universität innsbruck

Discrete structures

Cezary Kaliszyk Raoul Schikora
Vincent van Oostrom

http://cl-informatik.uibk.ac.at

Summary last week

- Hasse diagram of a partial order $\leq$ or strict order $<$
- least irreflexive, atransitive subrelation R of $\leq$ such that $\leq = R^*$ or $< = R^+$ (atransitive: $x \mathrel{R} y$ and $y \mathrel{R} z$ then not $x \mathrel{R} z$)
- for total orders, minimal = least and maximal = greatest
- finite partial orders have minimal and maximal elements
- the lexicographic order $<_{\text{lex}}$ on words; partial/total if $\leq$ is.

Summary last week

- Hasse diagram of a partial order $\leq$ or strict order $<$
- least irreflexive, atransitive subrelation R of $\leq$ such that $\leq = R^*$ or $< = R^+$ (atransitive: $x\ R\ y$ and $y\ R\ z$ then not $x\ R\ z$)
- for total orders, minimal = least and maximal = greatest
- finite partial orders have minimal and maximal elements
- the lexicographic order $<_{\text{lex}}$ on words; partial/total if $\leq$ is.
- well-founded relations as not having infinite descending chains
- Three methods to prove that all elements of set have some property:

1) by cases; for finite sets, enumerating all elts

2) by universal generalisation; for infinite sets, proving for some arbitrary elt

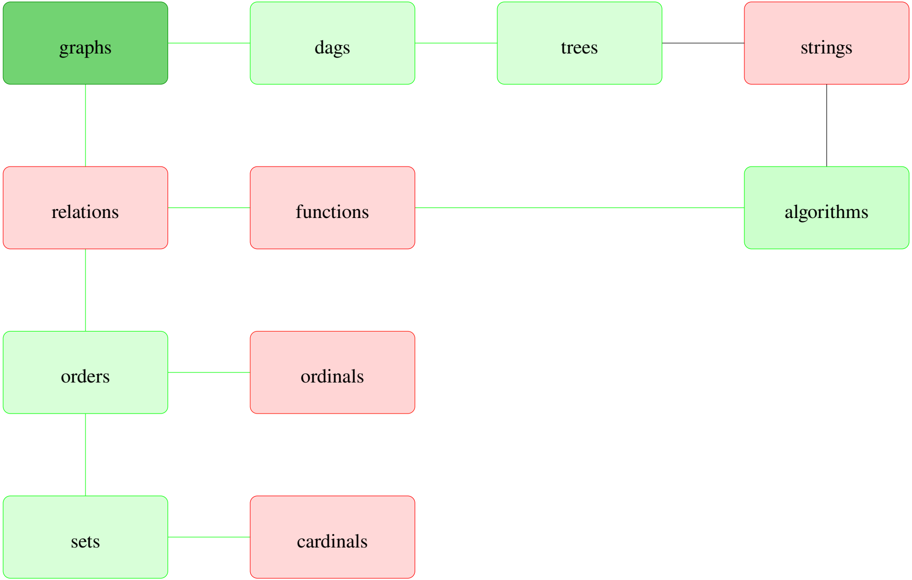
3) by well-founded induction; for infinite sets, using property (IH) for smaller elts

- well-founded induction principle for well-founded relation R, property P: $\forall n.((\forall m \text{ such that } m\ R\ n.P(m)) \rightarrow P(n)) \rightarrow (\forall n.P(n))$

Course themes

- directed and undirected graphs
- relations and functions
- orders and induction
- trees and dags
- finite and infinite counting
- elementary number theory
- Turing machines, algorithms, and complexity
- decidable and undecidable problem

Discrete structures



Well-founded relations

Definition (well-founded relation)

- Let R be a relation on a set M
- A sequence $(x_0, x_1, x_2, \ldots)$ of elements of M is an infinite descending R-chain, if
$$\ldots\ R\ x_2\ R\ x_1\ R\ x_0$$
- R is well-founded, if M has no infinite descending R-chains.
- When we say that partial order $\leq$ is well-founded we mean that its strict part $<$ is

Well-founded relations

Definition (well-founded relation)

- Let R be a relation on a set M
- A sequence $(x_0, x_1, x_2, \ldots)$ of elements of M is an infinite descending R-chain, if
$$\ldots R\ x_2\ R\ x_1\ R\ x_0$$
- R is well-founded, if M has no infinite descending R-chains.
- When we say that partial order $\leq$ is well-founded we mean that its strict part $<$ is

Principle of well-founded induction

Assumption: R a well-founded relation on set N. P a property of $n \in N$.
Induction: for arbitrary $n \in N$, show that if $P(m)$ for all m such that $m\ R\ n$, then $P(n)$
Conclude: for all $n \in N$, $P(n)$

Well-founded relations

Definition (well-founded relation)

- Let R be a relation on a set M
- A sequence $(x_0, x_1, x_2, \ldots)$ of elements of M is an infinite descending R-chain, if

$$\ldots R\ x_2\ R\ x_1\ R\ x_0$$

- R is well-founded, if M has no infinite descending R-chains.
- When we say that partial order $\leq$ is well-founded we mean that its strict part $<$ is

