Discrete Mathematic

Bechtold Simon
Matr.# 0826041
Kennzahl 938
Institute of Computer Engineering
Vienna University of Technology
e0826041@student.tuwien.ac.at

October 17, 2013

1 Graph Theory

1.1 Basics

A graph G is a pair of two sets V and E. Where V represents the vertices and E represents the edges.

Definition 1.1. graph

$$G = (V, E) \tag{1}$$

In a directed Graph each edge $e \in E$ is represented as a pair of Vertices $(v, w) \quad v, w \in V$. (v, w) is not equal to (w, v).

Definition 1.2. edge

$$\forall e \in E : e = (v, w) \quad v, w \in V \tag{2}$$

In a undirected Graph each edge is represented as a set of vertices v, w. Where this time v, w is equal to w, v. A loop edge is a edge v, w where $v = w$. A multiple edge consistes of multiple connections between two vertices v, w is several times a element of E.

Definition 1.3. simple graph
A graph G is called simple graph if there are no loops and multiple edges. Unless explicitly stated the graphs in the lecture alre always simple graphs.

A graph corresponds to a relation on $V \quad C \subseteq V \times V$. A undirected graph always corresponds to a symmetric relation.

Notations

- V Vertex set
- E Edge set
- $V(G)$ Vertex set of graph G
- $E(G)$ Edge set of graph E
- $\alpha_0 = |V|$ Number of vertices
- $\alpha_1 = |E|$ Number of edges

Definition 1.4. incident
A edge e is incident to a vertex v if v is a part of e.

Definition 1.5. Degree of a vertex in a undirected graph

$$v \in V : d(v) = \text{Number of edges which are incident to } v \tag{3}$$

Definition 1.6. Degree of a vertex in a directed graph

$$d^+(v) = \text{\# of edges of form } (v, w) \text{ (out-degree)} \tag{4}$$
$$d^-(v) = \text{\# of edges of form } (w, v) \text{ (in-degree)} \tag{5}$$

Definition 1.7. Neighborhood in a undirected graph

$$\Gamma(v) = \text{set of neighbors of } v \tag{6}$$

Definition 1.8. Neighborhood in a directed graph

$$\Gamma^+(v) = \text{set of successors of } v \tag{7}$$
$$\Gamma^-(v) = \text{set of predecessors of } v \tag{8}$$

Lemma 1.1. *Handshaking lemma in a undirected graph*

$$\sum_{v \in V} d(v) = 2 * |E| \tag{9}$$

Proof. A Edge is always counted twice □

Lemma 1.2. *Handshaking lemma in a directed graph*

$$\sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v) = |E| \tag{10}$$

Proof. A edge outgoing is also counted as a edge incoming for another vertex □

Example 1.1. Hypercube

- A one dimensional cube is a line. The coordinate of the start and end point are given by 0 and 1.
- A two dimensional cube is a square whre the coordinates are given by 00, 01, 10, 11.
- A three dimensional cube is a qube given by the coordinate 000, 001,...
- A n-dimensional cube is a hypercube.

So a hypercube is a graph where the vertices are described as strings of 0 and 1 with the length of n. And on each adjacent vertices only one bit flips $\sum_{i=1}^{n} |v_i - w_i| = 1$

$$G = (0, 1^n, E) \tag{11}$$

$$(v, w) \in E \iff \sum_{i=1}^{n} |v_i - w_i| = 1 \tag{12}$$

Therefore we get the attributes of the hypercube easaly by using the handshaking lemma.

$$\alpha_0 = 2^n \tag{13}$$

$$\alpha_1 = \frac{1}{2} \sum_{v \in V} d(v) = 2^{n-1} * n \tag{14}$$

Definition 1.9. A n-regular graph is a graph where $d(v) = n$

Definition 1.10. Adjacency
If $e : (v, w) \in E \implies v, w$ are adjacent.

