Inductive Definitions and Proofs

Inductive Definitions

In mathematics, it is often the case that the most natural definition of an object (like of a set, a relation, a function, etc.) is not to define it all at once but proceed in incremental steps:

- to define it at first for some smallest, elementary case
  — **basis** of the definition

- to describe a way how to define it for a bigger case assuming it was already defined for all smaller cases; this definition can refer to these smaller cases
  — **inductive step**

Inductive Definitions

Example: We would like to define a function $S : \mathbb{N} \to \mathbb{N}$ that assigns to each natural number n the sum of all natural numbers from 0 to n, i.e.,

$$S(n) = 0 + 1 + 2 + 3 + \cdots + n$$

- **Basis:** $S(0) = 0$
- **Inductive step:** For $n > 0$ it holds that

$$S(n) = S(n-1) + n$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$S(5) = S(4) + 5$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$S(5) = S(4) + 5$$
$$S(4) = S(3) + 4$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$S(5) = S(4) + 5$$
$$S(4) = S(3) + 4$$
$$S(3) = S(2) + 3$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$S(5) = S(4) + 5$$
$$S(4) = S(3) + 4$$
$$S(3) = S(2) + 3$$
$$S(2) = S(1) + 2$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$S(5) = S(4) + 5$$
$$S(4) = S(3) + 4$$
$$S(3) = S(2) + 3$$
$$S(2) = S(1) + 2$$
$$S(1) = S(0) + 1$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$
\begin{aligned}
S(5) &= S(4) + 5 \\
S(4) &= S(3) + 4 \\
S(3) &= S(2) + 3 \\
S(2) &= S(1) + 2 \\
S(1) &= S(0) + 1 \\
S(0) &= 0
\end{aligned}
$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$S(5) = S(4) + 5$$
$$S(4) = S(3) + 4$$
$$S(3) = S(2) + 3$$
$$S(2) = S(1) + 2$$
$$S(1) = S(0) + 1 = 0 + 1 = 1$$
$$S(0) = 0$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n \quad \text{for } n > 0$

$$
\begin{aligned}
S(5) &= S(4) + 5 \\
S(4) &= S(3) + 4 \\
S(3) &= S(2) + 3 \\
S(2) &= S(1) + 2 = 1 + 2 = 3 \\
S(1) &= S(0) + 1 = 0 + 1 = 1 \\
S(0) &= 0
\end{aligned}
$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$
\begin{aligned}
S(5) &= S(4) + 5 \\
S(4) &= S(3) + 4 \\
S(3) &= S(2) + 3 = 3 + 3 = 6 \\
S(2) &= S(1) + 2 = 1 + 2 = 3 \\
S(1) &= S(0) + 1 = 0 + 1 = 1 \\
S(0) &= 0
\end{aligned}
$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$
\begin{aligned}
S(5) &= S(4) + 5 \\
S(4) &= S(3) + 4 = 6 + 4 = 10 \\
S(3) &= S(2) + 3 = 3 + 3 = 6 \\
S(2) &= S(1) + 2 = 1 + 2 = 3 \\
S(1) &= S(0) + 1 = 0 + 1 = 1 \\
S(0) &= 0
\end{aligned}
$$

Inductive Definitions

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n) = S(n-1) + n$ for $n > 0$

$$S(5) = S(4) + 5 = 10 + 5 = 15$$
$$S(4) = S(3) + 4 = 6 + 4 = 10$$
$$S(3) = S(2) + 3 = 3 + 3 = 6$$
$$S(2) = S(1) + 2 = 1 + 2 = 3$$
$$S(1) = S(0) + 1 = 0 + 1 = 1$$
$$S(0) = 0$$

Inductive Definitions

Remark: For natural numbers we often use a variant where the inductive step gives the definition for $n+1$ using already defined case for n.

The above definition in this variant looks as follows:

- **Basis:** $S(0) = 0$
- **Inductive step:** $S(n{+}1) = S(n) + (n{+}1)$ for $n \geq 0$

Inductive Definitions

To ensure **correctness** of an inductive definition, the following must be satisfied:

- there must be elementary basic cases (the basis of the definition) where the definition does not refer to smaller cases.
- It must be ensured that for every element these elementary cases are reached after finite number of step, i.e., there can not exist infinite sequences, where each element is smaller than the previous one.

Examples of incorrect definitions:

- To define function $f : \mathbb{Z} \to \mathbb{Z}$ by an inductive step where $f(x)$ is defined using $f(x-1)$.
- To define function $g : \mathbb{R}_+ \to \mathbb{R}$ by an inductive step where $g(x)$ is defined using $g(x/2)$.

Proofs by Induction

n	$S(n)$
0	0
1	1
2	3
3	6
4	10
5	15
6	21
7	28
8	36
9	45
10	55
11	66
12	78
13	91

Proofs by Induction

n	$S(n)$
0	0
1	1
2	3
3	6
4	10
5	15
6	21
7	28
8	36
9	45
10	55
11	66
12	78
13	91

Hypothesis:

It seems that for every n it holds that

$$S(n) = \frac{n \cdot (n+1)}{2}$$

Proofs by Induction

n	$S(n)$
0	0
1	1
2	3
3	6
4	10
5	15
6	21
7	28
8	36
9	45
10	55
11	66
12	78
13	91

Hypothesis:

It seems that for every n it holds that

$$S(n) = \frac{n \cdot (n+1)}{2}$$

We would like to prove this hypothesis,
resp. to verify that it really holds for all $n \in \mathbb{N}$.

Proofs by Induction

Induction is used not only for defining objects but also as a tool for **proving** propositions that assert something about these inductively defined objects.

Induction is typically used for proofs of propositions that claim that something (e.g., a proposition $\varphi(n)$) holds in all cases (e.g., "for all natural numbers n").

Such **inductive proof** has the always the following form:

- **Basis**: We will prove that the proposition holds in elementary smallest cases — e.g., that it holds that $\varphi(0)$.
- **Inductive step**: We will prove that if the given proposition holds in smaller cases then it also holds in a bigger case.
  The assumption that the proposition holds in smaller cases is called the **inductive hypothesis**.

Proofs by Induction

In the case of natural numbers, an inductive proof of a proposition of the form

Proved proposition

for each $n \in \mathbb{N}$ it holds that $\varphi(n)$

is often of the following form:

- **Basis:** A proof that $\varphi(0)$ holds.
- **Inductive step:** A proof that the following implication holds for all $n \in \mathbb{N}$:

  "if $\varphi(n)$ then $\varphi(n+1)$"

Proofs by Induction

In the case of natural numbers, an inductive proof of a proposition of the form

Proved proposition

for each $n \in \mathbb{N}$ it holds that $\varphi(n)$

is often of the following form:

- **Basis:** A proof that $\varphi(0)$ holds.
- **Inductive step:** A proof that the following implication holds for all $n \in \mathbb{N}$:

  "if $\varphi(n)$ then $\varphi(n+1)$"

Remark: It is not necessary that the basis considers only the case $n = 0$. Sometimes it can be more convenient to include also other cases into the basis, e.g., all values of n where $n \leq 2$.

Proofs by Induction

Example: We want to prove the proposition

For each $n \in \mathbb{N}$ it holds that $S(n) = \frac{n \cdot (n+1)}{2}$.

- **Basis** $(n = 0)$: $S(0) = \frac{0 \cdot (0+1)}{2}$
- **Inductive step**: For each $n \in \mathbb{N}$ it holds that

$$\text{if } S(n) = \frac{n \cdot (n+1)}{2} \text{ then } S(n+1) = \frac{(n+1) \cdot ((n+1)+1)}{2}$$

Remark: The proposition $\varphi(n)$ in this example is

$$S(n) = \frac{n \cdot (n+1)}{2}$$

Proofs by Induction

Note that when we would have proofs that:

- $\varphi(0)$ holds
- for each n it holds that if $\varphi(n)$ then $\varphi(n+1)$ (i.e., $\varphi(n) \Rightarrow \varphi(n+1)$)

then for arbitrary concrete values of $n \in \mathbb{N}$ we can easily construct a proof that $\varphi(n)$ holds:

- $n = 0$: $\varphi(0)$ is proved directly (basis)
- $n = 1$: $\varphi(1)$ follows from $\varphi(0)$ and $\varphi(0) \Rightarrow \varphi(1)$
- $n = 2$: $\varphi(2)$ follows from $\varphi(1)$ and $\varphi(1) \Rightarrow \varphi(2)$
- $n = 3$: $\varphi(3)$ follows from $\varphi(2)$ and $\varphi(2) \Rightarrow \varphi(3)$
- $n = 4$: $\varphi(4)$ follows from $\varphi(3)$ and $\varphi(3) \Rightarrow \varphi(4)$
- $n = 5$: $\varphi(5)$ follows from $\varphi(4)$ and $\varphi(4) \Rightarrow \varphi(5)$

$\vdots$

Proofs by Induction

Finishing the previous proof:

- **Basis:** We must prove that $S(0) = \frac{0 \cdot (0+1)}{2}$

  $S(0) = 0$ — from the definition of function S

  $\frac{0 \cdot (0+1)}{2} = 0$ — simple rewriting of arithmetic expressions

Proofs by Induction

- **Inductive step:** We must show that for each n it holds that

$$\text{if } S(n) = \frac{n \cdot (n+1)}{2} \text{ then } S(n+1) = \frac{(n+1) \cdot ((n+1)+1)}{2}$$

Proofs by Induction

- **Inductive step:** We must show that for each n it holds that

$$\text{if } S(n) = \frac{n \cdot (n+1)}{2} \text{ then } S(n+1) = \frac{(n+1) \cdot ((n+1)+1)}{2}$$

Let us assume an arbitrary $n \in \mathbb{N}$.