Principle of mathematical induction

Assumption: R the well-founded relation $\{(n, n+1) \mid n \in \mathbb{N}\}$. P a property of $n \in \mathbb{N}$.
Induction: for arbitrary $n \in \mathbb{N}$, show that if $P(m)$ for all m such that $m\ R\ n$, then $P(n)$
Conclude: for all $n \in \mathbb{N}$, $P(n)$

Well-founded relations

Definition (well-founded relation)

- Let R be a relation on a set M
- A sequence $(x_0, x_1, x_2, \ldots)$ of elements of M is an infinite descending R-chain, if

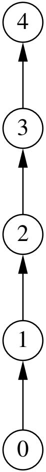
$$\ldots R\ x_2\ R\ x_1\ R\ x_0$$

- R is well-founded, if M has no infinite descending R-chains.
- When we say that partial order $\leq$ is well-founded we mean that its strict part $<$ is

Principle of mathematical induction

Assumption: R the well-founded relation $\{(n, n+1) \mid n \in \mathbb{N}\}$. P a property of $n \in \mathbb{N}$.
Induction: for arbitrary $n \in \mathbb{N}$, show that $P(0)$ and if $n > 0$ and $P(n-1)$ then $P(n)$
Conclude: for all $n \in \mathbb{N}$, $P(n)$

Mathematical induction

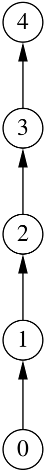


Induction basis: P(0)

Induction step: if n > 0 and P(n−1) then P(n)

P(3)?

Mathematical induction



Induction basis: P(0)

Induction step: if n > 0 and P(n−1) then P(n)

P(3)?

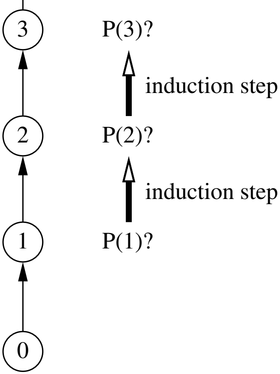
induction step

P(2)?

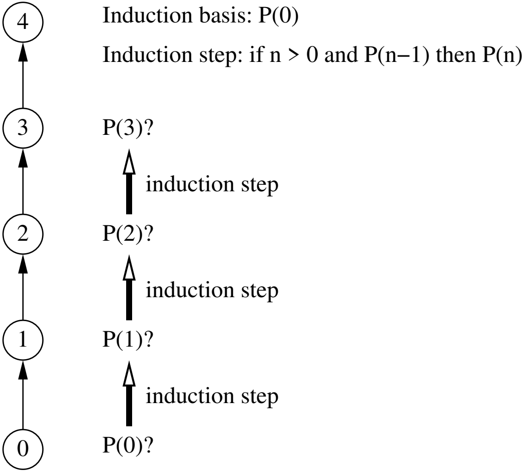
Mathematical induction

Induction basis: P(0)

Induction step: if n > 0 and P(n−1) then P(n)



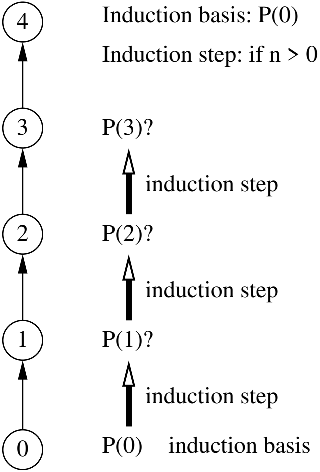
Mathematical induction



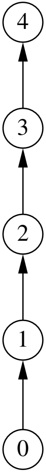
Mathematical induction

Induction basis: P(0)

Induction step: if n > 0 and P(n−1) then P(n)



Mathematical induction



Induction basis: P(0)

Induction step: if n > 0 and P(n−1) then P(n)

P(3)?

induction step

P(2)?

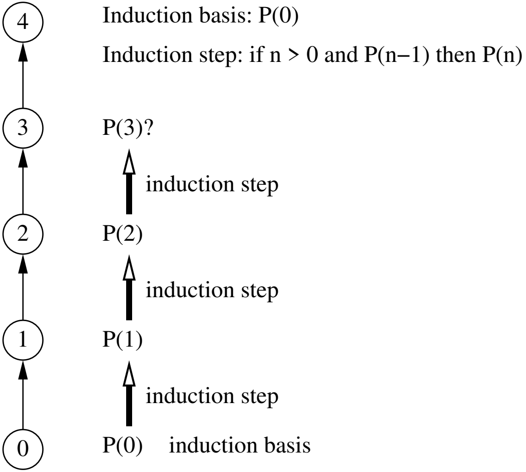
induction step

P(1)

induction step

P(0) induction basis

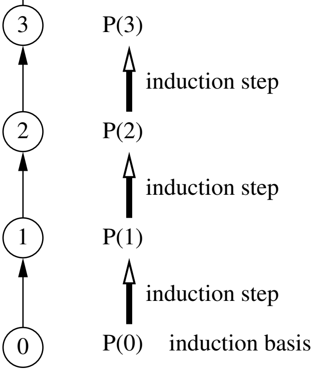
Mathematical induction



Mathematical induction

Induction basis: P(0)

Induction step: if n > 0 and P(n−1) then P(n)



When and how to apply induction?

