Chapter 7

Graphs

7.1 Directed and Undirected Graphs

Theory of graphs is a modern branch of mathematics. The first problem which is believed to lie in the foundation of it is the problem of seven bridges in Königsberg in Prussia (now Kaliningrad in Russia) over the river Pregel. There are two islands in the city. One island is connected by two bridges with each riverside, and by one bridge to the other island, moreover there is one bridge from the second island to each riverside. The problem was to find a walk through the city that would cross each bridge exactly once, and "swimming is forbidden", which means that once an island / a riverside is reached the walk must continue from the island / riverside.

In 1736 Leonard Euler proved that such a walk does not exist. He described the above situation by four points (vertices) representing two islands and two riversides, connected by seven lines (edges) representing bridges. We will learn more about his argument in the section Euler graphs 7.7.

Let us start with the notion of a directed graph even though the above problem led to an undirected one.

7.1.1 Directed Graphs. Roughly speaking, a directed graph consists of points called vertices, lines that go from one vertex to other vertex called edges. Because there may be more than one edge from A to B, in a general graph we distinguish between a name of an edge and the order pair A and B. More precisely:

Definition. A *directed graph* is a triple $G = (V, E, \varepsilon)$ where V is a nonempty finite set of *vertices* (also called *nodes*), E is a finite set of names of *directed edges* (also called *arrows*), and ε is an *incidence relation* which assigns to any edge $e \in E$ an ordered pair (u, v) of vertices $u, v \in V$. □

Further notions for directed graphs. Let $\varepsilon(e) = (u, v)$. Then u is called the *initial vertex* of e, denoted by $IV(e)$, and v the *terminal vertex* of e, denoted by $TV(e)$. We also say that vertices u, v are *end vertices* of e, or that e is *incident* to u, v.

If $u = v$ we call the edge e a *directed loop*.

If for two edges e_1 and e_2 we have $\varepsilon(e_1) = \varepsilon(e_2)$, the edges e_1 and e_2 are called *parallel.* □

7.1.2 Undirected Graphs. Roughly speaking, undirected graphs are graphs in which any edge "can be used in both directions" or "where the direction is not important". More precisely:

Definition. An *undirected graph* is a triple $G = (V, E, \varepsilon)$ where V is a nonempty finite set of *vertices* (also called *nodes*), E is a finite set of names of *edges*, and ε an *incidence relation* which assigns to any edge $e \in E$ a set $\{u, v\}$ where $u, v \in V$. □

Further notions for undirected graphs. Let $\varepsilon(e) = \{u, v\}$. Then u and v are called *end vertices* of e.

If $u = v$ then we call the edge e an *undirected loop.*

If for two edges e_1 and e_2 we have $\varepsilon(e_1) = \varepsilon(e_2)$, the edges e_1 and e_2 are called *parallel edges.*

If $\varepsilon(e) = \{u, v\}$ we say that vertices u, v are *incident to the edge* e and that the edge e is *incident to vertices* u, v.

7.1.3 Simple Graphs.

Definition. A graph (directed or undirected) is called *simple* if it does not contain parallel edges. □

Remark. In a simple graph the incidence relation is not necessary. Indeed, assume that G is a simple directed graph. Then any edge can be named by the ordered pair $(IV(e), TV(e))$ of its initial vertex $IV(e)$ and terminal vertex $TV(e)$. Therefore, in a simple directed graph the set of edges E will be a subset of $V \times V$. Similarly, if G is undirected graph, then for each $\{u, v\}$, $u, v \in V$ there is at most one edge with $\varepsilon(e) = \{u, v\}$. Hence $\{u, v\}$ can serve as the name of e. Here, $E \subseteq \{\{u, v\} \mid u, v \in V\}$.

In the following text, a simple graph (directed or undirected) will be denoted by $G = (V, E)$.

We will denote the set of vertices of a graph G by $V(G)$ and the set of edges by $E(G)$. If there is no fear of misunderstanding, we will write only V for the set of vertices, and E for the set of edges.

7.1.4 Vertex Degrees.

Roughly speaking, degree of a vertex is the number of edges that end or start in a given vertex. In directed graphs it will be useful to distinguish between the number of edges that start in v (out degree) and the number of edges that terminate in v (in degree).

Definition. Given a directed graph $G = (V, E, \varepsilon)$. The *in-degree* of a vertex v, denoted by $d^-(v)$, equals to the number of edges for which v is the terminal vertex; i.e.

$$d^-(v) = |\{e \in E; TV(e) = v\}|.$$

The *out-degree* of a vertex v, denoted by $d^+(v)$, equals to the number of edges for which v is its initial vertex; i.e.

$$d^+(v) = |\{e \in E; IV(e) = v\}|.$$

The *degree* of a vertex v, denoted by $d(v)$, is

$$d(v) = d^-(v) + d^+(v),$$

(i.e. it is the number of edges that are incident to v). □

Notice that any loop is calculated twice; indeed, once for the in-degree, once for the out-degree of v.