Let us assume for this n that $S(n) = \frac{n \cdot (n+1)}{2}$ — this is the **inductive hypothesis**.

We must show that $S(n+1) = \frac{(n+1) \cdot ((n+1)+1)}{2}$.

Proofs by Induction

- **Inductive step:** We must show that for each n it holds that

$$\text{if } S(n) = \frac{n \cdot (n+1)}{2} \text{ then } S(n+1) = \frac{(n+1) \cdot ((n+1)+1)}{2}$$

Let us assume an arbitrary $n \in \mathbb{N}$.

Let us assume for this n that $S(n) = \frac{n \cdot (n+1)}{2}$ — this is the **inductive hypothesis**.

We must show that $S(n+1) = \frac{(n+1) \cdot ((n+1)+1)}{2}$.

According to the definition of function S we have $S(n+1) = S(n) + (n+1)$.

So it holds that:

$$S(n+1) = S(n) + (n+1) = \frac{n \cdot (n+1)}{2} + (n+1) = \frac{n \cdot (n+1) + 2 \cdot (n+1)}{2} =$$

$$= \frac{(n+1) \cdot (n+2)}{2} = \frac{(n+1) \cdot ((n+1)+1)}{2}$$

Proofs by Induction

Sometimes, it may be more convenient to use, instead of a proof of the form

- **Basis:** A proof that $\varphi(0)$ holds.
- **Inductive step:** A proof that for each $n \in \mathbb{N}$ the following implication holds

  "if $\varphi(n)$ then $\varphi(n+1)$"

an inductive proof of the form

- **Inductive step:** A proof that for each $n \in \mathbb{N}$ the following implication holds

  "if $\varphi(i)$ holds for each $i \in \mathbb{N}$ such that $i < n$, then $\varphi(n)$ also holds"

Proofs by Induction

Example:

Theorem

Every natural number n, such that $n > 1$, can be expressed as a product of some finite number of primes, i.e., for each such n there exists a sequence of primes $p_1, p_2, \ldots, p_k$ such that

$$n = p_1 \cdot p_2 \cdot \cdots \cdot p_k$$

Remark: A natural number n is a **prime** if it is greater than 1 and is divisible only by numbers 1 and n.

Few of the first primes: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, $\ldots$

Proofs by Induction

Proof: Let us assume an arbitrary n where $n > 1$.

By induction hypothesis, for every $i < n$ it holds that if $i > 1$ then i can be expressed as a product of a finite number of primes.

There are two possibilities:

- n is a prime: the given sequence consists of the number n itself
- n is not a prime: so there is a number $i \in \mathbb{N}$ such that $1 < i < n$ and i is a divisor of number n, which means that there is some number $j \in \mathbb{N}$ such that $i \cdot j = n$. It is obvious that $1 < j < n$.

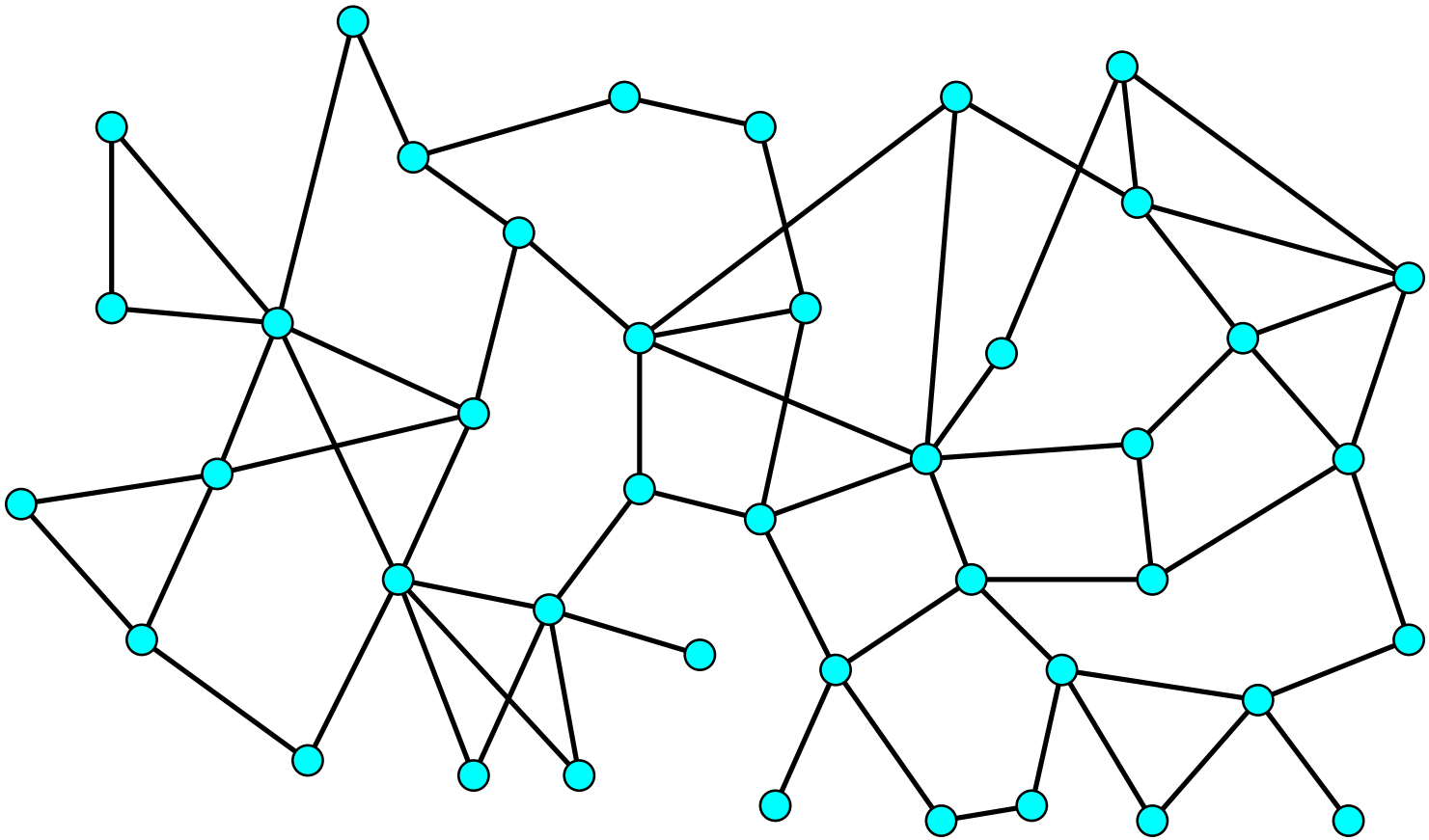
  So we can use the inductive hypothesis for both i and j.

  $$i = p_1 \cdot p_2 \cdot \cdots \cdot p_k \qquad\qquad j = q_1 \cdot q_2 \cdot \cdots \cdot q_\ell$$

  And so $n = p_1 \cdot p_2 \cdot \cdots \cdot p_k \cdot q_1 \cdot q_2 \cdot \cdots \cdot q_\ell$.