Definition 1.11. Incident
Translated as connected..?

Definition 1.12. Adjacency Matrix

$$A = (a_{ij})_{i,j=1...(n=\alpha_0)} \tag{15}$$

$$a_{ij} = \begin{cases} 1 & v_i \sim v_j \text{ adjacent} \\ 0 & v_i \nsim v_j \text{ not adjacent} \end{cases} \tag{16}$$

Remark. If G is undirected $\implies A$ is a symmetric Matrix

Remark.

$$A^k = (a_{ij}^{[k]})_{i,j=1..n} \tag{17}$$

$$a_{ij}^{[k]} = \sum_{l=1..n}^{n} a_{il} * a_{lj}^{[k-1]} \tag{18}$$

- A connection in one step
- A^k connection in k steps
- $A^0 = I$ identity matrix

Definition 1.13. .

- A **walk** in a graph is a sequence of edges where any two successive edges have one same vertex.
- A **trail** is a walk where no edge is repeated.
- A **closed trail** is a circiut

Definition 1.14. Subgraph
Considered a Graph $G = (V, E)$ then its subgraph H is defined as $H = (V', E')$ (short form $H \leq G$) where $V' \subseteq V$, $E' \subseteq E$ and H is also a graph ($\forall (v', w') \in E' : v', w' \in V'$).

Definition 1.15. connection relation R
v is connected to w (vRw) $\iff$ there is a walk from v to w ($v \rightsquigarrow w$). The connection matrix is defined as

$$C = \sum_{k=0}^{L} A^k \quad L = \min(|E|, |V - 1|) \tag{19}$$

where L is the length of the walk

$$C = (c_{ij}) \tag{20}$$

c_{ij} is the number of walks of length $l \leq L$ that connect v_i to v_j

Definition 1.16.

$$M = sgn(C) \tag{21}$$

Remark. .

- $\forall v \in V : vRV$
- $\forall v, w \in V : vRw \implies wRv$
- $\forall v, w, u \in V : vRw\hat{w}Ru \implies vRu$

These show that R is a equivalence relation. Since R is a equivalence relation, R induces a partitioning of V.

Definition 1.17. Connected components (partitions induced by R)

$$V = V_1 \cup V_2 \cup ...V_n \tag{22}$$

$$V_i \cap V_j = \emptyset \quad i \neq j \tag{23}$$

Definition 1.18. G connected if $\forall v, w : vRw$

Definition 1.19. $H \leq G$ connected components of G if H is connected and H is maximal.

Definition 1.20. H is maximal if there is no graph H' : $H < H' \leq G$ and H' is connected. The maximal definition supplies to vertices and edges

Definition 1.21. $vSw :\iff \exists$ a walk $v \rightsquigarrow w$ and a walk $w \rightsquigarrow v$

S like R is a equivalent relation and thus also induces partiotioning.

Definition 1.22. G is strongly connected $:\iff \forall v, w \in V : vSw$

Definition 1.23. $H \leq G$, H is maximal strongly connected. If $H \leq H' \leq G$ and $H \neq H'$ then H' is not strongly connected $\implies$ H is strongly connected component of G

Definition 1.24. G is strongly connected if $\forall v, w \quad vSw$. $H \leq G \quad H$ strongly connected component if H is maximal strongly connected.

Remark. G strongly connected component $\iff$ H has 1 strongly connected component.

Definition 1.25. shadow
G is directed, $H = G$ without directions and multiple connections deleted $\implies$ H shadow of G.

Definition 1.26. weakly connected
G is weakly connected $\iff$ H (shadow of G) is connected.

Definition 1.27. reduction of G
G is a directed graph.. $G_R = (V_R, E_R)$ is simple. $V_R = \{K_1, K_2, ..., K_m\}$ is the set of strongly connected components of G. $E_R = \{(K_i, K_j) | \exists v \in V(K_i), \exists w \in V(K_j), (v, w) \in E$

The reduction is the graph resulting of the interconnect between strongly connected components.