When

Prove a property P of elements of an infinite set N.

When and how to apply induction?

When

Prove a property P of elements of an infinite set N.

How

Try to see whether, when proving $P(n)$ for an arbitrary element $n \in N$, it may be helpful to know that $P(m)$ holds already for m smaller than n, for an appropriate notion of smaller. For instance:

- on natural numbers: less-than, successor ('sub-number')
- on positive natural numbers: divisibility

When and how to apply induction?

When

Prove a property P of elements of an infinite set N.

How

Try to see whether, when proving $P(n)$ for an arbitrary element $n \in N$, it may be helpful to know that $P(m)$ holds already for m smaller than n, for an appropriate notion of smaller. For instance:

- on natural numbers: less-than, successor ('sub-number')
- on positive natural numbers: divisibility
- on strings: prefix, suffix, subsequence, sub-string

When and how to apply induction?

When

Prove a property P of elements of an infinite set N.

How

Try to see whether, when proving $P(n)$ for an arbitrary element $n \in N$, it may be helpful to know that $P(m)$ holds already for m smaller than n, for an appropriate notion of smaller. For instance:

- on natural numbers: less-than, successor ('sub-number')
- on positive natural numbers: divisibility
- on strings: prefix, suffix, subsequence, sub-string
- on (finite) graphs: sub-graph

When and how to apply induction?

When

Prove a property P of elements of an infinite set N.

How

Try to see whether, when proving $P(n)$ for an arbitrary element $n \in N$, it may be helpful to know that $P(m)$ holds already for m smaller than n, for an appropriate notion of smaller. For instance:

- on natural numbers: less-than, successor ('sub-number')
- on positive natural numbers: divisibility
- on strings: prefix, suffix, subsequence, sub-string
- on (finite) graphs: sub-graph
- on (finite) sets: sub-set

When and how to apply induction?

When

Prove a property P of elements of an infinite set N.

How

Try to see whether, when proving $P(n)$ for an arbitrary element $n \in N$, it may be helpful to know that $P(m)$ holds already for m smaller than n, for an appropriate notion of smaller. For instance:

- on natural numbers: less-than, successor ('sub-number')
- on positive natural numbers: divisibility
- on strings: prefix, suffix, subsequence, sub-string
- on (finite) graphs: sub-graph
- on (finite) sets: sub-set
- on inductively defined structures: sub-structure

Experimenting to find what may be helpful

Example

Let M be the set of all palindromes over the alphabet $\{a, b\}$. To show $\forall x \in M.P(x)$ where $P(x) =$ if $\ell(x)$ even, then x has an even number of as.

Experimenting to find what may be helpful

Example

Let M be the set of all palindromes over the alphabet $\{a, b\}$. To show $\forall x \in M.P(x)$ where $P(x) =$ if $\ell(x)$ even, then x has an even number of as.

Observation

if $x \in M$ then either x empty, or $x = ax'a$ or $x = bx'b$ with $x' \in M$ again

Experimenting to find what may be helpful

Example

Let M be the set of all palindromes over the alphabet $\{a, b\}$. To show $\forall x \in M.P(x)$ where $P(x) =$ if $\ell(x)$ even, then x has an even number of as.

Observation

if $x \in M$ then either x empty, or $x = ax'a$ or $x = bx'b$ with $x' \in M$ again

Proof.

By well-founded induction, taking $R = \{(w, w') \mid \ell(w) < \ell(w')\}$; ordered by length

Experimenting to find what may be helpful

Example

Let M be the set of all palindromes over the alphabet $\{a, b\}$. To show $\forall x \in M.P(x)$ where $P(x) =$ if $\ell(x)$ even, then x has an even number of as.

Observation

if $x \in M$ then either x empty, or $x = ax'a$ or $x = bx'b$ with $x' \in M$ again

Proof.

By well-founded induction, taking $R = \{(w, w') \mid \ell(w) < \ell(w')\}$; ordered by length

- $x = \epsilon$ is a palindrome, has even length, and an even number of as.

Experimenting to find what may be helpful

Example

Let M be the set of all palindromes over the alphabet $\{a, b\}$. To show $\forall x \in M.P(x)$ where $P(x) =$ if $\ell(x)$ even, then x has an even number of as.

Observation

if $x \in M$ then either x empty, or $x = ax'a$ or $x = bx'b$ with $x' \in M$ again

Proof.

By well-founded induction, taking $R = \{(w, w') \mid \ell(w) < \ell(w')\}$; ordered by length

- $x = \epsilon$ is a palindrome, has even length, and an even number of as.
- Suppose x non-empty palindrome, and of even length. Induction hypotheses: $P(x')$ holds for palindromes x' shorter than x
  - if first letter of x is a, then $x = ax'a$ for some $x' \in M$ of even length. By the IH $P(x')$ holds, i.e. x' has an even number of as. 2 + even is even.
  - if first letter of x is b, then $x = bx'b$ for some $x' \in M$ of even length. $0+$ even is even.