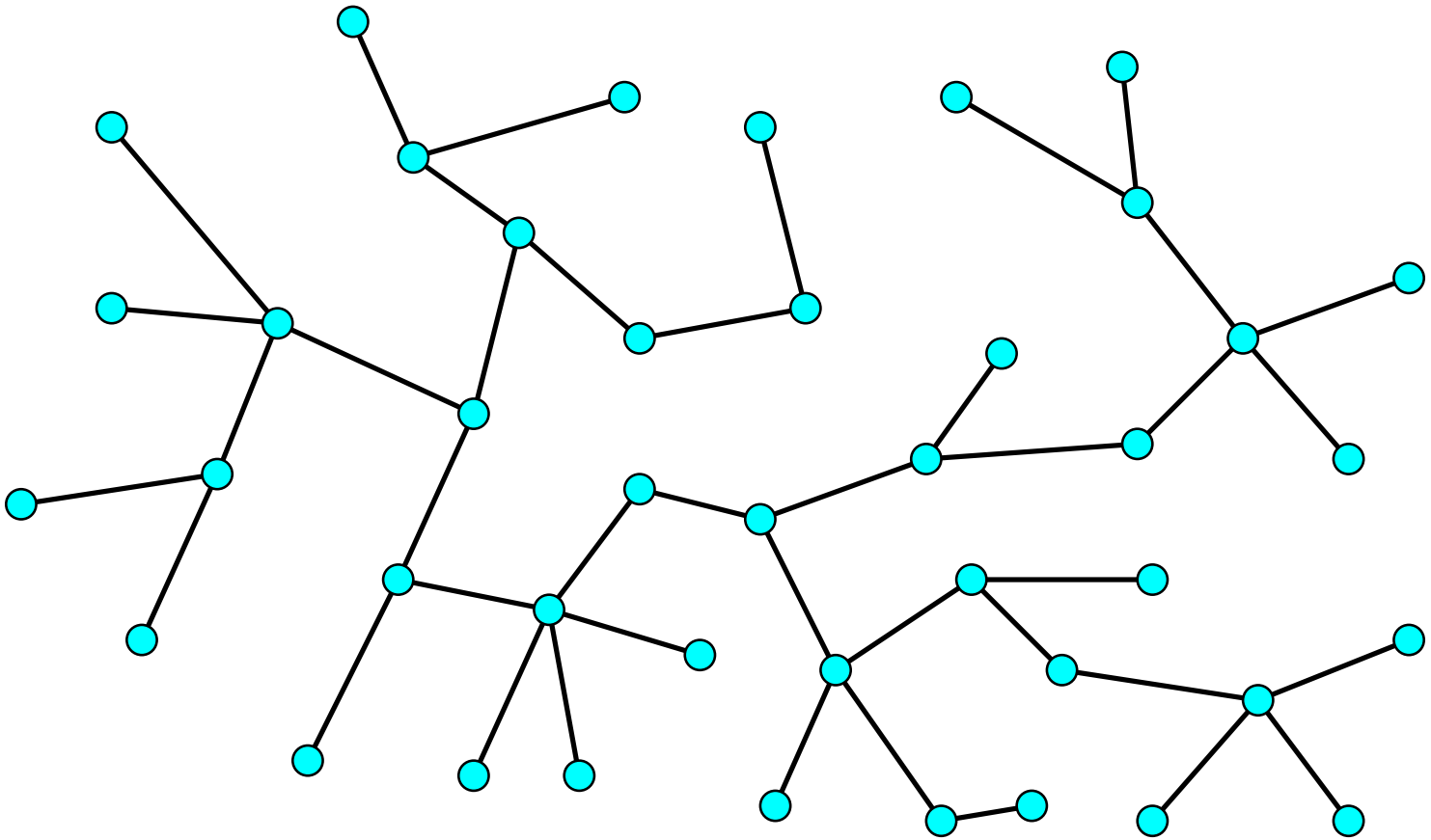
Graphs

A **graph** consists of **vertices** (or **nodes**) and **edges**.



Trees

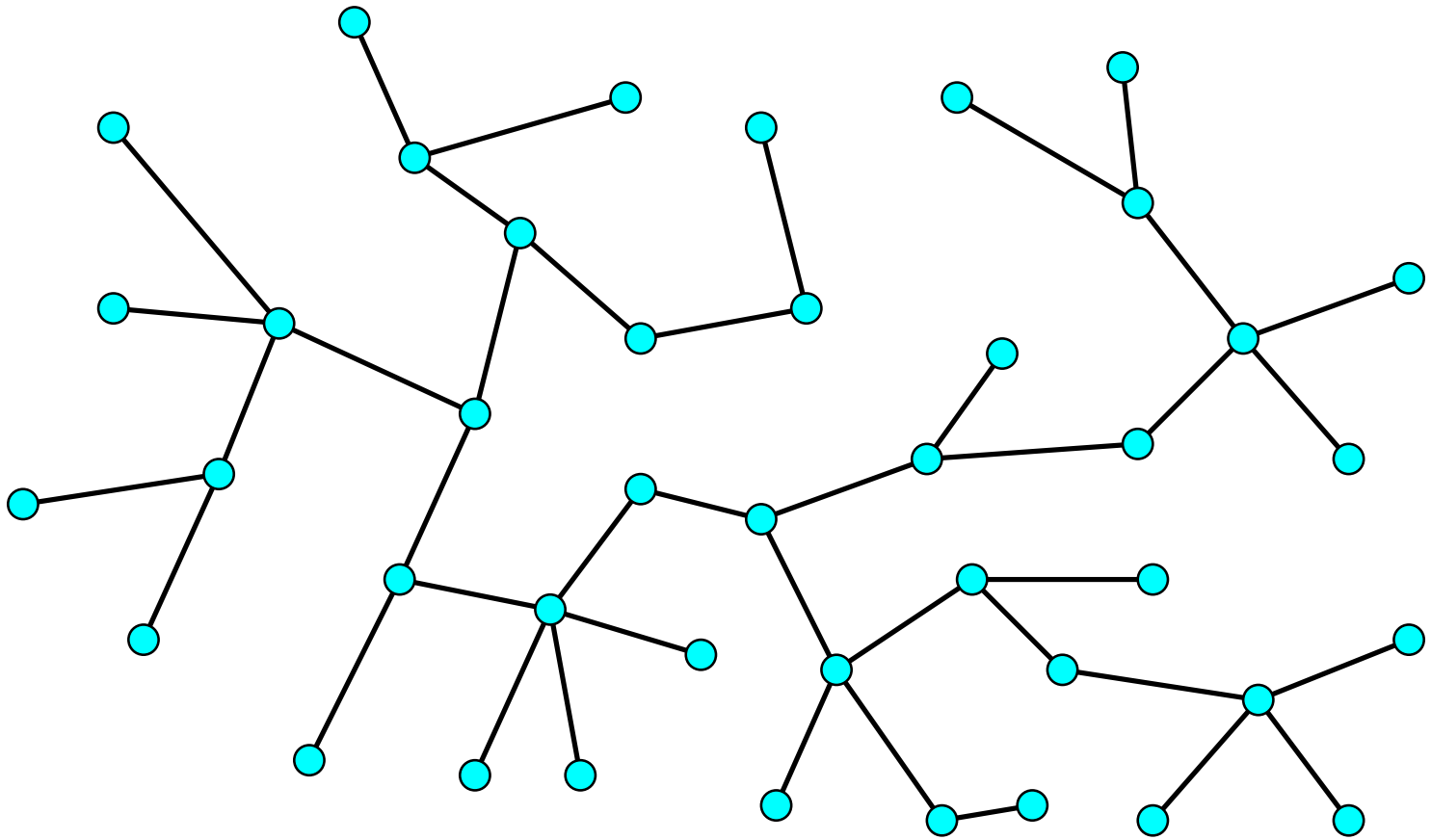
A **tree** is a special case of a graph where every node is reachable from each other and that contains no cycles.



Trees

Proposition

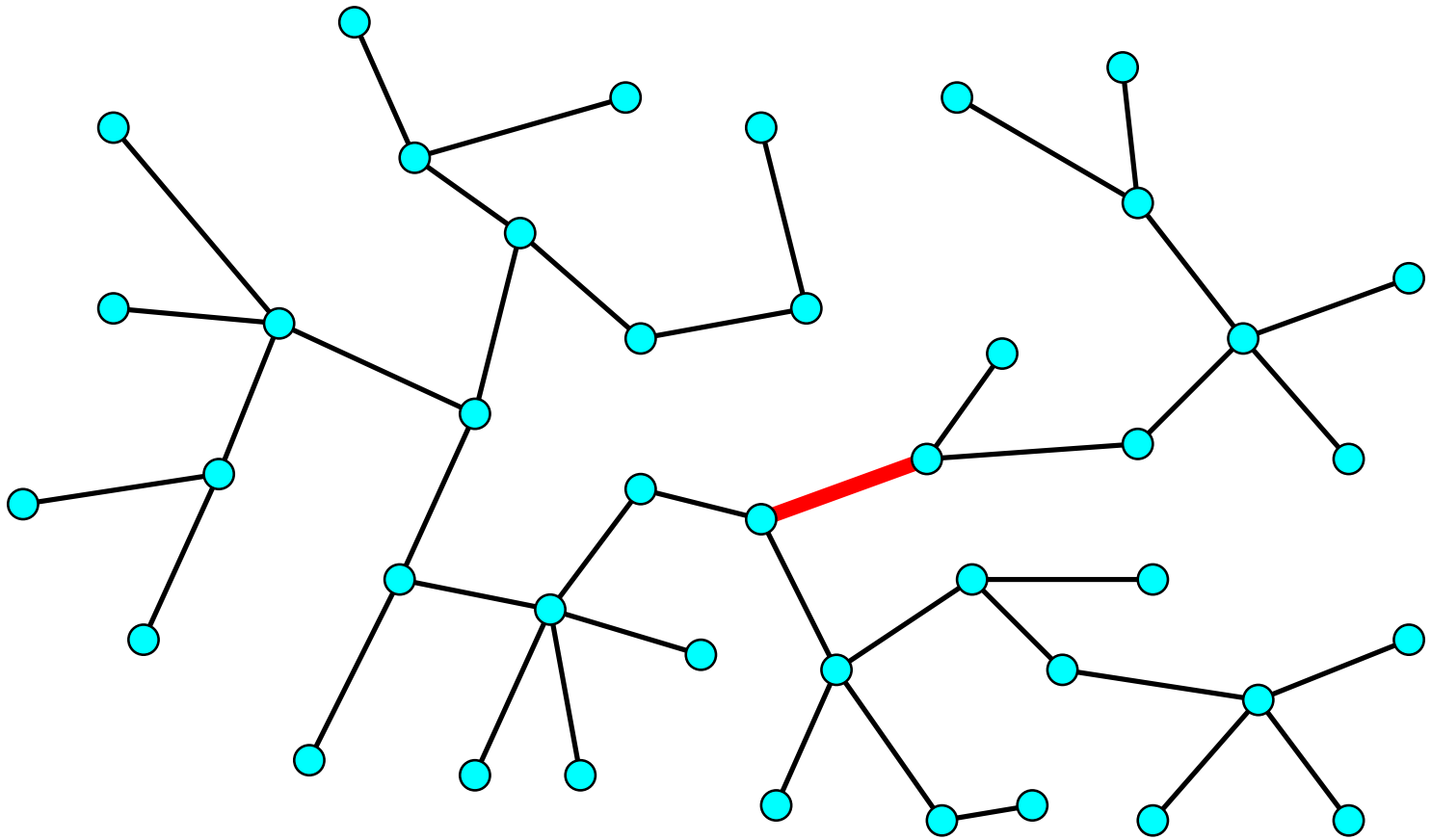
A tree with n vertices has $n - 1$ edges.



