Graph theory connectivity

Yulia Burkatovskaya
Department of Information Systems and Technologies
Associate professor

2. Connectivity

- Walks
- Distances
- Connectivity of simple graphs
- Connectivity of directed graphs

2.1. Walks

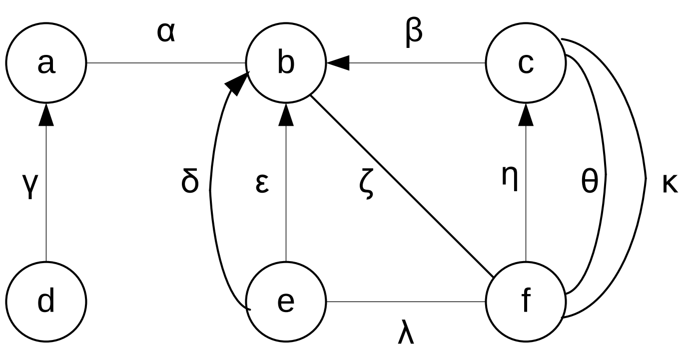
A **walk** is a sequence of vertices and edges

$$<v_0, v_n>=v_0e_1v_1\ldots v_{i-1}e_iv_i\ldots v_{n-1}e_nv_n,$$

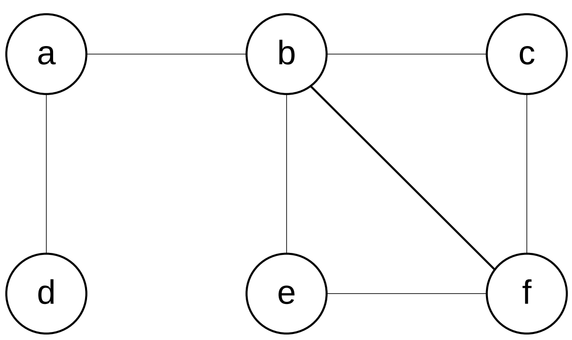
where $e_i=v_{i-1}v_i$. A walk is **closed** if its first and last vertices are the same, and **open** if they are different.

If there are no multiple edges then it is possible to omit edges

Examples.



<a,f> = a α b ζ f η c θ f κ c β b ζ f



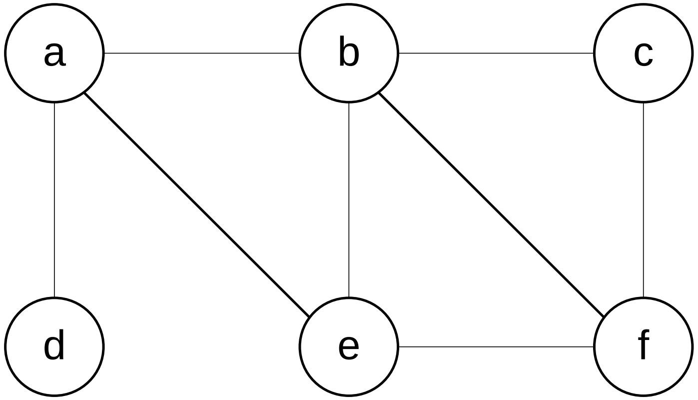
<a,f> = a b f c f c b f

Trail and tour

A **trail** is an open walk in which all the edges are different.

A **tour** (or a **circuit**) is a closed walk in which all the edges are different.

Examples.



Trail <a,f> = a b f c b e f

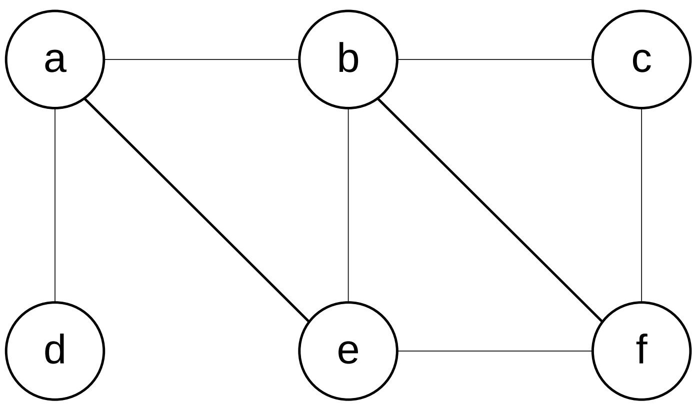
Tour <a,a> = a b f c b e a

Path and cycle

A **path** (or a **chain**) is an open walk in which all the vertices (and hence the edges) are different.