Experimenting with the Ackermann function

Ackermann function

Function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$?

```
ack 0 n = n+1
ack m 0 = ack (m-1) 1
ack m n = ack (m-1) (ack m (n-1))
```

Experimenting with the Ackermann function

Ackermann function

Function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$?

```
ack 0 n = n+1
ack m 0 = ack (m-1) 1
ack m n = ack (m-1) (ack m (n-1))
```

`ack 4 4` does not produce an answer? or does it?

Experimenting with the Ackermann function

Ackermann function

Function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$?

```
ack 0 n = n+1
ack m 0 = ack (m-1) 1
ack m n = ack (m-1) (ack m (n-1))
```

`ack 4 4` does not produce an answer? or does it?

Observation

The value of `ack` for *m* and *n* depends on its value for

Experimenting with the Ackermann function

Ackermann function

Function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$?

```
ack 0 n = n+1
ack m 0 = ack (m-1) 1
ack m n = ack (m-1) (ack m (n-1))
```

`ack 4 4` does not produce an answer? or does it?

Observation

The value of `ack` for m and n depends on its value for

- $m - 1$ and 1

Experimenting with the Ackermann function

Ackermann function

Function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$?

```
ack 0 n = n+1
ack m 0 = ack (m-1) 1
ack m n = ack (m-1) (ack m (n-1))
```

`ack 4 4` does not produce an answer? or does it?

Observation

The value of `ack` for m and n depends on its value for

- $m - 1$ and 1
- m and $n - 1$

Experimenting with the Ackermann function

Ackermann function

Function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$?

```
ack 0 n = n+1
ack m 0 = ack (m-1) 1
ack m n = ack (m-1) (ack m (n-1))
```

`ack 4 4` does not produce an answer? or does it?

Observation

The value of `ack` for *m* and *n* depends on its value for

- $m - 1$ and 1
- m and $n - 1$
- $m - 1$ and the value of previous item

Experimenting with the Ackermann function

Ackermann function

Function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$?

```
ack 0 n = n+1
ack m 0 = ack (m-1) 1
ack m n = ack (m-1) (ack m (n-1))
```

`ack 4 4` does not produce an answer? or does it?

Observation

The value of `ack` for m and n depends on its value for

- $m - 1$ and 1
- m and $n - 1$
- $m - 1$ and the value of previous item

well-founded relation such that all of these are smaller?

Lexicographic product

Definition

Let $\leq_1, \leq_2$ be partial orders. Their lexicographic product is defined by

$$(x_1, x_2) \quad \leq_1 \times_{\text{lex}} \leq_2 \quad (y_1, y_2)$$

if $x_1 <_1 y_1$ or ($x_1 = y_1$ and $x_2 \leq_2 y_2$).

Lexicographic product

Definition

Let $\leq_1, \leq_2$ be partial orders. Their lexicographic product is defined by

$$(x_1, x_2) \quad \leq_1 \times_{\text{lex}} \leq_2 \quad (y_1, y_2)$$

if $x_1 <_1 y_1$ or ($x_1 = y_1$ and $x_2 \leq_2 y_2$).

Remarks

First compare the first elements; if that does not decide compare the second elements.
Case $\leq_1 = \leq_2$ corresponds to lexicographic order restricted to strings of length 2.

Lexicographic product

Definition

Let $\leq_1, \leq_2$ be partial orders. Their lexicographic product is defined by

$$(x_1, x_2) \quad \leq_1 \times_{\text{lex}} \leq_2 \quad (y_1, y_2)$$

if $x_1 <_1 y_1$ or ($x_1 = y_1$ and $x_2 \leq_2 y_2$).

Lemma

Lexicographic product preserves well-foundedness: $\leq_1 \times_{lex} \leq_2$ well-founded if $\leq_1, \leq_2$.

Lexicographic product

Definition

Let $\leq_1, \leq_2$ be partial orders. Their lexicographic product is defined by

$$(x_1, x_2) \quad \leq_1 \times_{\text{lex}} \leq_2 \quad (y_1, y_2)$$

if $x_1 <_1 y_1$ or ($x_1 = y_1$ and $x_2 \leq_2 y_2$).

Lemma

Lexicographic product preserves well-foundedness: $\leq_1 \times_{lex} \leq_2$ well-founded if $\leq_1, \leq_2$.

Example

Computing `ack m n` always yields a (unique) value, because recursive calls have arguments that are strictly smaller w.r.t. $\leq \times_{\text{lex}} \leq$. That is, we may speak of the Ackermann function.

Dags and trees motivation

Example (Dags)

- resource dependencies (build, citation)
- statement dependencies (out-of-order execution)
- sub-expression sharing (call-by-need)
- binary decision diagrams
- ...

Dags and trees motivation

Example (Dags)

- resource dependencies (build, citation)
- statement dependencies (out-of-order execution)
- sub-expression sharing (call-by-need)
- binary decision diagrams
- . . .