Trees

Proposition

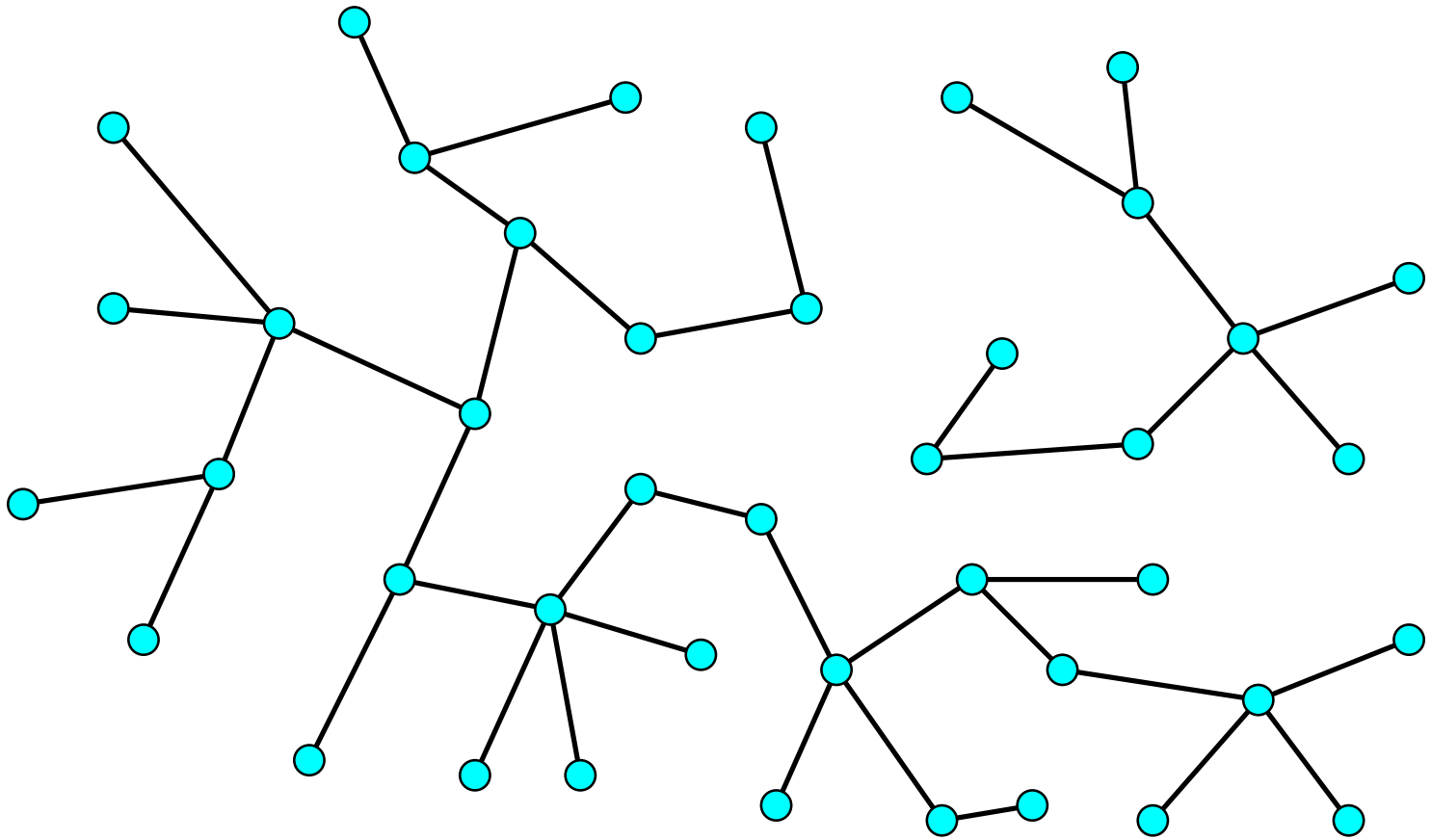
A tree with n vertices has $n - 1$ edges.



Trees

Proposition

A tree with n vertices has $n - 1$ edges.



Trees

When we remove one edge from a tree on n vertices ($n > 1$), the tree will be decomposed into two trees:

Tree 1:

- Vertices: k
- Edges: $k - 1$

Tree 2:

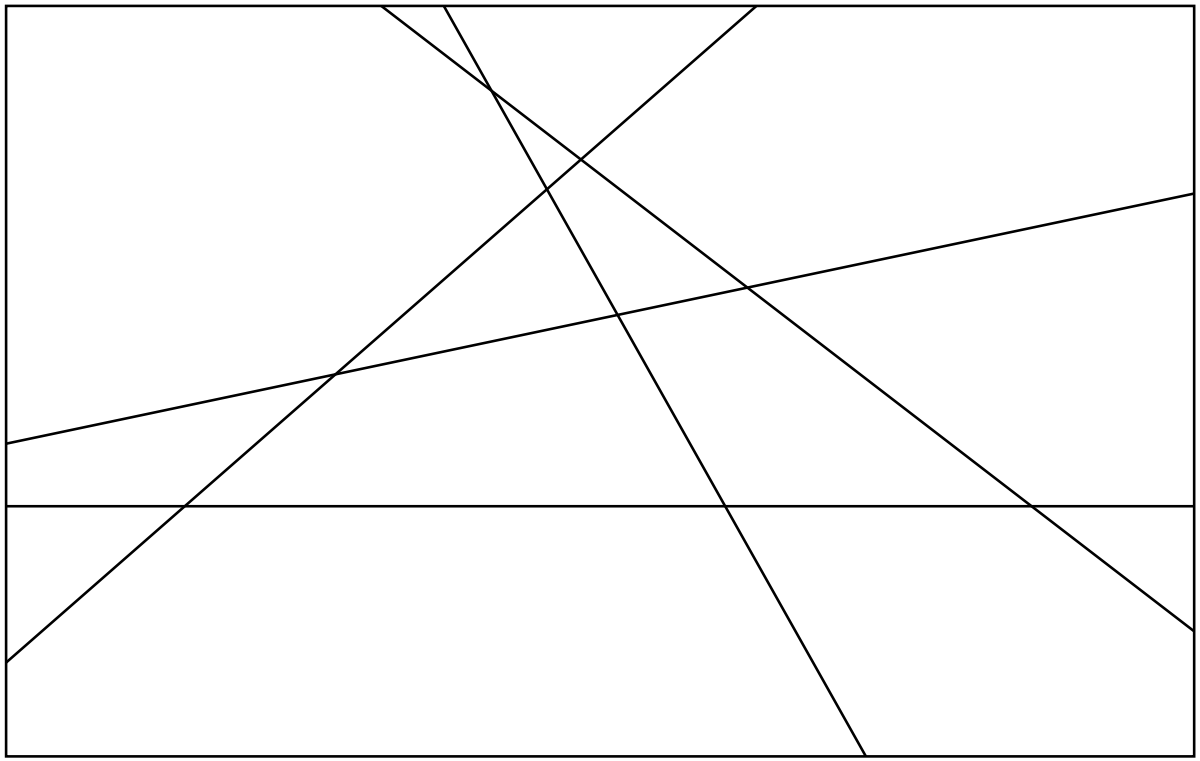
- Vertices: $n - k$
- Edges: $n - k - 1$

The number of edges of the original graph:

$$(k - 1) + (n - k - 1) + 1 = n - 1$$

Colouring of Areas Delimited by Lines

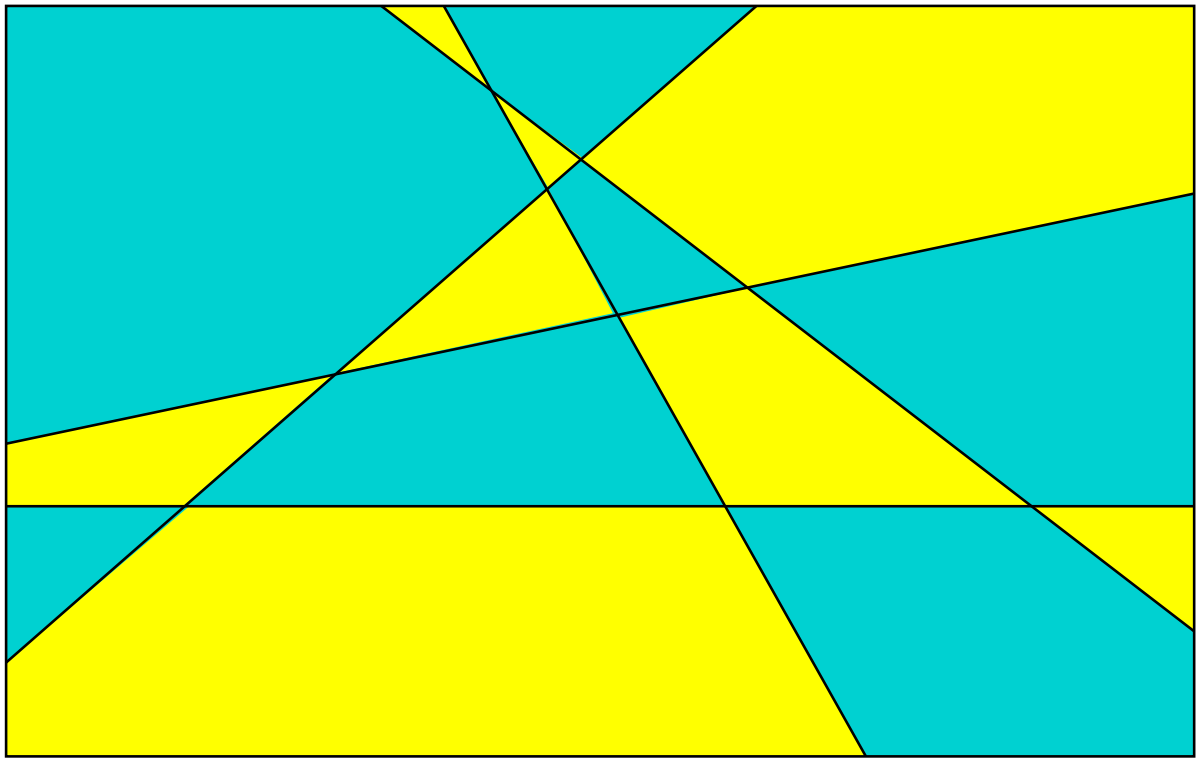
Consider an arbitrary rectangle divided to subareas by several lines.
(We assume a finite number of lines.)