A **cycle** (or a **circuit**) is a closed walk in which all the vertices are distinct.

Examples.



Path <a,f> = a b e f

Cycle <a,a> = a b f e a

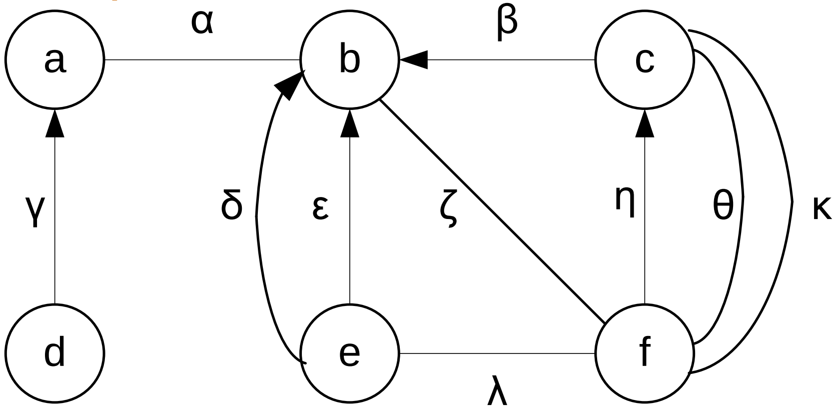
2.2. Distances

The **length** of a walk is the number of edges that it uses.

The **shortest** path <u,v> is a path of minimum length |<u,v>|.

The **distance** between two vertices d(u,v) is the length of a shortest path <u,v>, if one exists, and otherwise the distance is infinity.

Examples.



<a,f> = a α b ζ f η c θ f κ c β b ζ f

|<a,f>| = 7

Shortest path <a,f> = a α b ζ f

d(a,f) = 2

Distances

The **eccentricity** ε(*v*) of a vertex *v* is the maximum distance from *v* to any other vertex.

$$\varepsilon(v) = \max_{u \in V} d(u, v).$$

The **diameter** *D*(*G*) of a graph *G* is the maximum distance between two vertices in a graph or the maximum eccentricity over all vertices in a graph.

$$D(G) = \max_{v \in V} \varepsilon(v) = \max_{u,v \in V} d(u, v).$$

The **radius** *R*(*G*) is the minimum eccentricity over all vertices in a graph.

.

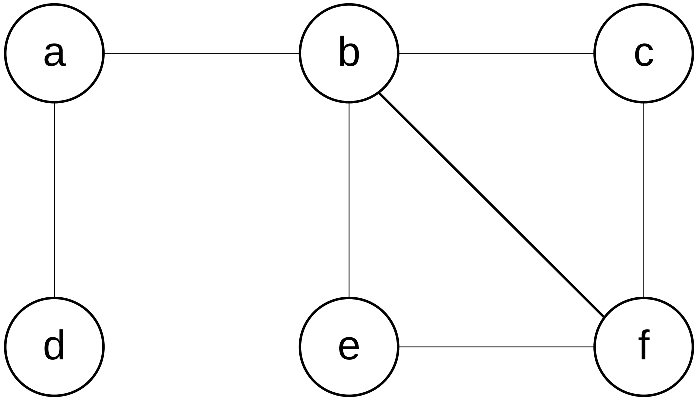
$$R(G) = \min_{v \in V} \varepsilon(v) = \min_{v \in V} \max_{u \in V} d(u, v).$$

Distances

Vertices with maximum eccentricity are called **peripheral vertices**.

Vertices of minimum eccentricity form the **center**.

Examples.