Example (Trees)

- data structures (searching, sorting, XML)
- parse tree (of text)/abstract syntax tree (of program)
- spanning tree (of graph)
- computation tree (of non-deterministic machines)
- . . .

Dags and trees

Definition (Cycle)

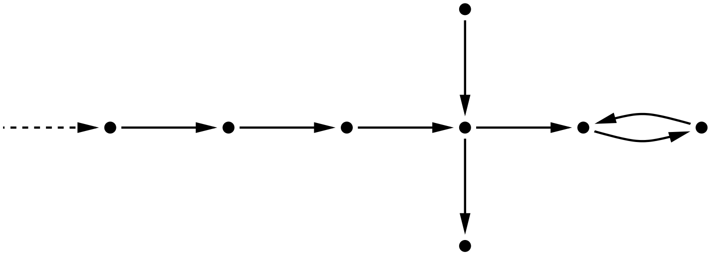
Let (V, E, src, tgt) be a directed multigraph

- a path is closed if its source is its target
- a non-empty closed path without repeated edges is a cycle
- directed multigraphs without cycles are cycle-free

Definition (Dags, forests and (rooted) trees)

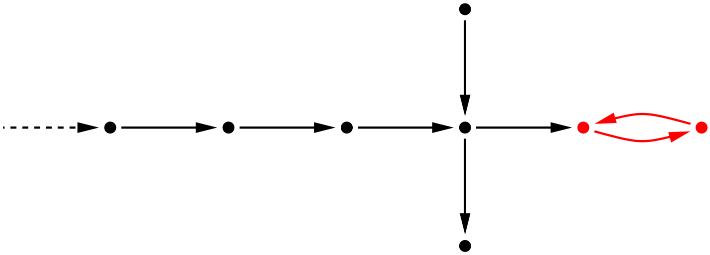
- a dag is a directed acyclic graph
- a forest is a dag with nodes of in-degree ≤ 1
- in a forest, nodes with out-degree 0 are called leaves
- a tree is a forest where all v_1, v_2 have a common ancestor v having paths to both
- a rooted tree is a tree with a node, the root, having a path to all nodes

Dags and trees example



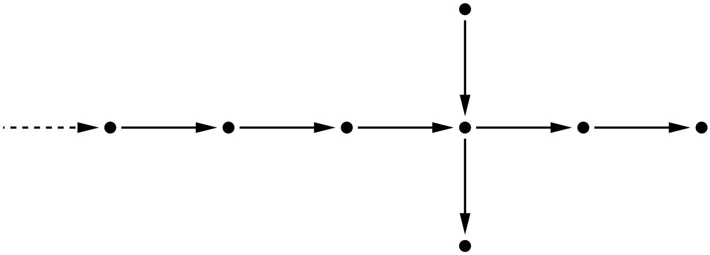
graph

Dags and trees example



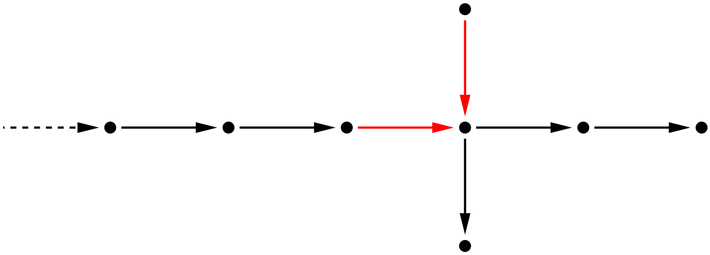
graph but not a dag (cycle)

Dags and trees example



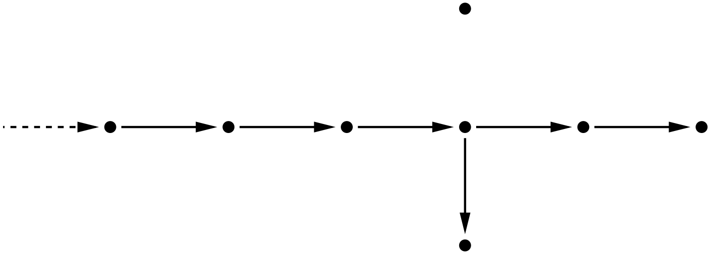
dag

Dags and trees example



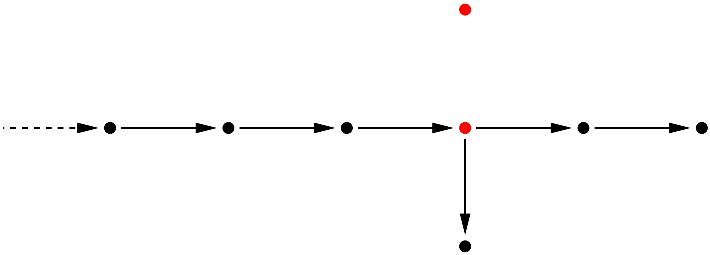
dag but not a forest (indegree 2)