Colouring of Areas Delimited by Lines

Consider an arbitrary rectangle divided to subareas by several lines. (We assume a finite number of lines.)

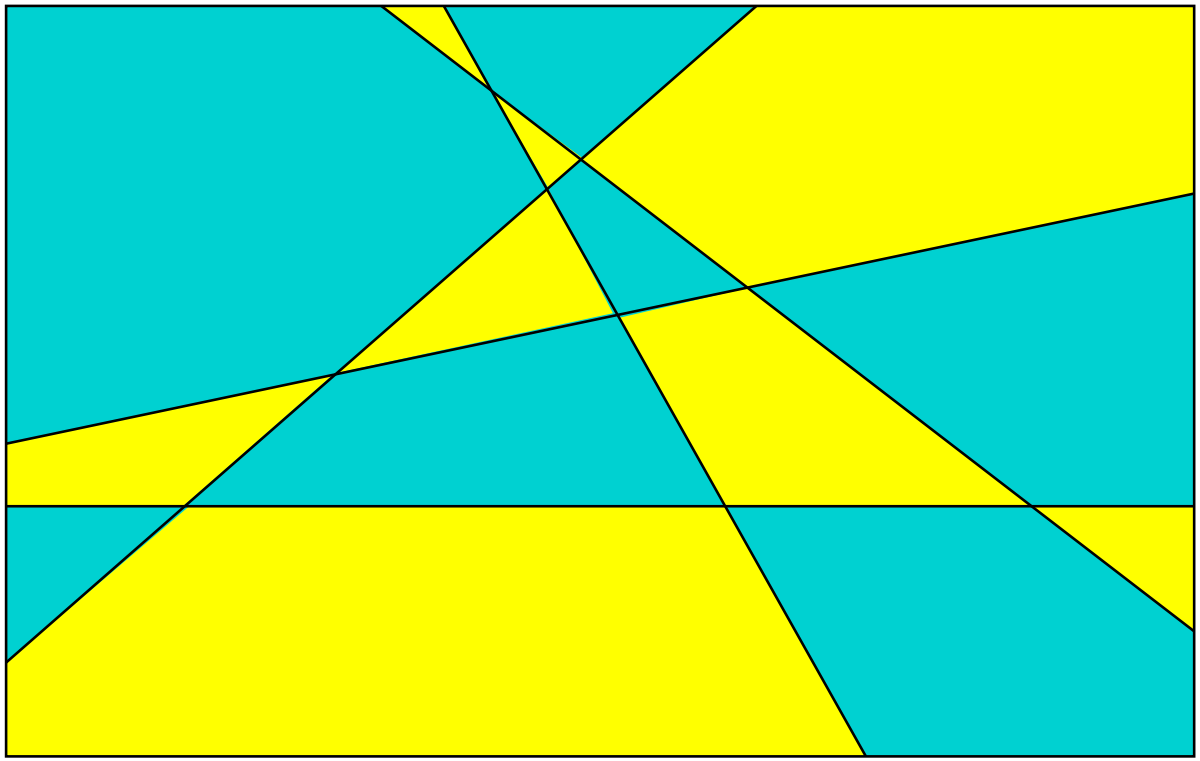
We would like to colour these subareas in such a way that every pair of subareas that have a common borderline will have different colours.



Colouring of Areas Delimited by Lines

Proposition

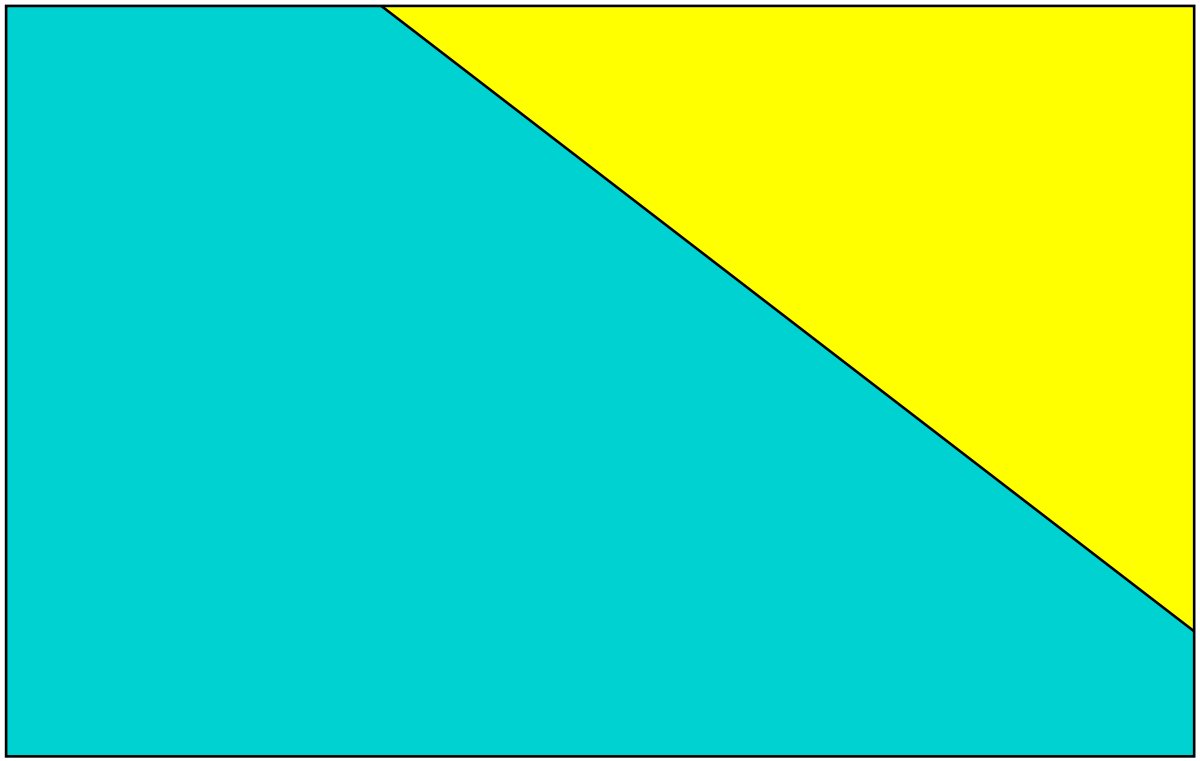
Every rectangle divided into subareas by a finite number of lines can be coloured by two colours.



Colouring of Areas Delimited by Lines

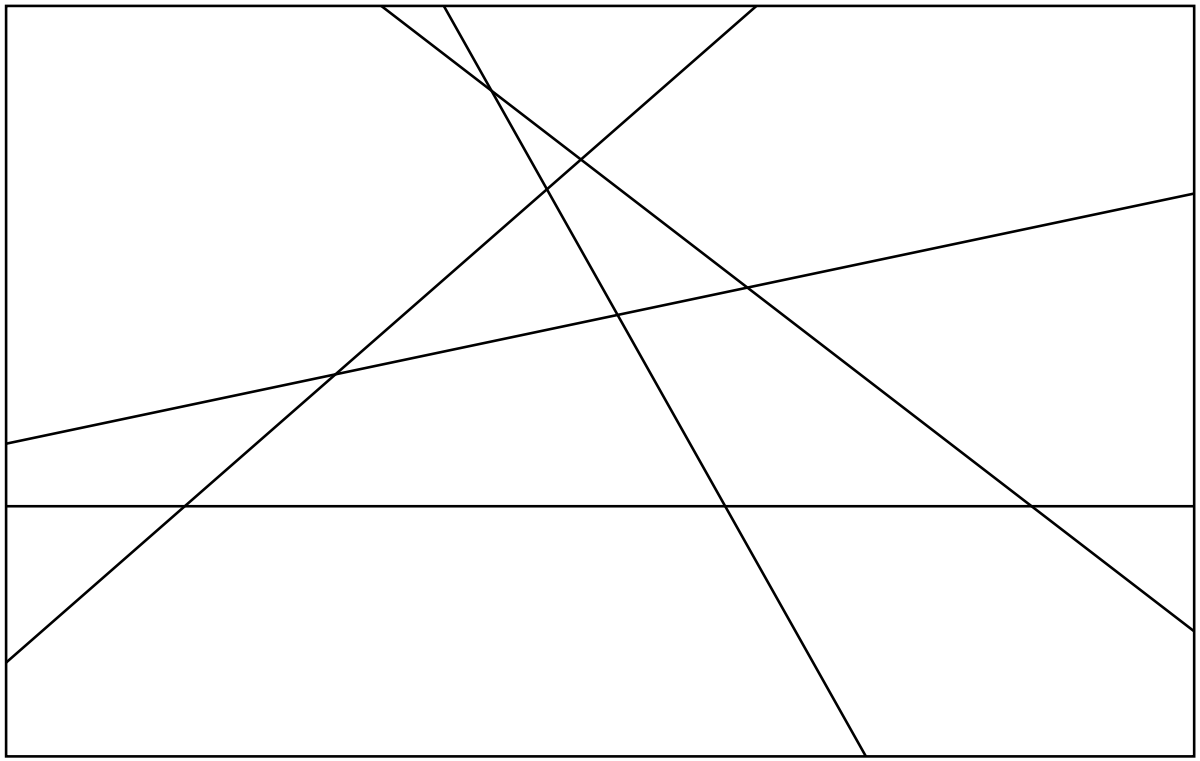
Proof: By induction on the number of lines n.