ε(a)=ε(b)=2

ε(c)=ε(d)=ε(e)=ε(f)=3

R(G)=2

D(G)=3

Peripheral vertices c, d, e, f

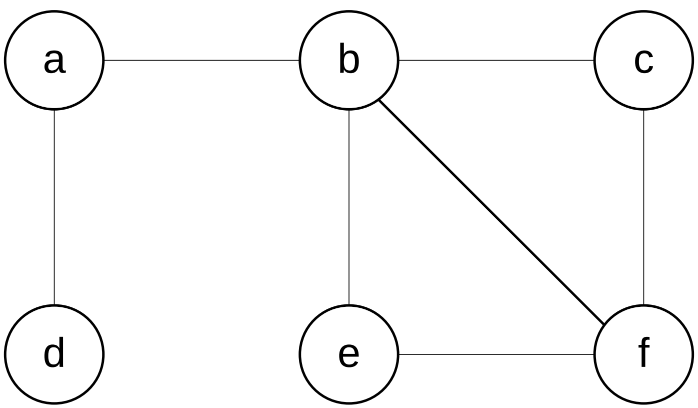
Centre a, b

2.3. Connectivity of simple graphs

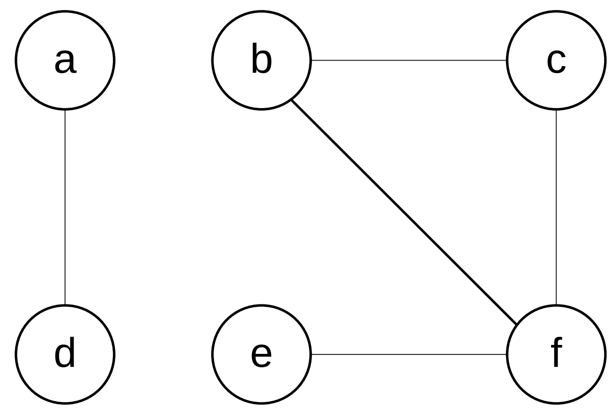
If it is possible to establish a path <u,v> from vertex u to other vertex v, the vertices u and v are **connected**.

If all the pairs of vertices are connected, the graph is said to be **connected**; otherwise, the graph is **disconnected**.

Examples.



Connected graph

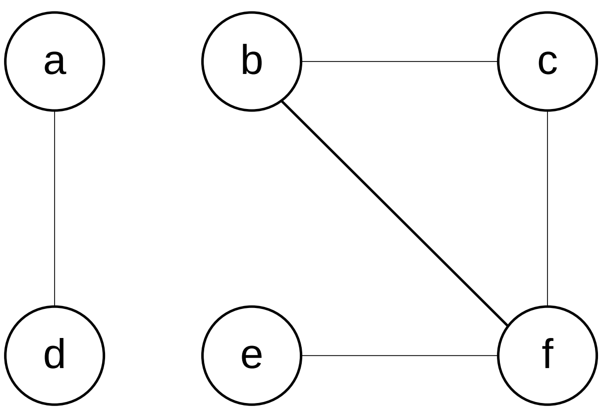


Disconnected graph

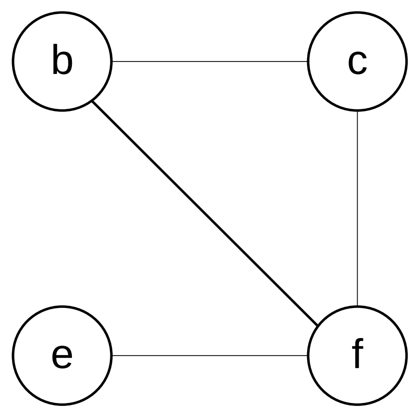
Connected component

A **connected component** of a graph G(V,E) is any its maximally connected subgraph, i.e. an induced subgraph which is not a proper subgraph of any other connected subgraph of G(V,E) .

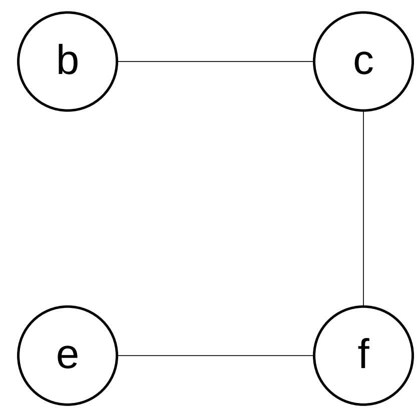
Examples.