Dags and trees example



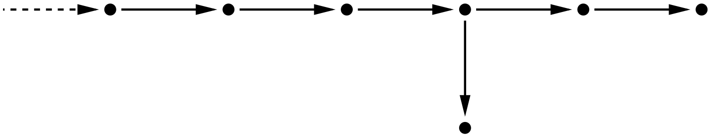
forest

Dags and trees example



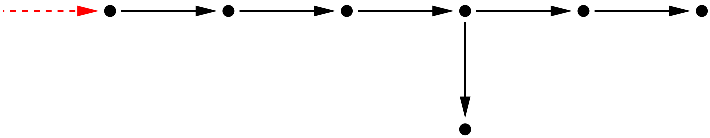
forest but not a tree (no common ancestor)

Dags and trees example



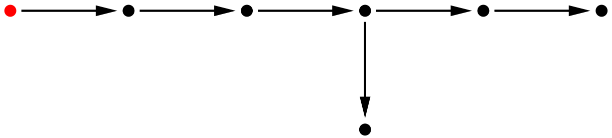
tree

Dags and trees example



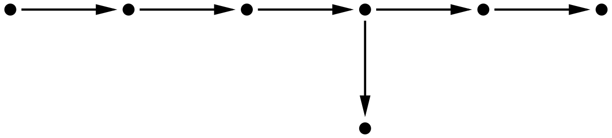
tree but not a rooted tree (no root)

Dags and trees example



rooted tree (root)

Dags and trees example



rooted tree

Simplicity in cycles

Lemma

simple paths do not have repeated edges.

Proof.

Let p be a simple path. if some edge e were to occur twice in it, the source node v of both occurrences of e would occur twice as well. ∎

Simplicity in cycles

Lemma

simple paths do not have repeated edges.

Proof.

Let p be a simple path. if some edge e were to occur twice in it, the source node v of both occurrences of e would occur twice as well. ∎

Corollary

every simple closed path in a multigraph is a cycle

Simplicity in cycles

Lemma

simple paths do not have repeated edges.

Proof.

Let p be a simple path. if some edge e were to occur twice in it, the source node v of both occurrences of e would occur twice as well. ■

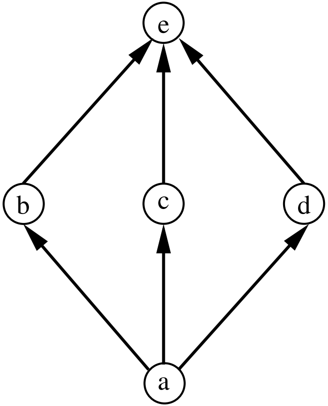
Corollary

every simple closed path in a multigraph is a cycle

Remark

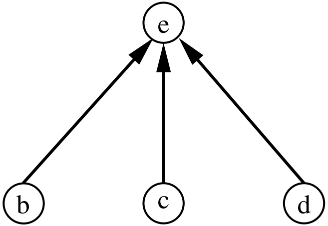
Since paths may be shortened to simple paths, cycles represent closed paths.

Topological sorting example



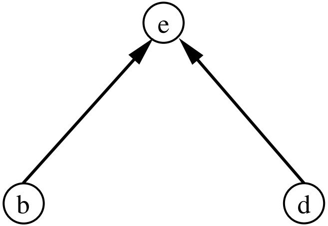
topological sorting: ()

Topological sorting example



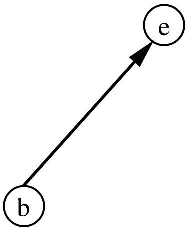
topological sorting: (a)

Topological sorting example



topological sorting: (a, c)

Topological sorting example



topological sorting: (a, c, d)

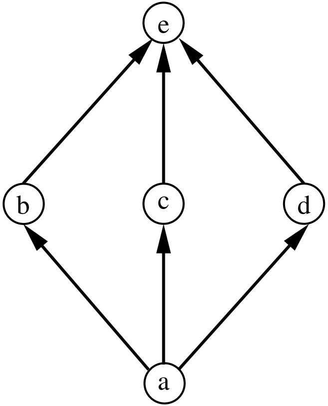
Topological sorting example

topological sorting: (a, c, d, b)

Topological sorting example

topological sorting: (a, c, d, b, e)

Topological sorting example



topological sorting: (a, c, d, b, e)

others: (a, c, b, d, e), (a, b, c, d, e), (a, b, d, c, e), (a, d, b, c, e), (a, d, c, b, e)

Topological sorting

Definition

a list $(a_0, \ldots, a_{n-1})$ is topologically $\leq$-sorted for partial order $\leq$, if $a_i < a_j$ implies $i < j$

Remark

if $\leq$ is a total order, then topologically $\leq$-sorted iff globally $\leq$-sorted

Topological sorting

Definition

a list $(a_0, \ldots, a_{n-1})$ is topologically $\leq$-sorted for partial order $\leq$, if $a_i < a_j$ implies $i < j$

Lemma

a finite set A can be topologically $\leq$-sorted by repeatedly removing minimal elements

Topological sorting

Definition

a list $(a_0, \ldots, a_{n-1})$ is topologically $\leq$-sorted for partial order $\leq$, if $a_i < a_j$ implies $i < j$