- **Basis** ($n \leq 1$): For $n = 0$ or $n = 1$, the proposition is obvious.



Colouring of Areas Delimited by Lines

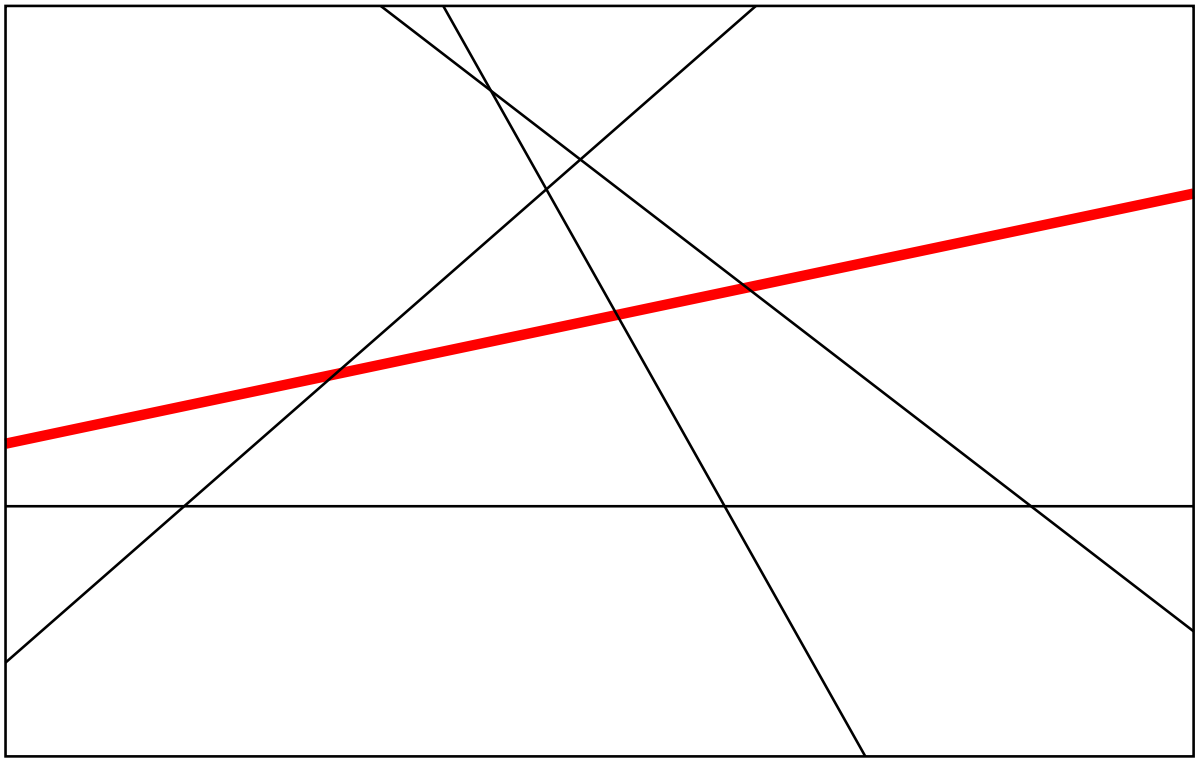
- **Inductive step** ($n > 1$):



Colouring of Areas Delimited by Lines

- **Inductive step** ($n > 1$):

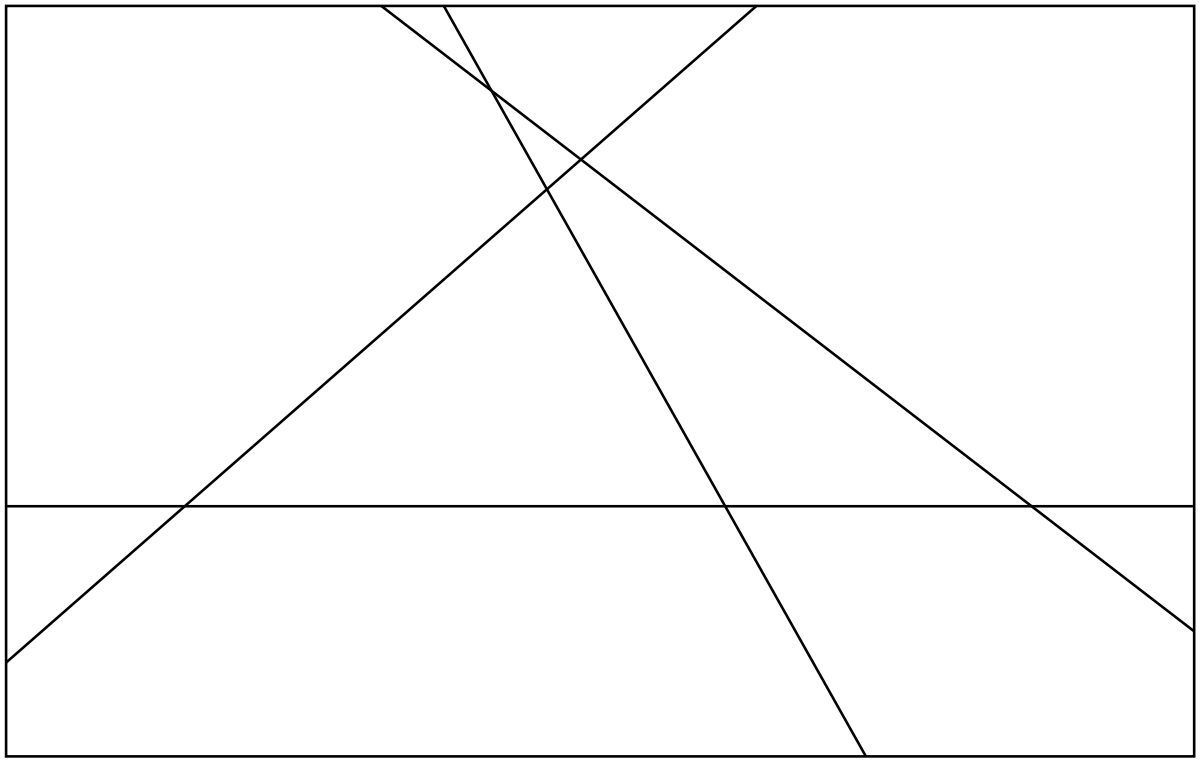
  Consider one of the lines.



Colouring of Areas Delimited by Lines

- **Inductive step** ($n > 1$):

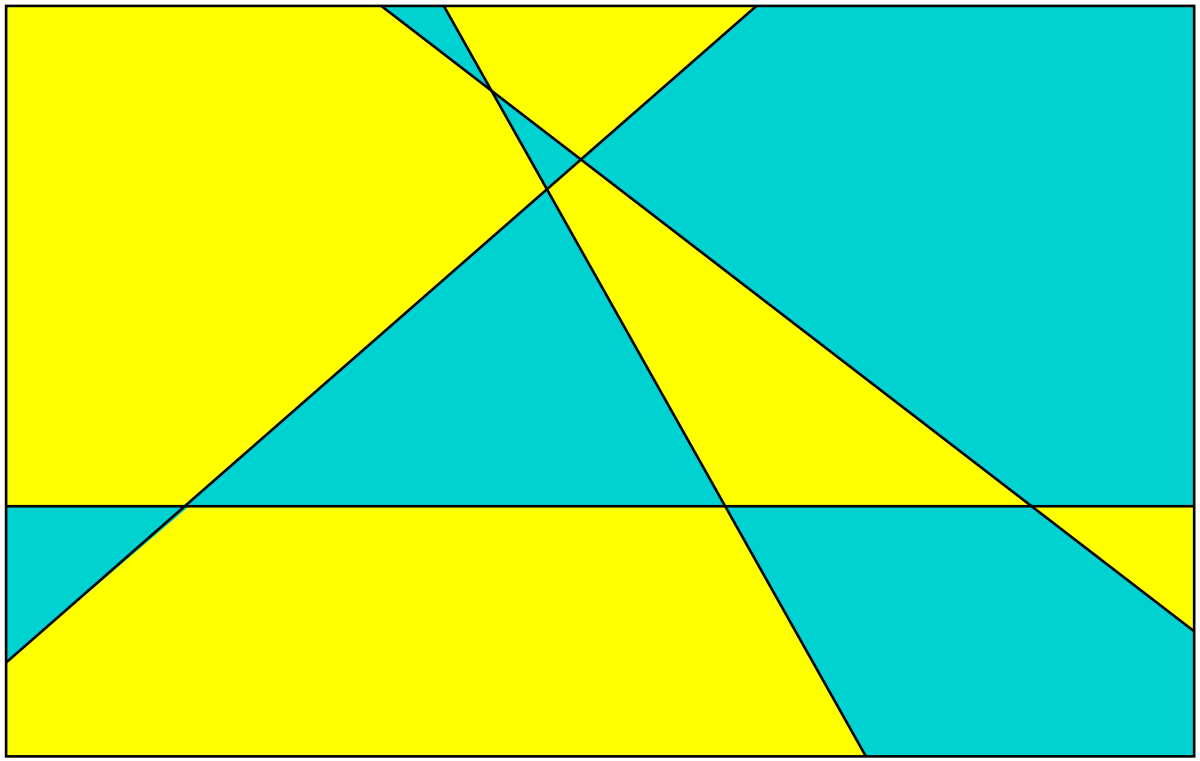
  Remove the selected line.