Graph with two components



Component



Not a component

Articulation point and bridge

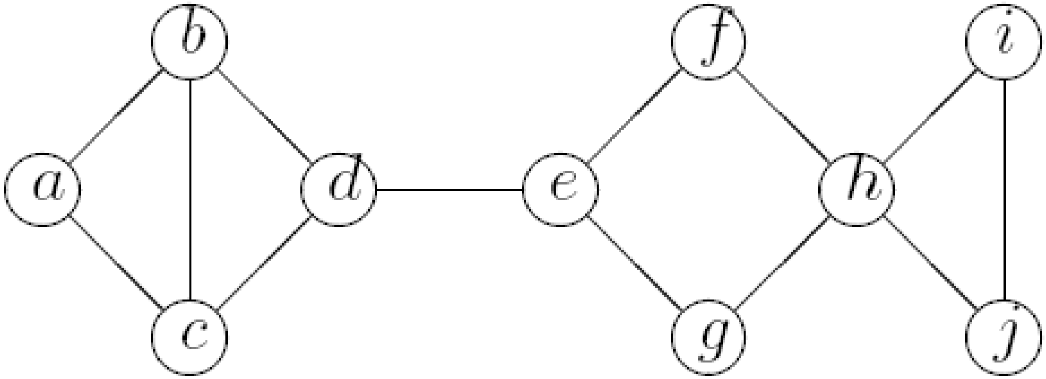
An **articulation point** (or **separating vertex**) of a graph is a vertex whose removal from the graph increases its number of connected components.

A **bridge**, or (**cut edge**) is an analogous edge.

Examples.

de – bridge

d, e, h – articulation points

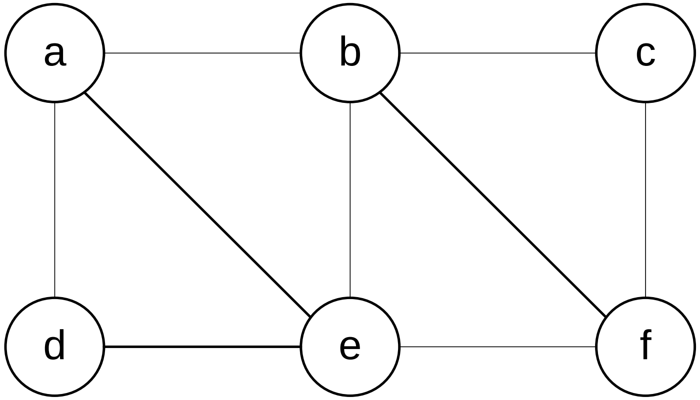


Cuts

A **vertex cut**, (or **separating set**) of a connected graph *G* is a set of vertices whose removal makes *G* disconnected or trivial.

Analogous concept can be defined for edges.

Examples.



{ b, e } – vertex cut

{ ab, be, ef } – edge cut

Graph invariants

k(G) – the number of connected components

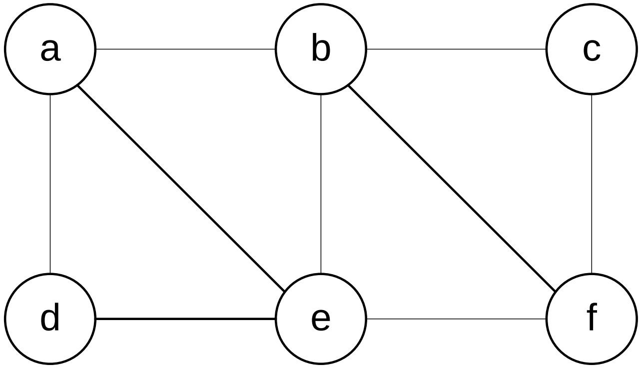
The **vertex connectivity** κ(G) is the size of a minimal vertex cut.

The **edge connectivity** λ(G) is the size of a smallest edge cut.

A graph is called **n-vertex-connected (n-edge-connected)** if its vertex (edge) connectivity is n or greater.

$$\kappa(G) \leq \lambda(G) \leq \delta(G)$$

Examples.



