

# Discrete Mathematics in Computer Science

## Acyclic (Di-) Graphs

Malte Helmert, Gabriele Röger

University of Basel

# Acyclic

Similarly to connectedness, the presence or absence of **cycles** is an important practical property for (di-) graphs.

## Definition (acyclic, forest, DAG)

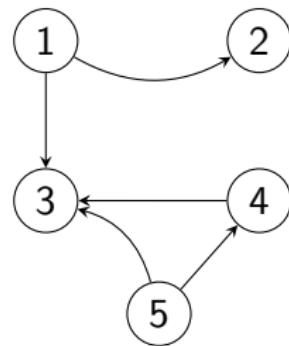
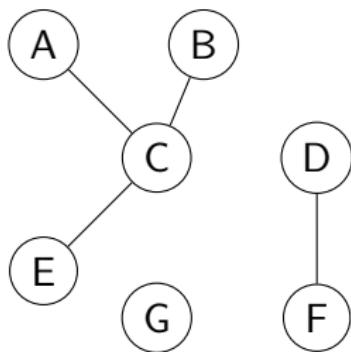
A graph or digraph  $G$  is called **acyclic** if there exists no cycle in  $G$ .

An acyclic graph is also called a **forest**.

An acyclic digraph is also called a **DAG** (directed acyclic graph).

German: azyklisch/kreisfrei, Wald, DAG

## Acyclic (Di-) Graphs – Example



# Trees

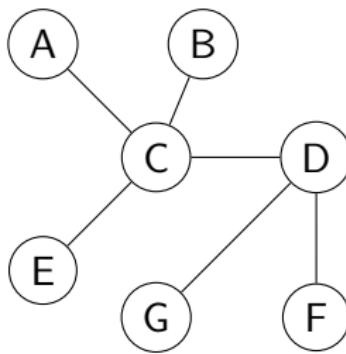
## Definition (tree)

A connected forest is called a **tree**.

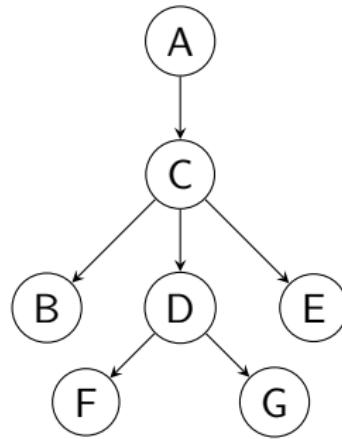
**German:** Baum

- **Tree** is also a word for a recursive data structure, which consists of either a **leaf** or a **parent node** with one or more **children**, which are themselves trees.
- This other kind of tree is also called a **rooted tree** to distinguish it from a tree as a graph.
- The two meanings of “tree” are distinct but closely related.

## Tree Graphs vs. Rooted Trees – Example (1)

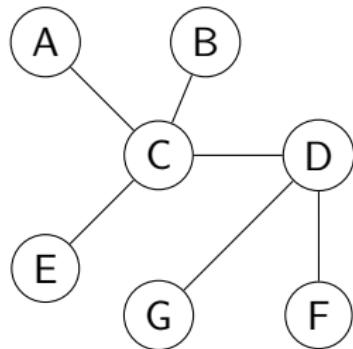


tree graph

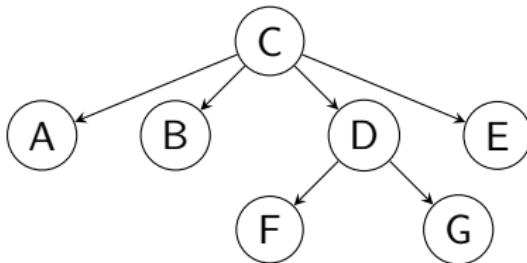


rooted tree with root A

## Tree Graphs vs. Rooted Trees – Example (2)

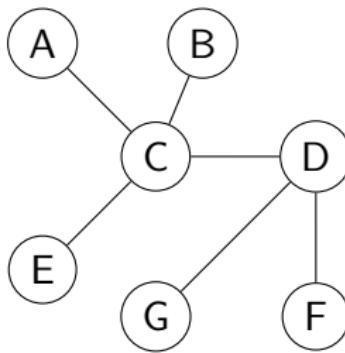


tree graph

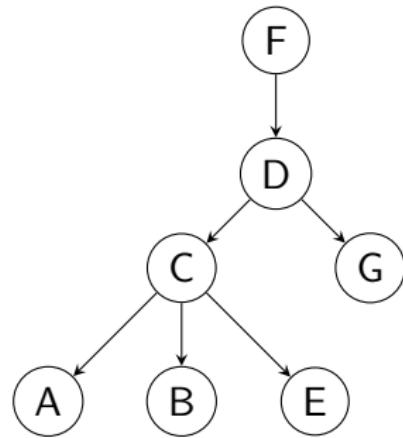


rooted tree with root *C*

## Tree Graphs vs. Rooted Trees – Example (3)



tree graph



rooted tree with root *F*

# From Tree Graphs to Rooted Trees

General procedure for converting tree graphs into rooted trees:

- Select any vertex  $v$ . Make  $v$  the root of the tree.
- Initially,  $v$  is the only **pending** vertex,  
and there are no **processed** vertices.
- As long as there are pending vertices:
  - Select any pending vertex  $u$ .
  - Make all neighbours  $v$  of  $u$  that are not yet processed  
children of  $u$  and mark them as pending.
  - Change  $u$  from pending to processed.

We do not prove that this procedure always works. A proof of correctness can be given based on the results we show next.

# Discrete Mathematics in Computer Science

## Unique Paths in Trees

Malte Helmert, Gabriele Röger

University of Basel

# Unique Paths in Trees

## Theorem

*Let  $G = (V, E)$  be a graph.*

*Then  $G$  is a tree iff there exists exactly one path from any vertex  $u \in V$  to any vertex  $v \in V$ .*

## Unique Paths In Trees – Proof (1)

Proof.

( $\Rightarrow$ ):  $G$  is a tree. Let  $u, v \in V$ .

We must show that there exists exactly one path from  $u$  to  $v$ .

We know that at least one path exists because  $G$  is connected.

It remains to show that there cannot be two paths from  $u$  to  $v$ .

If  $u = v$ , there is only one path (the empty one).

(Any longer path would have to repeat a vertex.)

We assume that there exist two different paths from  $u$  to  $v$

( $u \neq v$ ) and derive a contradiction.

...

## Unique Paths In Trees – Proof (2)

Proof (continued).

Let  $\pi = \langle v_0, v_1, \dots, v_n \rangle$  and  $\pi' = \langle v'_0, v'_1, \dots, v'_m \rangle$  be the two paths (with  $v_0 = v'_0 = u$  and  $v_n = v'_m = v$ ).

Let  $i$  be the smallest index with  $v_i \neq v'_i$ , which must exist because the two paths are different, and neither can be a prefix of the other (else  $v$  would be repeated in the longer path).

We have  $i \geq 1$  because  $v_0 = v'_0$ .

Let  $j \geq i$  be the smallest index such that  $v_j = v'_k$  for some  $k \geq i$ .

Such an index must exist because  $v_n = v'_m$ .

Then  $\langle v_{i-1}, \dots, v_{j-1}, v'_k, \dots, v'_{i-1} \rangle$  is a cycle, which contradicts the requirement that  $G$  is a tree.

...

## Unique Paths In Trees – Proof (3)

Proof (continued).

( $\Leftarrow$ ): For all  $u, v \in V$ , there exists exactly one path from  $u$  to  $v$ . We must show that  $G$  is a tree, i.e., is connected and acyclic.

Because there exist paths from all  $u$  to all  $v$ ,  $G$  is connected.

Proof by contradiction: assume that there exists a cycle in  $G$ ,  $\pi = \langle u, v_1, \dots, v_n, u \rangle$  with  $n \geq 2$ .

(Note that all cycles have length at least 3.)

From the definition of cycles, we have  $v_1 \neq v_n$ .

Then  $\langle u, v_1 \rangle$  and  $\langle u, v_n, \dots, v_1 \rangle$  are two different paths from  $u$  to  $v_1$ , contradicting that there exists exactly one path from every vertex to every vertex. Hence  $G$  must be acyclic. □

# Discrete Mathematics in Computer Science

## Leaves and Edge Counts in Trees and Forests

Malte Helmert, Gabriele Röger

University of Basel

# Leaves in Trees

## Definition

Let  $G = (V, E)$  be a tree.

A **leaf** of  $G$  is a vertex  $v \in V$  with  $\deg(v) = 1$ .

## Theorem

Let  $G = (V, E)$  be a tree with  $|V| \geq 2$ .

Then  $G$  has at least two leaves.

## Leaves in Trees – Proof

### Proof.

Let  $\pi = \langle v_0, \dots, v_n \rangle$  be path in  $G$  with maximal length among all paths in  $G$ .