Lemma

a finite set A can be topologically $\leq$-sorted by repeatedly removing minimal elements

Proof.

if $l = (a_0, \ldots, a_{n-1})$ is topologically sorted and no element of A is smaller than any element of l, then so are $l' = (a_0, \ldots, a_{n-1}, a_n)$ and $A - \{a_n\}$ for a_n minimal in A: l' is topologically sorted since l is, and $a_n \not< a_j$ since no element of A is smaller than any element of l, and if $a_i < a_n$ then $i < n$ because $0 \leq i < n$ is an index in l. ∎

Topological sorting

Definition

a list $(a_0, \ldots, a_{n-1})$ is topologically $\leq$-sorted for partial order $\leq$, if $a_i < a_j$ implies $i < j$

Lemma

a finite set A can be topologically $\leq$-sorted by repeatedly removing minimal elements

Lemma

if G is a dag, then $\leq_G$ defined by $v \leq_G v'$ if there is a path from v to v', is a partial order.

Topological sorting

Definition

a list $(a_0, \ldots, a_{n-1})$ is topologically $\leq$-sorted for partial order $\leq$, if $a_i < a_j$ implies $i < j$

Lemma

a finite set A can be topologically $\leq$-sorted by repeatedly removing minimal elements

Lemma

if G is a dag, then $\leq_G$ defined by $v \leq_G v'$ if there is a path from v to v', is a partial order.

Proof.

reflexivity and transitivity hold by empty paths and composing paths. To see anti-symmetry consider paths from v to v' and from v' to v. Both must be empty as otherwise their composition would yield a cycle in G, hence $v = v'$. ∎

Topological sorting

Definition

a list $(a_0, \ldots, a_{n-1})$ is topologically $\leq$-sorted for partial order $\leq$, if $a_i < a_j$ implies $i < j$

Lemma

a finite set A can be topologically $\leq$-sorted by repeatedly removing minimal elements

Lemma

if G is a dag, then $\leq_G$ defined by $v \leq_G v'$ if there is a path from v to v', is a partial order.

Corollary

every finite dag G can be topologically $\leq_G$-sorted.

Shortest paths in dags

Lemma

in a finite dag, shortest and longest paths can be computed in $O(n)$

Shortest paths in dags

Lemma

in a finite dag, shortest and longest paths can be computed in $O(n)$

Proof.

shortest path adapting topological sorting: let G be weighted graph with nodes v, v'.

1. initialise v with distance 0
2. while G is non-empty
   a) set w to a minimal node having some distance (no edges from other such to w), say d
   b) if $w = v'$ return d
   c) for each edge $e : w \to_k w'$ set the distance d' of w' to $\min(d', d + k)$.
   d) remove w and all edges from it, from G
3. return ∞

∎

Facts on trees

Lemma

every finite tree is a rooted tree.

Facts on trees

Lemma

every finite tree is a rooted tree.

Proof.

let G be a finite tree having, say, n nodes $\{v_1, \ldots, v_n\}$. Setting $v'_1 = v_1$ and v'_{i+1} to be a common ancestor of v'_i and v_{i+1}, we obtain that v'_n is a common ancestor of all nodes. Therefore, v'_n is the root. ■ ■

Lemma (Characterising forests and rooted trees)

1. *in a forest there is at most one path from a node v to a node v'*

Proof.

Lemma (Characterising forests and rooted trees)

1. *in a forest there is at most one path from a node v to a node v'*
2. *a multigraph is a rooted tree iff there is a node v with a unique path to each node*

Proof.

Lemma (Characterising forests and rooted trees)

1. *in a forest there is at most one path from a node v to a node v'*
2. *a multigraph is a rooted tree iff there is a node v with a unique path to each node*

Proof.

1. For a proof by contradiction, suppose there were two paths from v to v'. If $v = v'$, then one of them would be a cycle, contradicting acyclicity. If $v \neq v'$ let $e \neq f$ be the last edges where the paths differ, starting comparing from v'. By being the last such, e and f must have the same target, contradicting in-degree ≤ 1.

Lemma (Characterising forests and rooted trees)

1. *in a forest there is at most one path from a node v to a node v'*
2. *a multigraph is a rooted tree iff there is a node v with a unique path to each node*

Proof.

1. For a proof by contradiction, suppose there were two paths from v to v'. If $v = v'$, then one of them would be a cycle, contradicting acyclicity. If $v \neq v'$ let $e \neq f$ be the last edges where the paths differ, starting comparing from v'. By being the last such, e and f must have the same target, contradicting in-degree ≤ 1.
2. (only–if) By the definition of rooted tree and the previous item.
   (if) Uniqueness of paths entails the multigraph can have neither parallel edges nor cycles, so is a dag. If there were edges $e \neq f$ with the same target v', then there would be distinct paths from v to v' via the respective sources of e and f, which cannot be, so in-degree ≤ 1 and we have a forest. Taking v as root shows the forest is a rooted tree. ∎

The number of edges and vertices in a tree

Lemma

The number of vertices in a finite tree is the number of edges $+1$

The number of edges and vertices in a tree

Lemma

The number of vertices in a finite tree is the number of edges $+1$

Proof.