κ = 2 (vertices d, e)

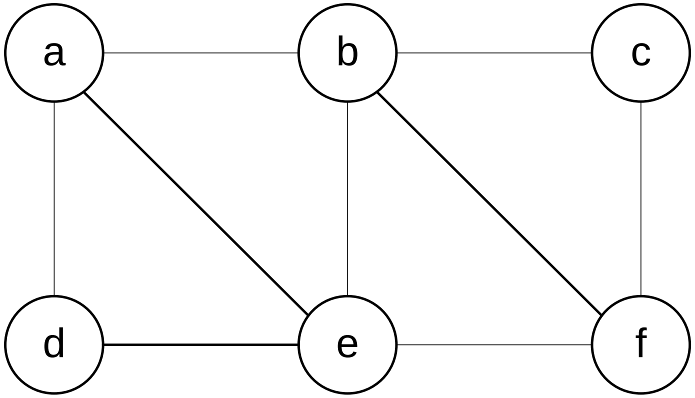
λ = 2 (edges ad, de)

Cuts for a pair of vertices

A **vertex cut S(u,v)**, (or **separating set**) for two connected vertices u and v is a set of vertices whose removal mekes the vertices u and v disconnected.

Analogous concept can be defined for edges.

Examples.



Vertex cut S(a,f)={b,d,e}

Edge cut S(a,f)={ab,ae,ef}

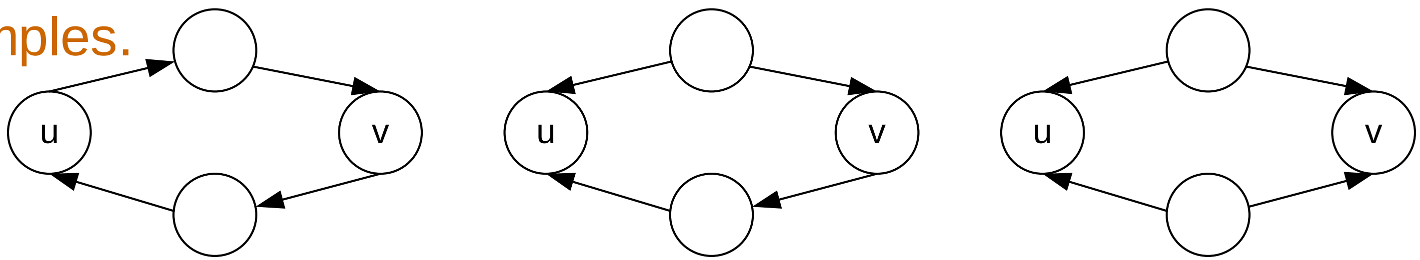
2.4. Connectivity of directed graphs

If it is possible to establish a path <u,v> and a path <v,u> in a digraph, the vertices u and v are **strongly connected**.

If it there exists either a path <u,v> or a path <v,u> in a digraph, the vertices u and v are **unilaterally connected**.

If it there exists a path <u,v> in a graph obtained from a digraph by canceling of edges direction, the vertices u and v are **weakly connected**.

Examples.

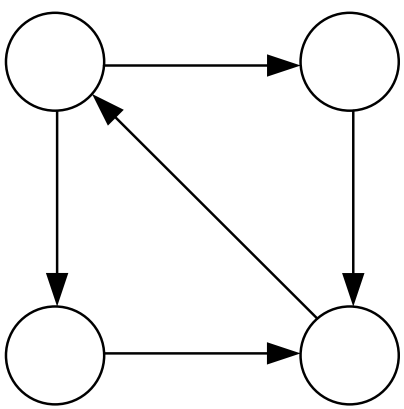


Strongly connected · Unilaterally connected · Weakly connected

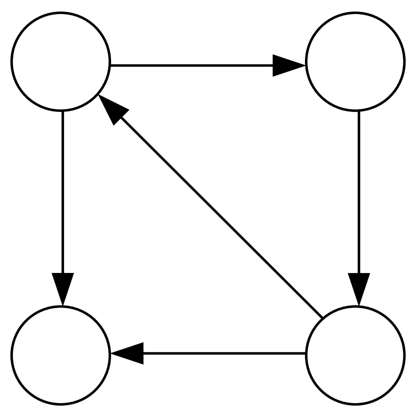
Connectivity of directed graphs

If all the pairs of vertices of a digraph are strongly / unilaterally / weakly connected, the digraph is **strongly / unilaterally / weakly connected**.