Because  $|V| \geq 2$ , we have  $n \geq 1$  (else  $G$  would not be connected).

We show that vertex  $v_n$  has degree 1:  $v_{n-1}$  is a neighbour in  $G$ .

Assume that it were not the only neighbour of  $v_n$  in  $G$ , so  $u$  is another neighbour of  $v_n$ . Then:

- If  $u$  is not on the path, then  $\langle v_0, \dots, v_n, u \rangle$  is a longer path: contradiction.
- If  $u$  is on the path, then  $u = v_i$  for some  $i \neq n$  and  $i \neq n - 1$ . Then  $\langle v_i, \dots, v_n, v_i \rangle$  is a cycle: contradiction.

By reversing  $\pi$  we can show  $\deg(v_0) = 1$  in the same way. □

## Edges in Trees

### Theorem

*Let  $G = (V, E)$  be a tree with  $V \neq \emptyset$ .  
Then  $|E| = |V| - 1$ .*

## Edges in Trees – Proof (1)

Proof.

Proof by induction over  $n = |V|$ .

## Edges in Trees – Proof (1)

Proof.

Proof by induction over  $n = |V|$ .

Induction base ( $n = 1$ ):

Then  $G$  has 1 vertex and 0 edges.

We get  $|E| = 0 = 1 - 1 = |V| - 1$ .

# Edges in Trees – Proof (1)

## Proof.

Proof by induction over  $n = |V|$ .

Induction base ( $n = 1$ ):

Then  $G$  has 1 vertex and 0 edges.

We get  $|E| = 0 = 1 - 1 = |V| - 1$ .

Induction step ( $n \rightarrow n + 1$ ):

Let  $G = (V, E)$  be a tree with  $n + 1$  vertices ( $n \geq 1$ ).

From the previous result,  $G$  has a leaf  $u$ .

Let  $v$  be the only neighbour of  $u$ .

Let  $e = \{u, v\}$  be the connecting edge.

...

## Edges in Trees – Proof (2)

Proof (continued).

Consider the graph  $G' = (V', E')$   
with  $V' = V \setminus \{u\}$  and  $E' = E \setminus \{e\}$ .

- $G'$  is acyclic: every cycle in  $G'$  would also be present in  $G$  (contradiction).
- $G'$  is connected: for all vertices  $w \neq u$  and  $w' \neq u$ ,  $G$  has a path  $\pi$  from  $w$  to  $w'$  because  $G$  is connected.  
Path  $\pi$  cannot include  $u$  because  $u$  has only one neighbour, so traversing  $u$  requires repeating  $v$ . Hence  $\pi$  is also a path in  $G'$ .

Hence  $G'$  is a tree with  $n$  vertices, and we can apply the induction hypothesis, which gives  $|E'| = |V'| - 1$ .

It follows that

$$|E| = |E'| + 1 = (|V'| - 1) + 1 = (|V'| + 1) - 1 = |V| - 1.$$

□

## Edges in Forests

### Theorem

*Let  $G = (V, E)$  be a forest.*

*Let  $C$  be the set of connected components of  $G$ .*

*Then  $|E| = |V| - |C|$ .*

This result generalizes the previous one.

## Edges in Forests – Proof

Proof.

Let  $C = \{C_1, \dots, C_k\}$ .

For  $1 \leq i \leq k$ , let  $G_i = (C_i, E_i)$  be  $G$  restricted to  $C_i$ , i.e., the graph whose vertices are  $C_i$

and whose edges are the edges  $e \in E$  with  $e \subseteq C_i$ .

We have  $|V| = \sum_{i=1}^k |C_i|$  because the connected components form a partition of  $V$ .

We have  $|E| = \sum_{i=1}^k |E_i|$  because every edge belongs to exactly one connected component. (Note that there cannot be edges between different connected components.)

Every graph  $G_i$  is a tree with at least one vertex: it is connected because its vertices form a connected component, and it is acyclic because  $G$  is. This implies  $|E_i| = |C_i| - 1$ .

Putting this together, we get

$$|E| = \sum_{i=1}^k |E_i| = \sum_{i=1}^k (|C_i| - 1) = \sum_{i=1}^k |C_i| - k = |V| - |C|. \quad \square$$

# Discrete Mathematics in Computer Science

## Characterizations of Trees

Malte Helmert, Gabriele Röger

University of Basel

# Characterizations of Trees

## Theorem

Let  $G = (V, E)$  be a graph with  $V \neq \emptyset$ .