Definition. Given an undirected graph $G = (V, E, \varepsilon)$. The *degree* of a vertex v, denoted by $d(v)$, equals to the number of edges for which v is their end vertex where a loop is calculated twice. □

Notice that if we "forget" direction in a directed graph G and obtain an undirected graph G', then for every vertex v we have $d_G(v) = d_{G'}(v)$, i.e. the degrees are the same. Indeed, this is due to the fact that each loop is calculated twice even in an undirected graph.

7.1.5 One of the easy but extremely useful facts about degrees in a graph is the following proposition. (In the English literature, it is also called Handshaking lemma.)

Proposition. For every graph G (directed or undirected) we have

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

where $|E|$ is the number of edges in G. □

Justification. Let e be an edge of a graph G. Then e adds 2 to the sum of all degrees; indeed, if e is a loop then it is calculated twice by definition, if e is not a loop then it has two end vertices. Therefore, the sum of all degrees is twice the number of edges. □

Since by the proposition above, the sum of all degrees is an even number, we get the next corollary.

Corollary. Every graph has an even number of vertices with odd degree. □

7.1.6 Walks in a Graph. In many problems using graphs, "walking" through vertices and edges is the main goal. We start with the most general notion, the one that only requires that a next edge starts in the vertex where the previous edge terminates. In directed graphs we distinguish between directed and undirected walks. The difference is, roughly speaking, that in directed walks we must go from the initial vertex of an edge to its terminal vertex (so in the direction of the edge). In an undirected walk, we can go even against the direction of the edge, i.e. from the terminal vertex to the initial vertex of an edge.

Definition. Given a directed graph G. A *directed walk* in G is a sequence of vertices and edges

$$v_1, e_1, v_2, e_2, \ldots, v_{k-1}, e_{k-1}, v_k$$

such that for every $i = 1, 2, \ldots, k-1$ it holds that $v_i = IV(e_i)$ and $v_{i+1} = TV(e_i)$.

An *undirected walk* in a directed or undirected graph G is a sequence of vertices and edges

$$v_1, e_1, v_2, e_2, \ldots, v_{k-1}, e_{k-1}, v_k$$

such that for every $i = 1, 2, \ldots, k-1$ vertices v_i and v_{i+1} are end vertices of e_i.

Given a directed (or undirected) walk. We say that v_1 is the *initial vertex* of the walk, and v_k is the *terminal vertex* of the walk. Also we say that the *walk goes from v_1 to v_k*. □

Remark: We define an undirected walk also in an undirected graph because the definition is the same. Note that the notion of a directed walk in undirected graphs is meaningless.

7.1.7 A Trivial Walk. A walk is defined as a sequence of vertices and edges with some additional properties. The sequence may contain only one vertex and no edge. In that case it is called trivial.

Definition. A *trivial walk* is a walk that contains only one vertex and no edge. □

We consider it as a directed and also an undirected walk.

7.1.8 Closed Walks. Roughly speaking, a walk (directed or undirected) is closed if the initial vertex of it equals the terminal one and the walk contains at least one edge.

Definition. A directed (an undirected) walk is called *closed* if $k > 1$ and $v_1 = v_k$. Otherwise it is called an *open* walk. □

Hence, a trivial walk is not closed; indeed, $k = 1$.

7.1.9 Trails and Paths, Cycles and Circuits. In 7.1.6 we defined the notion of a walk. There are special types of walks that play an important role in applications. These are trials and their special cases paths.

Definition. A directed (an undirected) walk $v_1, e_1, \ldots, e_{k-1}, v_k$ is called a *directed trail* (an *undirected trail*) if it contains every edges at most once. In other words, for $i \neq j$ it holds that $e_i \neq e_j$.

A directed (or an undirected) trail $v_1, e_1, \ldots, e_{k-1}, v_k$ is called a *directed path* (an *undirected path*) if it contains every vertex at most once with the exception that v_1 may be the same as v_k.

A *cycle* is a closed directed path, i.e. a directed path with $v_1 = v_k$, $k > 1$.

A *circuit* is a closed undirected path, i.e. i.e. an undirected path with $v_1 = v_k$, $k > 1$. □

7.1.10 Remarks. A cycle (a circuit) can also be defined as an open directed (undirected) path together with one edge from the terminal vertex to the initial vertex (between initial and terminal vertices) of the path.

Every path is a trail, and every trail is a walk; the opposite does not hold. Also every cycle is a circuit, but not every circuit in a directed graph is a cycle.