Remark. G_R is always acyclic. If G is strongyl connected then $G_R = (\{.\}, \emptyset)$ while $\{.\}$ is one vertex.

Definition 1.28. node base
we have $G = (V, E)$ directed graph then B is a nodebase if

- $B \subseteq V$
- $\forall v \in V \quad \exists w\, inB : wSv$ ... There is a path from v to any other element of V
- B minimal with relation to (w.r.t) $\subseteq$ ... no subset of B is a node base

Remark. The node base of G can be constructed from the node bases of G_R. $\{K_1 ... K_i\}$ node base of $G_R \implies \{\{b_1, ..., b_i | b_i \in V(K_i)\}\}$ ... set of all node bases of G ... Take one node of every component of the node base of G_R

Definition 1.29. Node base of G_R is $\{K \in V_R | d^-_{G_R}(K) = 0\}$ of a acyclic graph.

1.2 TREES AND FORESTS

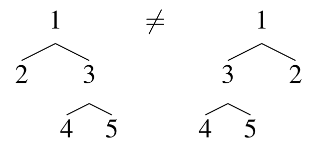
Definition 1.30. forest
A forest is a undirected graph without cycles.

Definition 1.31. tree
A tree is a connected forest.

Definition 1.32. rooted tree
One vertex of the tree is considered as the root.

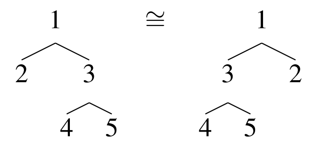
Remark. The root node is never considered to be a leaf.

Definition 1.33. plane tree
A tree that is embeddded into the plane. This means the left and right order of the tree is important.



Definition 1.34. isomorphism
Two graphs G, H are isomorphic $G \cong H$ if $\exists \varphi V(G) \to V(H)$ where

- φ is bijective (1 to 1 correspondance between the vertices of the first and the second graph)
- $(v, w) \in E(G) \iff (\varphi(v), \varphi(w)) \in E(H)$



Definition 1.35. leaf
A leaf is a vertex of degree 1

Lemma 1.3. *T is a Tree and $|V(T)| \geq 2 \implies T$ has at least 2 leaves*

Proof. .

- Tree of size two: Only possible tree is a tree with one edge between the two vertices. Therefore both vertices have degree 1.

  Remark. This is only true for unrooted trees, since a root is never considered to be a leaf
- $T \geq 3$ start at any Vertex; thre must be at least one neighbor..
  1. remove of a leave: $|V(T)| = k + 1 \implies |V(T'| = k \implies T'$ has 2 leaves
  2. remove of a node: $\implies T', T'' \implies$ two new trees with each of them must have ≥ 2 leafes

□

Definition 1.36. characterization
A characterization is a neccessary sufficient definition.

Definition 1.37. bridge
Removal of a bridge would increase the numer of components in the graph.

Theorem 1.4. *The following 5 statements are equivalent*

1. *T is a Tree*
2. $\forall v, w \in V(T) \exists!$ *(exists only one) path from v to w*
3. *T is connected an* $|V| = |E| + 1$
4. *T is a minimal connected graph (every edge is a bridge)*
5. *T is a maximal acyclic graph*

Proof. of $(1) \implies (3)$; induction on $n = \alpha_0 = |V(T)|$

- $n = 1$: The only possible graph is a vertex with no edges
- $n \to n + 1$: choose a leaf v of T. $T' = T$ without the leaf. T' is a tree because removing a leaf does not induce a cycle. $\implies |V(T')| = |E(T')| + 1$
  $|V(T)| = |V(T')| + 1$
  $|E(T)| = |E(T')| + 1$

To complete the proof of the equivalence $(3) \implies (1)$ needs also to be profed. □

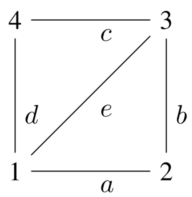
1.2.1 spanning subgraph

Definition 1.38. spanning forrest:
G is a undirected graph. F is a spanning forrest of $G \iff$

1. $V(F) = V(G)$
   $E(F) \subseteq E(G)$
2. F is a forrest
3. F has the same connected components as G

Definition 1.39. spanning tree:
If F is connected then F is a spanning tree.