The following statements are equivalent:

- ①  $G$  is a tree.
- ②  $G$  is acyclic and connected.
- ③  $G$  is acyclic and  $|E| = |V| - 1$ .
- ④  $G$  is connected and  $|E| = |V| - 1$ .
- ⑤ For all  $u, v \in V$  there exists exactly one path from  $u$  to  $v$ .

# Characterizations of Trees – Proof (1)

Reminder:

- (1)  $G$  is a tree.
- (2)  $G$  is acyclic and connected.
- (3)  $G$  is acyclic and  $|E| = |V| - 1$ .
- (4)  $G$  is connected and  $|E| = |V| - 1$ .
- (5) For all  $u, v \in V$  there exists exactly one path from  $u$  to  $v$ .

Proof.

We know already:

- (1) and (2) are equivalent by definition of trees.
- We have shown that (1) and (5) are equivalent.
- We have shown that (1) implies (3) and (4).

We complete the proof by showing (3)  $\Rightarrow$  (2) and (4)  $\Rightarrow$  (2). . .

## Characterizations of Trees – Proof (2)

Reminder:

- (2)  $G$  is acyclic and connected.
- (3)  $G$  is acyclic and  $|E| = |V| - 1$ .

Proof (continued).

(3)  $\Rightarrow$  (2):

Because  $G$  is acyclic, it is a forest.

From the previous result, we have  $|E| = |V| - |C|$ ,  
where  $C$  are the connected components of  $G$ .

## Characterizations of Trees – Proof (2)

Reminder:

- (2)  $G$  is acyclic and connected.
- (3)  $G$  is acyclic and  $|E| = |V| - 1$ .

Proof (continued).

(3)  $\Rightarrow$  (2):

Because  $G$  is acyclic, it is a forest.

From the previous result, we have  $|E| = |V| - |C|$ ,  
where  $C$  are the connected components of  $G$ .

But we also know  $|E| = |V| - 1$ . This implies  $|C| = 1$ .

Hence  $G$  is connected and therefore a tree.

...

## Characterizations of Trees – Proof (3)

Reminder:

- (2)  $G$  is acyclic and connected.
- (4)  $G$  is connected and  $|E| = |V| - 1$ .

Proof (continued).

(4)  $\Rightarrow$  (2):

In graphs that are not acyclic, we can remove an edge without changing the connected components: if  $\langle v_0, \dots, v_n, v_0 \rangle$  ( $n \geq 2$ ) is a cycle, remove the edge  $\{v_0, v_1\}$  from the graph.

Every walk using this edge can substitute  $\langle v_1, \dots, v_n, v_0 \rangle$  (or the reverse path) for it.

## Characterizations of Trees – Proof (3)

Reminder:

- (2)  $G$  is acyclic and connected.
- (4)  $G$  is connected and  $|E| = |V| - 1$ .

Proof (continued).

(4)  $\Rightarrow$  (2):

In graphs that are not acyclic, we can remove an edge without changing the connected components: if  $\langle v_0, \dots, v_n, v_0 \rangle$  ( $n \geq 2$ ) is a cycle, remove the edge  $\{v_0, v_1\}$  from the graph.

Every walk using this edge can substitute  $\langle v_1, \dots, v_n, v_0 \rangle$  (or the reverse path) for it.

Iteratively remove edges from  $G$  in this way while preserving connectedness until this is no longer possible. The resulting graph  $(V, E')$  is acyclic and connected and therefore a tree.

## Characterizations of Trees – Proof (3)

Reminder:

- (2)  $G$  is acyclic and connected.
- (4)  $G$  is connected and  $|E| = |V| - 1$ .

Proof (continued).

(4)  $\Rightarrow$  (2):

In graphs that are not acyclic, we can remove an edge without changing the connected components: if  $\langle v_0, \dots, v_n, v_0 \rangle$  ( $n \geq 2$ ) is a cycle, remove the edge  $\{v_0, v_1\}$  from the graph.

Every walk using this edge can substitute  $\langle v_1, \dots, v_n, v_0 \rangle$  (or the reverse path) for it.

Iteratively remove edges from  $G$  in this way while preserving connectedness until this is no longer possible. The resulting graph  $(V, E')$  is acyclic and connected and therefore a tree.

This implies  $|E'| = |V| - 1$ , but we also have  $|E| = |V| - 1$ .

This yields  $|E| = |E'|$  and hence  $E' = E$ : the number of edges removable in this way must be 0. Hence  $G$  is already acyclic. □