Notice, that a trivial walk is a trail and a path, but it is neither a circuit nor a cycle.

7.1.11 Reachability.

Definition. Given a directed or undirected graph $G = (V, E, \varepsilon)$. We say that a vertex v is *reachable* from a vertex w if there exists an undirected path from w to v. □

Remarks. 1. In the definition above we could require the existence of a walk from w to v and we would get the same notion. Indeed, any path is a walk; on the other hand, any walk from w to v contains a path form w to v:

Assume that $P = v_1, e_1, v_2, \ldots, v_{k-1}, e_{k-1}, v_k$ is a walk which is not a path. Let $v_i = v_j$ for $i < j$. Then

$$P_1 = v_1, e_1, \ldots, v_i, e_j, \ldots v_k$$

is a walk from v_1 to v_k which is shorter. (Roughly speaking, we cut off one closed walk of P.) If P_1 is a path, we are done. If not, we again find $r \neq s$ such that $v_r = v_s$ and we omit the walk from v_r to v_s. Since any walk cannot have less than 0 edges, the procedure must end, and we are left with a path from v_1 to v_k.

2. The relation of reachability is reflexive; indeed, v is reachable from itself by the trivial path v.

3. The relation of reachability is symmetric; it means that if v is reachable from w then so is w from v. The reason is that any path from w to v can be viewed as a path from v to w. Indeed, if $w, e_1, \ldots, e_{k-1}, v$ is an undirected path, then so is $v, e_{k-1}, \ldots, e_1, w$.

4. Note that the relation of reachability is transitive; it means if v is reachable form w and u is reachable from v, then u is reachable from w. If we join a path from w to v and a path from v to u, we get a walk from w to u. And the walk contains a path with the same initial and terminal vertices.

7.1.12 Connected Graphs.

Definition. A graph G (directed or undirected) is called *connected* if for every two vertices u, v of G there exists an undirected path form u to v (i.e. every vertex is reachable from any vertex).

If a graph is not connected then it is called *disconnected.* □

7.2 Trees

Trees form a class of undirected (or directed) graphs that one will come across in many applications, not only in computer science but also in many engineering applications. Even in this course we have spoken about syntactic trees for propositional or predicate formulas.

7.2.1 Definition. A graph G (directed or undirected) is called a *tree* if it is connected and it does not contain a circuit. □

7.2.2 Trees satisfy an interesting property – every tree with n vertices has $n-1$ edges. To show this we need the following lemma.

Lemma. Let G be a tree with at least two vertices. Then G contains at least one vertex with degree 1. □

Justification: We proceed by contradiction: Assume that G does not have a vertex v with $d(v) = 1$, so $d(u) \geq 2$ for any vertex u. Let us from a walk as follows:

Start in any vertex and denote it by v_1. Since $d(v_1) \geq 2$, there is an edge, say e_1, incident with v_1. Denote by v_2 its second end vertex. Necessarily $v_2 \neq v_1$. Indeed, otherwise e_1 is a loop and every loop is a circuit. Since $d(v_2) \geq 2$ there is an edge e_2, $e_2 \neq e_1$, incident with v_2. Denote by v_3 the other end vertex of e_2. Since G does not contain a circuit, v_3 is a new vertex of degree at least 2, so we continue. In that way we obtain a path $v_1, e_1, v_2, \ldots, e_{n-1}, v_n$ where n is the number of vertices. But $d(v_n) \geq 2$, hence there exists an edge different form e_{n-1}, say e_n, incident to v_n. The other end vertex of e_n must be one of $v_1, v_2, \ldots, v_n$ (indeed, we do not have any other vertex), therefore e_n closes a circuit – a contradiction with the fact that G is without circuits. □

7.2.3 Theorem. Every tree with n vertices has precisely $n-1$ edges. □

Justification: We proceed by mathematical induction:

Basic step. If $n = 1$ there is only one tree – a vertex with no edge. If $n = 2$ there is again only one tree – two vertices joint by one edge. Hence for $n = 1$ or $n = 2$ the assertion is true.

Inductive step. Assume that every tree with n vertices has $n-1$ edges. Consider any tree G with $n+1$ vertices. From the lemma above, we know that G contains a vertex v with $d_G(v) = 1$. Let us remove v and the one edge incident to v from the graph G. We obtain a graph G' which is connected and has no circuit, so G' is a tree with n vertices. By the induction assumption, G' has $n-1$ edges, hence G has $n-1+1 = n$ edges. The proof is complete. □

7.2.4 Proposition. Every tree with at least two vertices contains at least two vertices of degree 1. □

Justification. There are two different (and easy) proofs.