Example 1.2. Construction of a spanning subgraph



$$\widetilde{A} = \begin{pmatrix} 0 & a & d & e \\ a & 0 & 0 & b \\ d & 0 & 0 & c \\ e & b & c & 0 \end{pmatrix} \tag{24}$$

From the *adjacency* matrix the row sum matrix is formed.

$$\widetilde{D} = \begin{pmatrix} a+b+e & 0 & 0 & 0 \\ 0 & a+b & 0 & 0 \\ 0 & 0 & c+d & 0 \\ 0 & 0 & 0 & b+c+e \end{pmatrix} \tag{25}$$

$$\widetilde{D} - \widetilde{A} = \begin{pmatrix} a+b+e & -a & -d & -e \\ -a & a+b & 0 & -b \\ -d & 0 & c+d & -c \\ -e & -b & -c & b+c+e \end{pmatrix} \tag{26}$$

Erste Zeile und erste Spalte werden gestrichen. Daraus dann die Determinante bestimmt.

$$\begin{vmatrix} a+b & 0 & -b \\ 0 & c+d & -c \\ -b & -c & b+c+e \end{vmatrix} = \begin{matrix} bcd + abc + abd + acd + \\ ace + ade + bce + bde \end{matrix} \tag{27}$$

Every term represets a sanning tree.

Example 1.3. To calculate the number of spanning trees set $a = b = c = d = e = 1$

$$\widetilde{A} \to A \tag{28}$$

$$\widetilde{D} \to D = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix} = 8 \tag{29}$$

Theorem 1.5. *Matrix-Tree-Theorem (Kirchoff)*
G is a undirected connected graph. $V = \{v_1, v_2, .., v_n\}$; $A =$ adjecency matrix; $D =$ degree matrix. $\implies |\det((D-))')| =$ number of spanning trees of G. The $'$ notates the deletion of one column and one row no matter which one. If G is not connected then do it for every component and multiply $\to$ spanning forest

Definition 1.40. Minimal spanning treee (from mst.pdf)
The unique spanning tree T of a weighted graph $G = (V, E, w)$ is the tree that minimizes $\sum_{e \in T} w(e)$.

Algorithm 1.6. *Kruskals MST algorithm*

Algorithm 1.7. *Prim's MST algorithm*

1.2.2 Matroids and greedy algorithms

Remark. Kruskal and prim's are greedy algorithms for maximization or minimization.

Definition 1.41. Matroid
A structure such that greedy algorithms work

In kruskals algorithm the edges are sorted in decreasing order. We take them in a greedy way but such that no cycles are induced.

$G = (V, E), S = \{F \subseteq E | F \text{is a forrest}\}, \emptyset \in S$. Kuskals constructs a set T such that $T = T \cup \{e\}$ if $T \cup \{e\} \in S$. Where $T \cup \{e\} \in S$ means that the tree with the edge e is still a forrest or a empty set.

Definition 1.42. independence system
(E, S) is a independence system if $S \subseteq 2^E$ and S is closed under inclusion (i.e. $A \in S, B \in A \implies B \in S$.

Definition 1.43. independence set
S set of independent sets. If a edge-set contains cycles, it is a dependend set. A set is independent if it contains no cycles (is a forrest).

