# preliminaries

## *graph theory*

Burak Galip ASLAN, PhD

# graphs

generally two types of graphs are discerned:

- *undirected graph*
- *directed graph*

# undirected graph

***undirected graph*** G is a pair $G=\mathbf{(}V(G), E(G)\mathbf{)}$ where V(G) is a finite set of **vertices** (a.k.a. **nodes**) and E(G) is a set of <u>unordered</u> pairs $(v_i, v_j)$ where $\mathbf{(}v_i, v_j \in V(G)\mathbf{)}$ called **edges**.

# directed graph

***directed graph*** G is a pair $G=\big(V(G), A(G)\big)$ where V(G) is a finite set of **vertices** (a.k.a. **nodes**) and A(G) is a set of ordered pairs $(v_i, v_j)$ where $\big(v_i, v_j \in V(G)\big)$ called **arcs**.

an arc $(v_i,v_j)$ is often written as;
$v_i \longrightarrow v_j$ or $v_j \longleftarrow v_i$

# predecessors in directed graphs

in a ***digraph*** G, vertex $v_j$ is called a **predecessor** (**parent**) of vertex $v_i$ if $(v_j, v_i) \in A(G)$

all predecessors of vertex $v_i$ in G is denoted with $P_G(v_i)$

# successors in directed graphs

in a ***digraph*** G, vertex $v_j$ is called a **successor** (**child**) of vertex $v_i$ if $(v_i,v_j) \in A(G)$

all successors of vertex $v_i$ in G is denoted with $\sigma_G(v_i)$

# neighborhood

the set of neighbors of a vertex $V_i$ is shown as;

$v_g(V_i) = \sigma_G(V_i) \cup P_G(V_i)$ *(if G is **directed**)*

$v_g(V_i) = V_j \mid \{V_i, V_j\} \in E(G)$ *(if G is **undirected**)*

# neighborhood

the set of neighbors of a vertex $V_i$ is shown as;

$v_g(V_i) = \sigma_G(V_i) \cup P_G(V_i)$ *(if G is **directed**)*

$v_g(V_i) = V_j \mid \{V_i, V_j\} \in E(G)$ *(if G is **undirected**)*

the size of neighbor set of a vertex is called its ***degree***. For a **digraph** G;

***in-degree*** ***:*** # of predecessors of a vertex

***out-degree*** ***:*** # of successors of a vertex

***self-loop*** *adds +2 to degree in undirected graphs*

*adds +1 to in-degree and +1 to out-degree in directed graphs*

# neighborhood

the set of neighbors of a vertex $V_i$ is shown as;

$v_g(V_i) = \sigma_G(V_i) \cup P_G(V_i)$ *(if G is **directed**)*

$v_g(V_i) = V_j \mid \{V_i, V_j\} \in E(G)$ *(if G is **undirected**)*

the size of neighbor set of a vertex is called its ***degree***. For a **digraph** G;

***in-degree*** ***:*** # of predecessors of a vertex

***out-degree*** ***:*** # of successors of a vertex

the consideration of incoming and outgoing arcs together are called the ***incident arcs***.

# walks

a ***walk*** is any route through an **undirected graph** from vertex to vertex along edges.

a ***path*** is a walk that does not include any vertex twice except that its first vertex might be the same as its last.

a path from $V_0$ to $V_i$ is a sequence of vertices $V_0,\ldots,V_k$ where $k \geq 0$ with distinct edges;
and $(V_{i-1},V_i) \in E(G)$ where $i=1,\ldots,k$
and ***k*** is called the **length** of a path.

a path is called ***simple path*** if all vertices are distinct.

# walks

a ***cycle*** is a path that begins and ends on the same vertex with non-zero length.

a ***trail*** is a walk that does not pass over the same edge twice.

a ***circuit*** is a trail that begins and ends on the same vertex.

a graph G is called **cyclic** if it contains at least 1 cycle, else it is called **acyclic**.

self-loops are generally not allowed in undirected graphs e.g. an edge $(v_0,v_0)$ is not allowed

# walks in digraphs

the concepts of **path** and **cycle** introduced for undirected graphs directly apply to directed graphs by considering **arcs** rather than **edges**, ***however paths in digraphs can travel arcs only in the direction of arrows***.

# underlying graph of a digraph

*underlying graph* concept associates an undirected graph with a directed one

let *G*=$\boldsymbol{(}V(G),A(G)\boldsymbol{)}$ *be a digraph*

*the underlying graph H of G is the undirected graph H*=$\boldsymbol{(}V(H),E(H)\boldsymbol{)}$
*where* $V(H) = V(G)$ *and*
$E(H)$ *is obtained from* $A(G)$ *by replacing each arc* $(v_i,v_j) \in A(G)$ *by an edge* $(v_i,v_j)$

# underlying graph of a digraph

let **G** be a digraph and

let **H** be the underlying graph of G

a **chain** from $V_0$ to $V_k$ in G is a sequence of vertices $V_0, \ldots, V_k$ where $k \geq 0$ and *the length of the chain is* ***k*** **(***(# of vertices visited) – 1***)**.

<u>this chain is a path</u> in the underlying graph of H of G

if this chain doesn't end in itself it is called a **simple chain**

a **loop** in G is a sequence of vertices that is a *cycle* in the underlying graph H of G

# connectivity in digraphs

a digraph G is **connected** if there exists at least one chain between any two vertices in G; otherwise it is called **unconnected**

# connectivity in digraphs

a digraph G is **connected** if there exists at least one chain between any two vertices in G; otherwise it is called **unconnected**

a digraph G is called **singly connected** if it does not contain any loops; otherwise it is called **multiply connected**

# connectivity in digraphs

a digraph G is **connected** if there exists at least one chain between any two vertices in G; otherwise it is called **unconnected**

a digraph G is called **singly connected** if it does not contain any loops; otherwise it is called **multiply connected**

hence in a **singly connected** graph, there is **at most one chain between any two vertices**; this property does not hold for multiply connected digraphs

# connectivity in digraphs

a digraph G is **connected** if there exists at least one chain between any two vertices in G; otherwise it is called **unconnected**

a digraph G is called **singly connected** if it does not contain any loops; otherwise it is called **multiply connected**

hence in a **singly connected** graph, there is **at most one chain between any two vertices**; this property does not hold for multiply connected digraphs

a singly connected digraph is called a **directed tree** if each vertex in G has **at most one predecessor**

# connectivity in digraphs

a digraph is **acyclic** if it contains no directed cycles

however an acyclic digraph (DAG) can contain loops

# subgraphs

let $G=\big(V(G),E(G)\big)$ be an undirected graph

the **subgraph** $H$ induced by $V \subseteq V(G)$ is the undirected graph $H=\big(V, (V \times V) \cap E(G)\big)$ where any different combinations of edges are existent **from that subset of vertices**.

# subgraphs

let $G=\big(V(G),E(G)\big)$ be an undirected graph

the **subgraph** H induced by $V \subseteq V(G)$ is the undirected graph $H=\big(V, (V\mathrm{x}V) \cap E(G)\big)$ where any different combinations of edges are existent **from that subset of vertices**.

the **induced (*generated*) subgraph** H must contain all combinations of edges existent **from that subset of vertices**.

# references

Probabilistic Reasoning

(Probabilistisch Redeneren)

*authors: Linda van der Gaag*
*Silja Renooij*

Fall 2017

Universiteit Utrecht

WIKIPEDIA

Google™