1. We know that $\sum_{v \in V} d(v) = 2\,|E|$ (see 7.1.5), and hence $\sum_{v \in V} d(v) = 2\,(n-1)$, where $n \geq 2$ is the number of vertices. If there was only one vertex v with $d(v) = 1$, then $\sum_{v \in V} d(v) \geq 1 + 2(n-1)$; a contradiction.

2. There is also a direct proof which does not use 7.2.3. Consider a maximal path in the tree G, i.e. a path that is not contained in any longer path. Denote the path by $P = u_1, e_1, \ldots, e_{k-1}, u_k$. Then $d(u_1) = 1 = d(u_k)$. Indeed, if $d(u_1) \geq 2$ then either the path P is not maximal, or there is a circuit containing u_1. Similarly for u_k. □

7.2.5 Several Characterizations of Trees. The following theorem gives two other characterizations of trees.

Theorem. Given a graph G. Then the following are equivalent:

1. G is a tree.

2. G contains no circuit and if we add any new edge (the set of vertices remains the same), then we close just one circuit.
3. G is connected and removing any edge disconnects G.

□

Justification. 1. implies 2. Assume that G is a tree. Then from the definition, G does not contain a circuit. Assume that we add a new edge e between u and v. Since G is connected, there exists a path P from u to v. Now, the path P together with e (e does not belong to P) forms a circuit. Hence e has closed a circuit.

Assume that e has closed two circuits, say C_1 and C_2. Then there are two distinct paths P_1 and P_2 from u to v. If we join P_1 and P_2 we get a new circuit in this case belonging to G – a contradiction with the fact that G does not have a circuit.

2. implies 3. Assume that G contains no circuit and adding any new edge closes just one circuit. Then G is connected; indeed, assume that there was no path between u and v; in this case, adding a new edge with end vertices u and v does not close a circuit.

Let us remove an edge e with end vertices x and y. If G stays connected then there is a path P in G from x to y which does not contain e. Hence, P together with e forms a circuit – a contradiction with the fact that G does not contain a circuit. So removing any edge disconnects the graph G.

3. implies 1. Assume that G is connected and removing any of its edges disconnects it. We have to show that G is a tree, i.e. it is connected and does not contain a circuit. Assume for contrary that G contained a circuit. Take any edge e from the circuit (it must exist since any circuit contains at least one edge). Then removing e from G does not disconnect it – a contradiction. □

7.2.6 Remarks. Given any connected graph G. If we add one edge (without adding a new vertex) to the set of edges of G, then the new graph remains connected.

If a graph G contains no circuit and we remove one edge, then the new graph will also contain no circuit.

A tree is a graph tha t has the smallest number of edges to be connected and the biggest number of edges to be without circuits.

7.2.7 Subgraphs. Roughly speaking, a subgraph of a given graph is obtained by: forgetting some (maybe none) vertices, forgetting some (maybe none, maybe all) edges, but if an edge is in the subgraph then also both its end vertices are in the subgraph. More precisely:

Definition. Given a graph $G = (V, E, \varepsilon)$. A *subgraph* of G is a triple $G' = (V', E', \varepsilon')$ where

- $V' \subseteq V$,
- $E' \subseteq E$, and
- ε' is the restriction of ε on the set E'.

□

We distinguish two special types of subgraphs:

Definition. Given a graph $G = (V, E, \varepsilon)$.

- A *factor* is a subgraph where $V' = V$, i.e. it contains all vertices of G.
- Let $A \subseteq V$. The *subgraph induced by A* is the subgraph with $V' = A$ and $e \in E$" if and only if $e \in E(G)$ and the end vertices of e belong to A.

□

Notice, that the subgraph induced by A is the "maximum" subgraph with the set of vertices A.

7.2.8 Components of Connectivity. Since not every graph is connected, it is useful to "divide" the set of vertices into parts that are connected. Roughly speaking, a component of connectivity is a maximal part of the graph "which is connected". More precisely:

Definition. Given a graph G (directed or undirected). A *component of connectivity* (sometimes also called a *component of weak connectivity*) is a maximal subset A of $V(G)$ such that the subgraph induced by A is connected. □

By a maximal subset we mean that the subgraph induced by A is connected but for every $v \in V(G)$, $v \notin A$, the subgraph induced by $A \cup \{v\}$ is not connected.

Remarks.

1. We can define the components of connectivity also in a different way. Given a graph G with the set of vertices V. Define a relation R on V to be the reachability relation, i.e. by

$$u\,R\,v \text{ if and only if } v \text{ is reachable from } u.$$

We know from 7.1.11 that R is an equivalence relation on V. Components of connectivity are now the classes of the equivalence R.

2. A graph is connected if and only if it has only one component of connectivity.