Colouring of Areas Delimited by Lines

- **Inductive step** ($n > 1$):

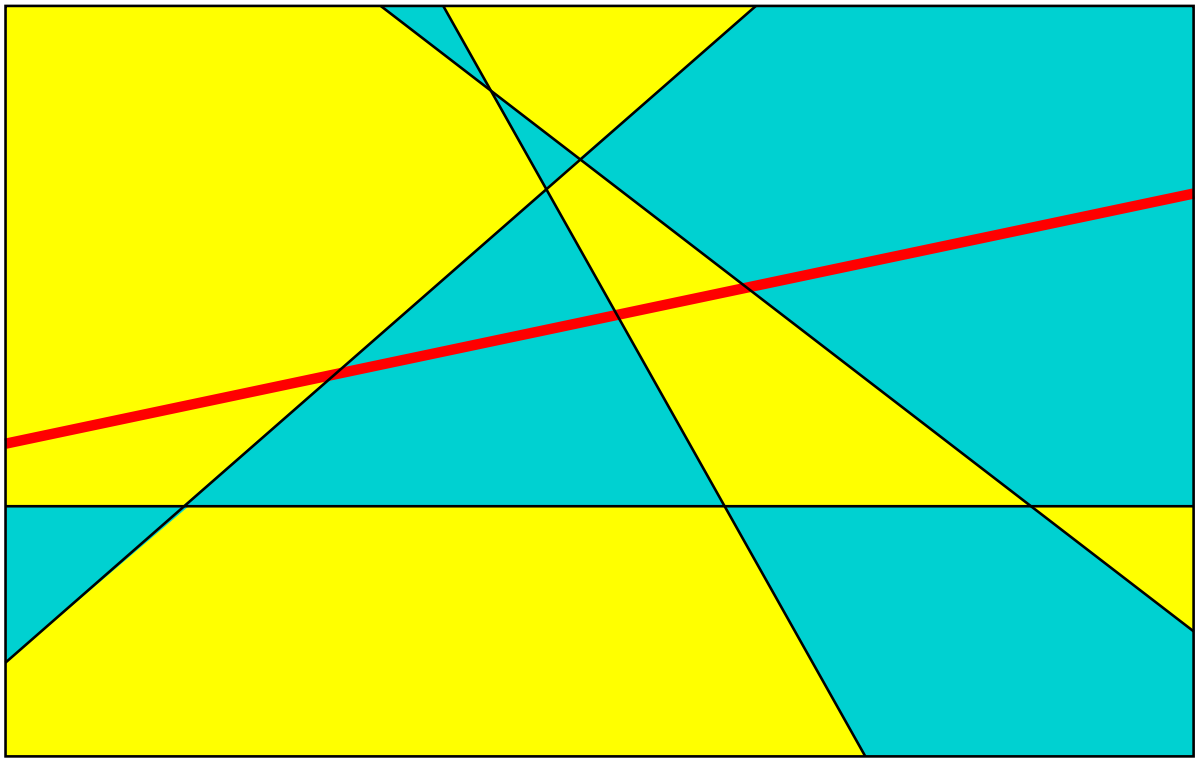
  By inductive hypothesis, the areas delimited by $n - 1$ lines can be coloured by two colours.



Colouring of Areas Delimited by Lines

- **Inductive step** ($n > 1$):

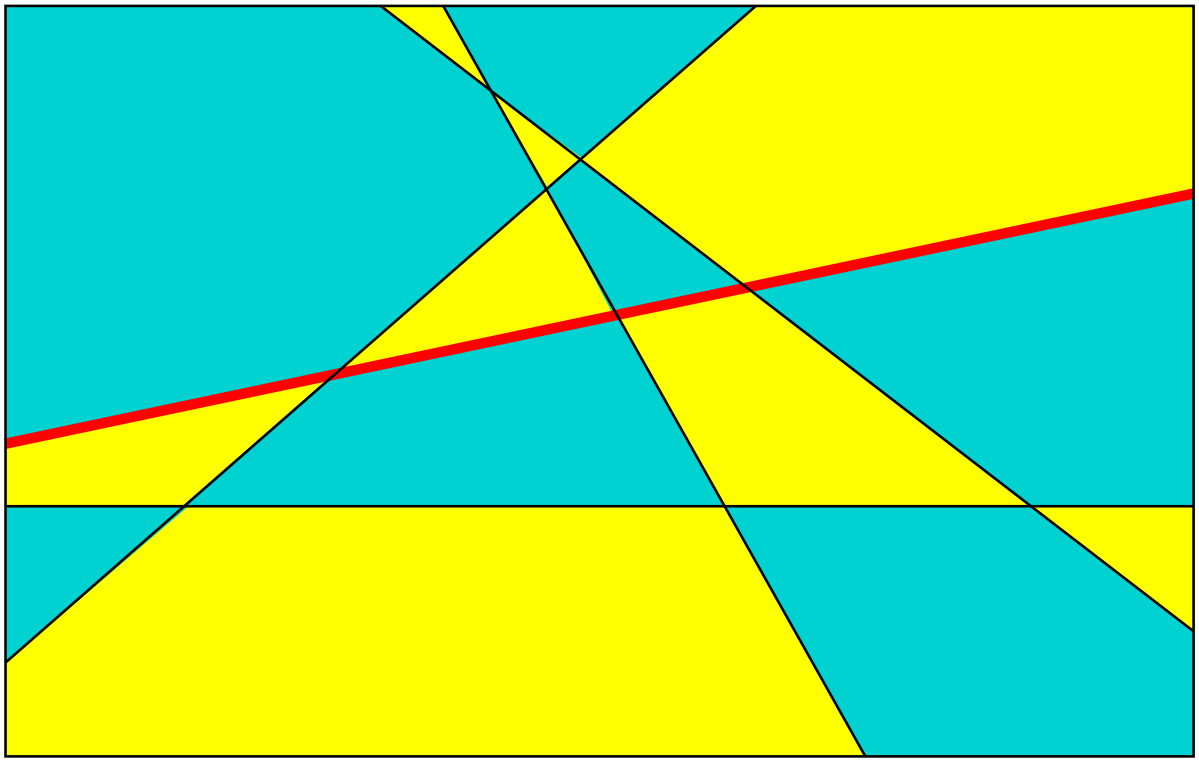
  Return the removed line back.



Colouring of Areas Delimited by Lines

- **Inductive step** ($n > 1$):

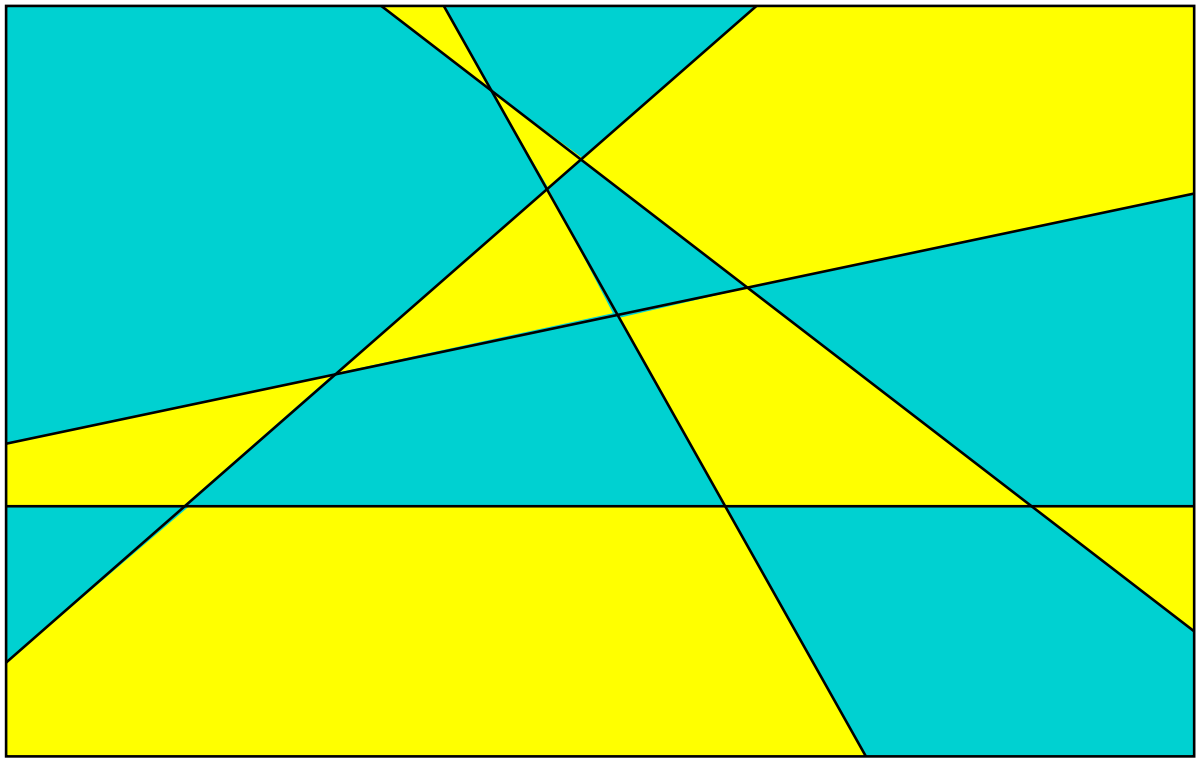
  Switch coulours in areas on one of sides of this line.