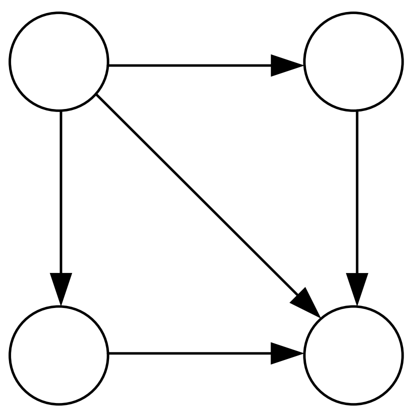
Examples.



Strongly connected



Unilaterally connected

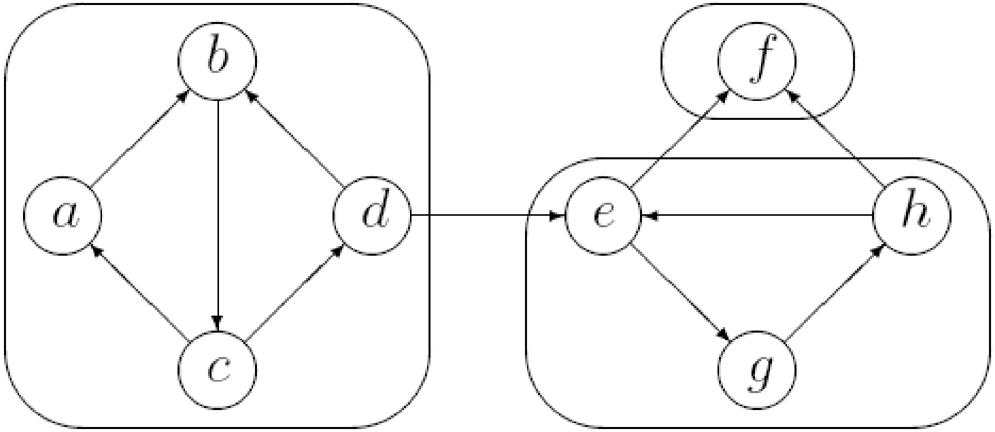


Weakly connected

Strongly connected component

A **strongly connected component** of a digraph G(V,E) is any its maximally strongly connected subgraph, i.e. an induced subgraph which is not a proper subgraph of any other strongly connected subgraph of G(V,E) .

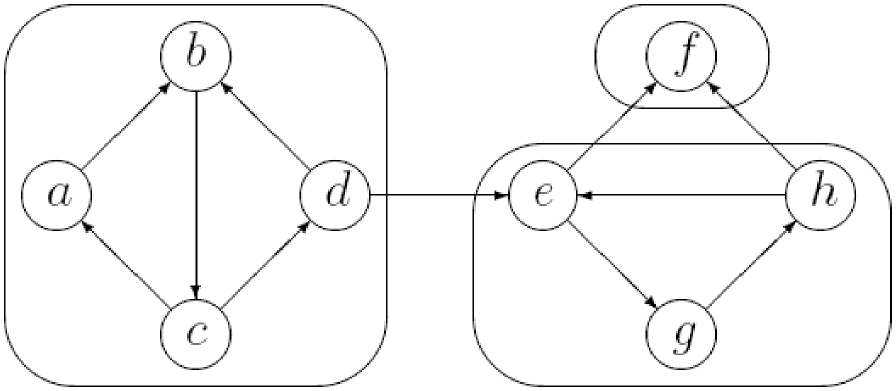
Example.



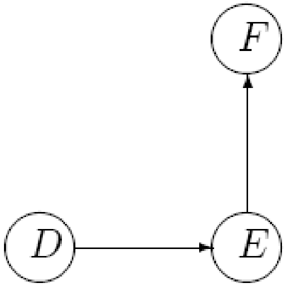
Quotient graph

The **quotient graph** of a digraph D(V,E) with k strongly connected components induced by sets of vertices $V_1,\ldots,V_k$ is a graph D'(V',E') where $V'=\{v_1,\ldots,v_k\}$, $v_iv_j \in E'$ if there is an edge $u_iu_j \in E$: $u_i \in V_i$, $u_j \in V_j$.

Example.



