* from u to v in G, $d_G(u, v)$, is the length of a shortest uv-path.
* *Euler trail* = trail using all edges **and vertices** of G,
Euler tour = closed euler trail,

Trees

acyclic graph or *forest*: no cycles;
tree: (nonnull) connected forest;
leaf: degree 1 vertex.
cutedge e: $G - e$ has more components than G;
cutvertex v: $G - v$ has more components than G.

Lemma: A nontrivial tree has at least two leaves.

Equivalent characterizations of a tree:
(i) connected and acyclic (definition!);
(ii) connected, $m = n - 1$;
(iii) acyclic, $m = n - 1$;
(iv) connected, every edge is a cutedge;
(v) loopless, unique uv-path for all vertices u, v.

Rooted trees: *r-tree* or *tree rooted at r* is tree with special designated vertex r, the *root*. In an r-tree T there is a unique rv-path rTv.
ancestor of v: any vertex of rTv (inc. v)
parent $p(v)$: immediate predecessor on rTv (root has no parent)
proper ancestor (not v itself), *descendant*, *related*

level of v: $\ell(v) = d_T(r, v)$

* **Global Tree Construction Method (Global TCM):** Start with edgeless spanning subgraph F. At each step choose an edge not forming a cycle (equivalently, joining two distinct components) with F and add it to F. When we cannot continue F is a spanning tree.
* **Local Tree Construction Method (Local TCM):** Choose particular vertex r. Apply Global TCM, at each step adding an edge leaving the component containing r.
* **Breadth First Search (BFS):** Use Local TCM, adding uv with $u \in V(T)$, $v \notin V(T)$, where u was added to T as early as possible.