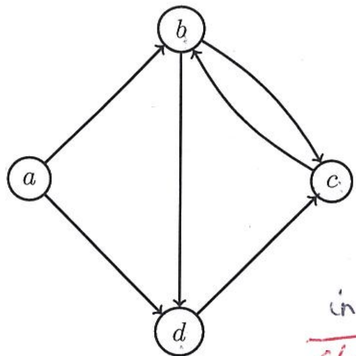
Lecture 16: Directed Graphs

Date: October 7, 2019.

Directed Graphs. G consists of nonempty set $V(G)$ of **vertices** (or **nodes**) and a set $E(G)$ of **edges**. Here $E(G) \subseteq V(G) \times V(G)$. An edge (u, v) has **source/tail** u and **target/head** v. A directed graph $G = (V(G), E(G))$ is also called a **digraph**.



$V(G) = \{a, b, c, d\}$

$E(G) = \{(a,b), (b,d), (b,c), (c,b), (d,c), (a,d)\}$

$\text{indeg}(a) = 0 = |\{\}|$

$\text{outdeg}(a) = 2 = |\{(a,b), (a,d)\}|$

$0+2+2+2 = 2+2+1+1$

indeg

Walk: $a\,(a,b)\,b\,(b,c)\,c\,(c,b)\,b$ — Not a path

$a\,(a,d)\,d\,(d,c)\,c\,(c,b)\,b$ — Path

Closed Walk: $b\,(b,c)\,c\,(c,b)\,b$ — cycle ; $d\,(d,c)\,c\,(c,b)\,b\,(b,c)\,c\,(c,b)\,b\,(b,d)\,d$ — Not cycle

Degrees. For a vertex $v \in V(G)$ of digraph G

$|\{(u,v) \in E(G) \mid u \in V(G)\}| = \text{indeg}(v) = |\{(u, v) \mid u \in V(G)\}|$ in-degree

$|\{(v,u) \in E(G) \mid u \in V(G)\}| = \text{outdeg}(v) = |\{(v, u) \mid u \in V(G)\}|$ out-degree

Proposition 1. *For any graph G, $\sum_{v \in V(G)} \text{indeg}(v) = \sum_{v \in V(G)} \text{outdeg}(v)$.* $= |E(G)|$

Walks. A walk is an alternating sequence of vertices and edges that begins with a vertex, ends with a vertex, and such that for every edge (u, v) in the walk, u is the element just before the edge, and v is the element just after the edge in the sequence. So it is of the form

$$v_0(v_0, v_1)v_1(v_1, v_2) \cdots (v_{k-1}, v_k)v_k.$$

The walk is said to **start** in v_0 and end in v_k, and is of **length** k.

Simplification. A walk is completely determined by just the (sub-)sequence of vertices or the (sub-)sequence of edges. So we will just use that when convenient.

Paths. Is a walk, where each vertex in the sequence is distinct.

Closed Walk. Is a walk that starts and ends in the same vertex.

Cycle. Is a closed walk of length > 0 where all vertices except the first and last vertex are distinct.

Combining walks. If a walk $\mathbf{f}$ ends in vertex v and a walk $\mathbf{g}$ starts at the same vertex v, then they can be *merged* to get a longer walk. We will denote the merged walk by $\mathbf{f}^\frown\mathbf{g}$. Sometimes to emphasize the vertex where the walks merge, we will denote this by $\mathbf{f}\,\widehat{v}\,\mathbf{g}$.

Note, that $|\mathbf{f}^\frown\mathbf{g}| = |\mathbf{f}| + |\mathbf{g}|$.

Examples of Graphs

State Machines

Vertices are states
Edges are transitions

Influence Graphs

Vertices people
Edges are indicate when one person can influence

Web Graph

Vertices are webpages
Edges are links

Call graphs:

Vertices are berties
Edges are calls.

Module Dependency Graphs:

Vertices are modules
Edges model dependencies.

Precedence Graphs:

Vertices are functions
Edges models functional dependency

Theorem 2. *A shortest walk between two vertices is a path.*

Suppose f is a shortest walk from u to v that is not a path (contradiction)
$f = x\,\hat{a}\,y\,\hat{a}\,z$. Consider $f' = x\,\hat{a}\,z$ is shorter walk from u to v.
Contradicts our assumption that f is the shortest walk.

Distance. $\text{dist}(u, v)$ is length of a shortest path from u to v.

Proposition 3. *For any graph G and vertices $u, v, w \in V(G)$, $\text{dist}(u, w) \leq \text{dist}(u, v) + \text{dist}(v, w)$.*

Suppose f is shortest path from u to v
and g is shortest path from v to w.
$f\,\hat{v}\,g$ — walk from u to w. $|f\,\hat{v}\,g| = |f| + |g|$
$= \text{dist}(u,v) + \text{dist}(v,w)$
$\text{dist}(u,w) \leq |f\,\hat{v}\,g| = \text{dist}(u,v) + \text{dist}(v,w)$

Adjacency Matrix. A graph G with $V(G) = \{v_0, v_1, \ldots v_{n-1}\}$ can be represented by a matrix A_G where$(A_G)_{ij} = 1$ if $(v_i, v_j) \in E(G)$ and is 0 otherwise.

Length k-walk counting matrix. For graph G with vertices $\{v_0, v_1, \ldots v_{n-1}\}$, a length k walk counting matrix is a $n \times n$ matrix C such that C_{ij} = number of length k walks from v_i to v_j. Identity matrix length 0

Theorem 4. *If C is a length k walk counting matrix, and D is a length m walk counting matrix, then CD is a length $k + m$ walk counting matrix.*

$$\begin{matrix} a \\ b \\ c \\ d \end{matrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Adjacency matrix is a length 1 counting matrix.