Digraph



Quotient graph

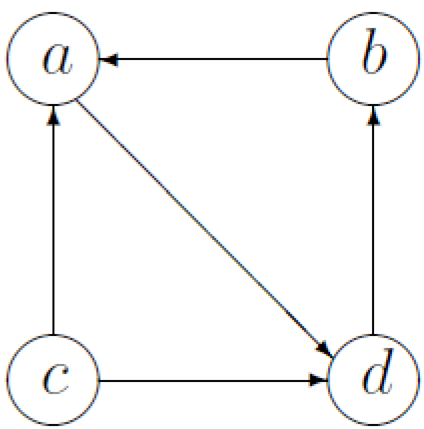
Transitive closure of a graph

Given a directed graph, find out if a vertex *j* is reachable from another vertex *i* for all the vertex pairs {*i*,*j*} in the given graph. Here **reachable** means that there is a path <*i*,*j*>. The results are usually presented in the **reachability matrix *T***, where

Ti,j=1 iff <I,j> exists.

The reachability matrix is called **transitive closure of a graph**.

Example.



$T(D) =$

	a	b	c	d
a	1	1	0	1
b	1	1	0	1
c	1	1	1	1
d	1	1	0	1

Transitive closure of a graph

- Let A:$n \times m$ and B:$n \times m$ be Boolean matrices. Then the **sum** of the matrices is a Boolean matrix C:$n \times m$ defined as follows:

$$C = A \vee B: \; C_{ij} = A_{ij} \vee B_{ij}.$$

- Let A:$n \times k$ and B:$k \times m$ be Boolean matrices. Then the **product** of the matrices is a Boolean matrix C:$n \times m$ defined as follows:

$$C = AB: \; C_{ij} = \bigvee_{l=k}^{m} A_{il} B_{lj}.$$

- Let A:$n \times n$ be a Boolean matrix. Then the ***k*-th power** of the matrix is a Boolean matrix A^k:$n \times n$ defined as follows

$$A^n = \underbrace{AA \ldots A}_{n}$$

Transitive closure of a graph

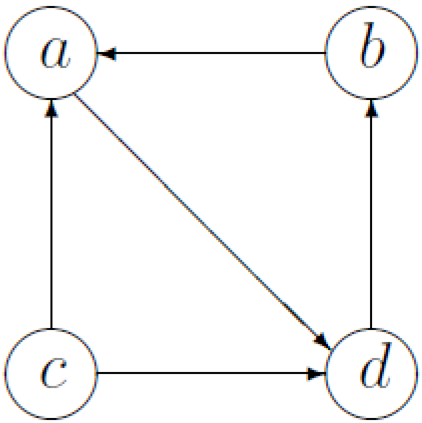
For a graph *G*(*V*,*E*), the matrix M^k, where *M* is the adjacency matrix of the graph *G*, contains some information about paths of the length *k*.

- **Lemma.** For the matrix M^k, element (I,j) is equal to 1 iff there is path <I,j> of the length *k*.
- **Lemma.** The reachability matrix *T* can be calculated as follows:

$$T(D) = E \vee M \vee M^2 \vee \ldots \vee M^{p-1}$$

Transitive closure of a graph

Example.



$$M = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 0 & 0 & 1 \\ b & 1 & 0 & 0 & 0 \\ c & 1 & 0 & 0 & 1 \\ d & 0 & 1 & 0 & 0 \end{array}$$

Transitive closure of a graph

Example.

$$M^2 = MM = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Transitive closure of a graph

Example.

$$M^3 = M^2 M = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transitive closure of a graph

Example.

$$T = E \vee M \vee M^2 \vee M^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \vee \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \vee$$

$$\vee \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \vee \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

Transitive closure of a graph

Warshall algorithm

The main idea of the algorithm is the following: a path <*i*,*j*> exists if and only if:

- *i*=*j*
- ; or there is an edge (*i*,*j*);
- or there is a path <*i*,*j*> going only through vertex 1;
- or there is a path <*i*,*j*> going through vertex 1 and/or 2;
- or there is a path <*i*,*j*> going through vertices 1, 2 and/or 3;
- ...
- or there is a path <*i*,*j*> going through vertices 1,…,$n-1$ and/or *n*.