Definition 1.44. optimization problem
$(E, S); w : E \to \mathbb{R}^+; A \subseteq E; w(A) = \sum_{e \in A} w(e); w(A) \to \max'(\min'), A \in S, A$ max w.r.t $\subseteq$

Example 1.4. (E, S) Edge set of forrestss $\implies$ independent system.
The generalized kruskal: GREEDY(E, S, w, T)

1. sort elements of E by weight (decreasing)
   $E = \{e_1, e_2, ..., e_3\}$
   $w(e_1) \geq w(e_2) \geq ... \geq w(e_n)$
2. $T := \emptyset$
3. for $k = 1$ to m do
   if $T \cup \{e_k\} \in S$ then $T := T \cup \{e_k\}$
   end

In general greedy fails, but it works or spanning trees, so there must be more in the properties of spanning trees which makes greedy work.

Definition 1.45. The independence system $M = (E, S)$ is called a matroid if $A, B \in S$ such that $|B| = |A| + 1 \implies \exists v \in B \setminus A$ with $A \cup \{v\} \in S$.

Remark. The matroid property holds for $A, B \in S$ such that $|A| < |B|$ as well.

Definition 1.46. $A \in S$ is a basis of $M \iff A$ is a maximal independent set w.r.t. $\subseteq$

Remark. A, B basis of $M \implies |A| = |B| = r(M)$ while $r(M)$ denotes the rank of M.

Theorem 1.8. *$G = (V, E)$ is a graph, $S = \{F \subseteq E | F$ forrest$\} \implies (E, S)$ is a matroid.*

Proof. $F_1, F_2 \subseteq E; F_2 \in S, F_1 \subseteq F_2 \implies F_1 \in S$
matroid property: $F_1, F_2 \in S \quad |F_2| = |F_1| + 1$
$F_1, ...m$ components $T_i = (V_i, E_i) \quad i = 1..n$
Observe:
$V = V_1 \cup V_2 \cup ... \cup V_m, F_1 = A_1 \cup ... \cup A_m, |A_i| = |V_i| - 1$
F_2 is a forrest $\implies$ there are at most $|V_i - 1|$ edges in F_2 which connect $v, w \in V_i, F_2 > F_1 \implies \exists$ edge e which connects two components $\to F_1 \cup \{e\}$ forrest. □

Theorem 1.9. *Let $M = (E, S)$ be a matroid with weight function $w : E \to \mathbb{R} \implies$ GREEDY solves A maximal w.r.t. $\subseteq$ such that $w(A)$ is minimal (max) correctly. GREEDY computes the basis with minimal(max) weight.*

Proof. $A = \{a_1, a_2, ..., a_r\}$ set after computing GREEDY $\implies$

1. A basis : $A \in S$ by construction.
   A is not maximal $\implies \exists e \in E$ such that $A \cup \{e\} \in S \implies$ contradiction
2. $w(a_1) \leq w(a_2) \leq ...$ true because sorted
3. $w(A)$ is minimal

LOOKUP THIS PROOF! □

Theorem 1.10. *$M = (E, S)$ independence system*
Assume: GREEDY solves the optimization problem A max such that $w(A)$ is max correctly for all weight functions. $\implies M$ is a matroid

Proof. Assume M is not a matroid $\implies \exists A, B \in S : |B| = |A| + 1 \wedge \forall x \in B \setminus A : A \cup \{x\} \notin S$
We set $w(e)$ such that:

$$w(e) = \begin{cases} |A| + 2 & \text{if } e \in A \\ |A| + 1 & \text{if } e \in B \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

$\implies w(A) = |A| + (|A| + 2) < (|A| + 1)^2 \leq w(B)$ while $|B| = |A| + 1$
$\implies A$ neighter a solution of the optimization problem nor $w(A) := maximum$ □

GREEDY chooses $x \in A$ first because $w(A) < w(B)$ then $w(A)$ can not be increased onymore $x \in B \setminus A \implies A \cup \{x\} \notin S$.
$x \in A \cup B \implies$ GREEDY arrives eventually at a set N such that $w(N) = w(A)$ is not maximal.