Since the tree is finite, it has some root v. Consider the relation R relating every vertex v' to the last edge on a path from v to v'.

The number of edges and vertices in a tree

Lemma

The number of vertices in a finite tree is the number of edges $+1$

Proof.

Since the tree is finite, it has some root v. Consider the relation R relating every vertex v' to the last edge on a path from v to v'.

- R is a function from $V - \{v\}$ to E, since for each node $v' \neq v$ there is a unique non-empty path from the root v to v'.

The number of edges and vertices in a tree

Lemma

The number of vertices in a finite tree is the number of edges $+1$

Proof.

Since the tree is finite, it has some root v. Consider the relation R relating every vertex v' to the last edge on a path from v to v'.

- R is a function from $V - \{v\}$ to E, since for each node $v' \neq v$ there is a unique non-empty path from the root v to v'.
- R^{-1} relates edges to their targets. It is a function from E to $V - \{v\}$, since any edge, say from v' to v'' is the last edge of the unique path from v to v' to v'', and its target v'' is distinct from the root (otherwise there would be a cycle).

The number of edges and vertices in a tree

Lemma

The number of vertices in a finite tree is the number of edges $+1$

Proof.

Since the tree is finite, it has some root v. Consider the relation R relating every vertex v' to the last edge on a path from v to v'.

- R is a function from $V - \{v\}$ to E, since for each node $v' \neq v$ there is a unique non-empty path from the root v to v'.
- R^{-1} relates edges to their targets. It is a function from E to $V - \{v\}$, since any edge, say from v' to v'' is the last edge of the unique path from v to v' to v'', and its target v'' is distinct from the root (otherwise there would be a cycle).

We obtain R is bijection, hence $V - \{v\}$ and E are equinumerous. ■

Definition (undirected multigraph)

An undirected multigraph is given by

- a set of nodes or vertices V
- a set of edges E
- a map $r\colon E \to \{\{c,d\} \mid c,d \in V\}$ with $e \mapsto r(e)$, that maps every edge e to a set $r(e)$ having one or two elements, its endpoints.
- e is an edge between, joining or incident on its endpoints

Definition (undirected multigraph)

An undirected multigraph is given by

- a set of nodes or vertices V
- a set of edges E
- a map $r\colon E \to \{\{c,d\} \mid c,d \in V\}$ with $e \mapsto r(e)$, that maps every edge e to a set $r(e)$ having one or two elements, its endpoints.
- e is an edge between, joining or incident on its endpoints

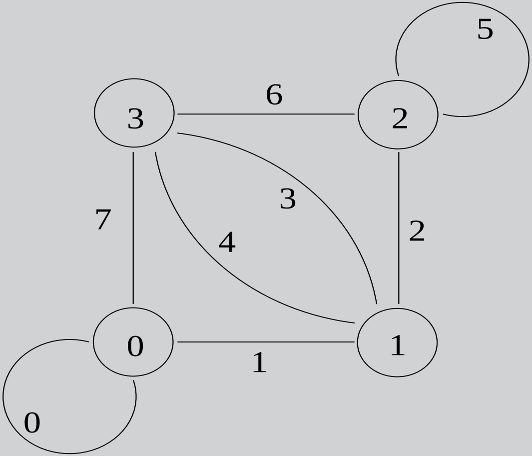
Example

Let $V = \{0,1,2,3\}$, $E = \{0,1,2,\ldots,7\}$ and the function r be defined by

e	$r(e)$
0	$\{0\}$
1	$\{0,1\}$
2	$\{1,2\}$
3	$\{1,3\}$

e	$r(e)$
4	$\{1,3\}$
5	$\{2\}$
6	$\{2,3\}$
7	$\{0,3\}$

Example (Continued)



From directed to undirected multigraphs, and back

Definition

To a directed multigraph an undirected multigraph can be associated by forgetting the directions of edges, defining the set of end-points of an edge e to comprise its source and target: $r(e) = \{src(e), tgt(e)\}$.

From directed to undirected multigraphs, and back

Definition

To a directed multigraph an undirected multigraph can be associated by forgetting the directions of edges, defining the set of end-points of an edge e to comprise its source and target: $r(e) = \{src(e), tgt(e)\}$.

Definition

To an undirected multigraph a directed multigraph can be associated by duplicating each edge e into e_l and e_r directed to the left resp. right, i.e. if $r(e) = \{c, d\}$ then e_l is from d to c, and e_r from c to d.

From directed to undirected multigraphs, and back

Definition

To a directed multigraph an undirected multigraph can be associated by forgetting the directions of edges, defining the set of end-points of an edge e to comprise its source and target: $r(e) = \{src(e), tgt(e)\}$.

Definition

To an undirected multigraph a directed multigraph can be associated by duplicating each edge e into e_l and e_r directed to the left resp. right, i.e. if $r(e) = \{c, d\}$ then e_l is from d to c, and e_r from c to d.

Remark

Not inverse to each other, but often preserve properties. For instance, there being a path between two nodes.