Transitive closure of a graph

Warshall algorithm

Define the sequence of matrices $B^{(0)},\ldots,B^{(p)}$ as

$$B^{(0)} = E \vee M;$$
$$B_{ij}^{(l)} = B_{ij}^{(l-1)} \vee B_{il}^{(l-1)} B_{lj}^{(l-1)}.$$

Then, $T=B^{(p)}$.

Transitive closure of a graph

Warshall algorithm

Define the sequence of matrices $B^{(0)},\ldots,B^{(p)}$ as

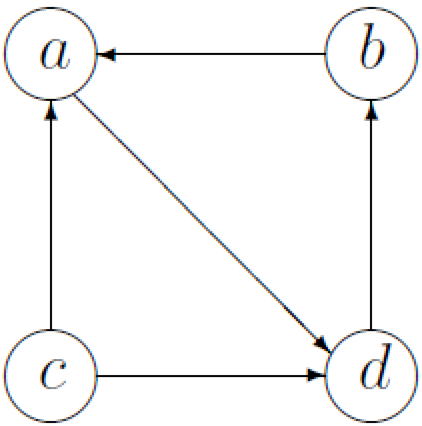
$$B^{(0)} = E \vee M;$$
$$B_{ij}^{(l)} = B_{ij}^{(l-1)} \vee B_{il}^{(l-1)} B_{lj}^{(l-1)}.$$

Then, $T=B^{(p)}$.

$$B^{(l)} = B^{(l-1)} \vee D^{(l)}, \quad D_{ij}^{(l)} = B_{il}^{(l-1)} B_{lj}^{(l-1)}$$

Transitive closure of a graph

Example.



$$M = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 0 & 0 & 1 \\ b & 1 & 0 & 0 & 0 \\ c & 1 & 0 & 0 & 1 \\ d & 0 & 1 & 0 & 0 \end{array}$$

Transitive closure of a graph

Example.

$$B^{(0)} = E \vee M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \vee \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Transitive closure of a graph

Example.

$$B^{(a)} = B^{(0)} \vee D^{(a)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$B^{(b)} = B^{(a)} \vee D^{(b)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \vee \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Transitive closure of a graph

Example.

$$B^{(c)} = B^{(b)} \vee D^{(c)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \vee \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

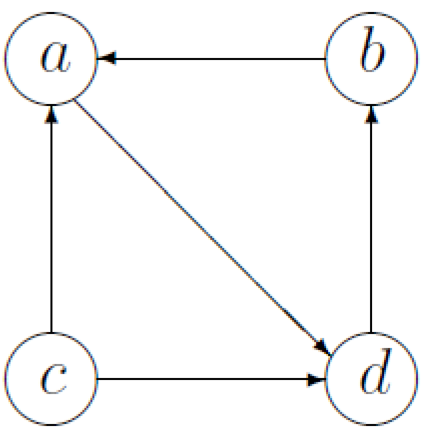
$$B^{(d)} = B^{(c)} \vee D^{(d)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Transitive closure of a graph

Strong connectivity matrix

Given a directed graph, the **strong connectivity matrix *S*** is defined as follows: $S_{i,j}$=1 iff vertices I and j are strongly connected.

Example.



$S =$

	a	b	c	d
a	1	1	0	1
b	1	1	0	1
c	0	0	1	0
d	1	1	0	1

Transitive closure of a graph

Defining strong connected components

- *Start.* All rows of the matrix *S* are unmarked. Set *k*=1.
- *Step 1.* Consider an unmarked row *I* and form *k*-th strongly connected component *Gk* induced by the vertices corresponding the columns containing 1 in row *i*. Mark all the rows containing 1 in column *i*.
- *Step 2*. If there are unmarked rows go to Step 1.
- *End.*

Transitive closure of a graph

Example. For the digraph $D(V,E)$ from the previous examples,

$$S = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 0 & 1 & * \\ b & 1 & 1 & 0 & 1 & * \\ c & 0 & 0 & 1 & 0 \\ d & 1 & 1 & 0 & 1 & * \end{array} \qquad S = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 0 & 1 & * \\ b & 1 & 1 & 0 & 1 & * \\ c & 0 & 0 & 1 & 0 & * \\ d & 1 & 1 & 0 & 1 & * \end{array}$$