Colouring of Areas Delimited by Lines

- **Inductive step** ($n > 1$):

  We have obtained a correct colouring for n lines.



Structural Induction

Inductive definitions are often used for specifying a subset C of some given set A in the following way:

- **Basis:** There is specified some particular subset of elements of A that must belong to the set C.
- **Inductive step:** There is specified some (finite) set of rules that prescribe that whenever some elements belong to C then also some other elelements, that can be computed from them, must also belong to C.
- It is specified that C is the smallest set (with respect to inclusion) that satisfies the conditions given above.

Structural Induction

Example: An inductive definition of a subset A of the set $\mathbb{Z}$:

- **Basis:** $4 \in C$
- **Inductive step:**
  - If $x \in C$ then $x - 12 \in C$.
  - If $x \in C$ then $x^2 \in C$.

The indend is to define the set C in such a way that it consists of exactly those elements $x \in \mathbb{Z}$, for which it is possible to justify in a finite number of steps, using the rules given above, that they must belong to C.

Structural Induction

We can imagine that we successively build a sequence of **approximations** of the set C:

- $C_0 = \{4\}$
- $C_1 = \{4, -8, 16\}$
- $C_2 = \{4, -8, 16, -20, 64, 256\}$
- $C_3 = \{4, -8, 16, -20, 64, 256, -32, 400, 52, 4096, 244, 65536\}$

$\vdots$

Structural Induction

We can imagine that we successively build a sequence of **approximations** of the set C:

- $C_0 = \{4\}$
- $C_1 = \{4, -8, 16\}$
- $C_2 = \{4, -8, 16, -20, 64, 256\}$
- $C_3 = \{4, -8, 16, -20, 64, 256, -32, 400, 52, 4096, 244, 65536\}$

$\vdots$

The set C is then defined as the union of all these sets C_i:

$$C = C_0 \cup C_1 \cup C_2 \cup \cdots$$

Structural Induction

We can imagine that we successively build a sequence of **approximations** of the set C:

- $C_0 = \{4\}$
- $C_1 = \{4, -8, 16\}$
- $C_2 = \{4, -8, 16, -20, 64, 256\}$
- $C_3 = \{4, -8, 16, -20, 64, 256, -32, 400, 52, 4096, 244, 65536\}$

$\vdots$

The set C is then defined as the union of all these sets C_i:

$$C = C_0 \cup C_1 \cup C_2 \cup \cdots$$

which can be written in the following way:

$$C = \bigcup_{i \in \mathbb{N}} C_i$$

Structural Induction

This can be understood as follows:

- **Basis** of the definition specifies the approximation C_0 (where $C_0 \subseteq A$).
- **Inductive step** specifies a function $\mathcal{F} : \mathcal{P}(A) \to \mathcal{P}(A)$ that adds to an arbitrary given set $X \subseteq A$ some other elements depending on the elements that already belong to X.

  To ensure that the definition is correct, it is necessary to check that the function $\mathcal{F}$ is **monotonic**, i.e., that

  $$\text{if} \quad X \subseteq X' \qquad \text{then} \qquad \mathcal{F}(X) \subseteq \mathcal{F}(X')$$

We can then define for each $i > 0$ the set C_i (i.e., the i-th approximation) inductively as follows

$$C_i = \mathcal{F}(C_{i-1})$$

Structural Induction

Let us consider only the simpler case when the rules in the inductive step are such that whether an element $x \in A$ will be added to $\mathcal{F}(X)$ depends always only on some finite set of elements from X.

It can be shown that in this case the set $C = \bigcup_{i \in \mathbb{N}} C_i$ has the following properties:

- For each C_i it holds that $C_i \subseteq C$.
- $\mathcal{F}(C) = C$
- For each set C' such that $C_0 \subseteq C'$ and $\mathcal{F}(C') = C'$, it holds that

$$C \subseteq C'$$

Structural Induction

When a set C is defined by an inductive definition as described above, we can prove propositions of the form

for each $x \in C$ it holds that $\varphi(x)$

using a proof by **structural induction**:

- **Basis:** To prove that $\varphi(x)$ holds for all elements $x \in C_0$.
- **Inductive step:** To prove that for each set $X \subseteq A$ that if $\varphi(x)$ is true for each $x \in X$ then for each $y \in \mathcal{F}(X)$ it holds that $\varphi(y)$.

Structural Induction

Let us assume that we have constructed a proof by structural induction of the above mentioned kind.

Let us define the set

$$C' = \{x \in A \mid \varphi(x)\}$$

It obviously holds that:

- for each $x \in C'$ it holds that $\varphi(x)$
- $C_0 \subseteq C'$
- $\mathcal{F}(C') = C'$

It also follows from that previous discussion that $C \subseteq C'$, and so for each $x \in C$ we have $\varphi(x)$.

Structural Induction

Example:

Proposition

All elements of the set C, which is inductively defined below, are multiples of 4.

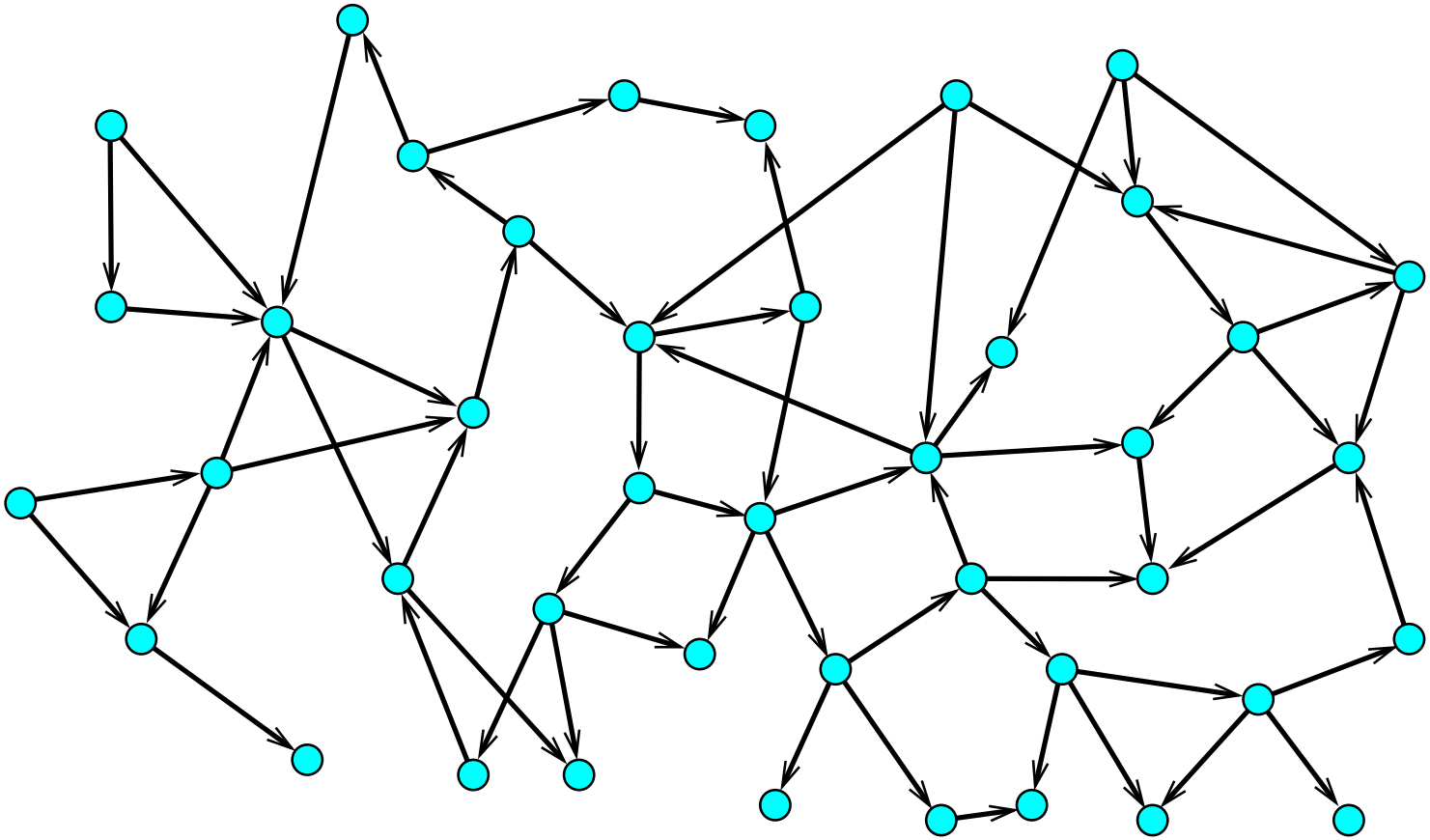
- **Basis:** $4 \in C$
- **Inductive step:**
  - If $x \in C$ then $x - 12 \in C$.
  - If $x \in C$ then $x^2 \in C$.

Proof:

- **Basis:** Number 4 is a multiple of 4.
- **Inductive step:**
  - If x is a multiple of 4 then also $x - 12$ is a multiple of 4.
  - If x is a multiple of 4 then also x^2 is a multiple of 4.

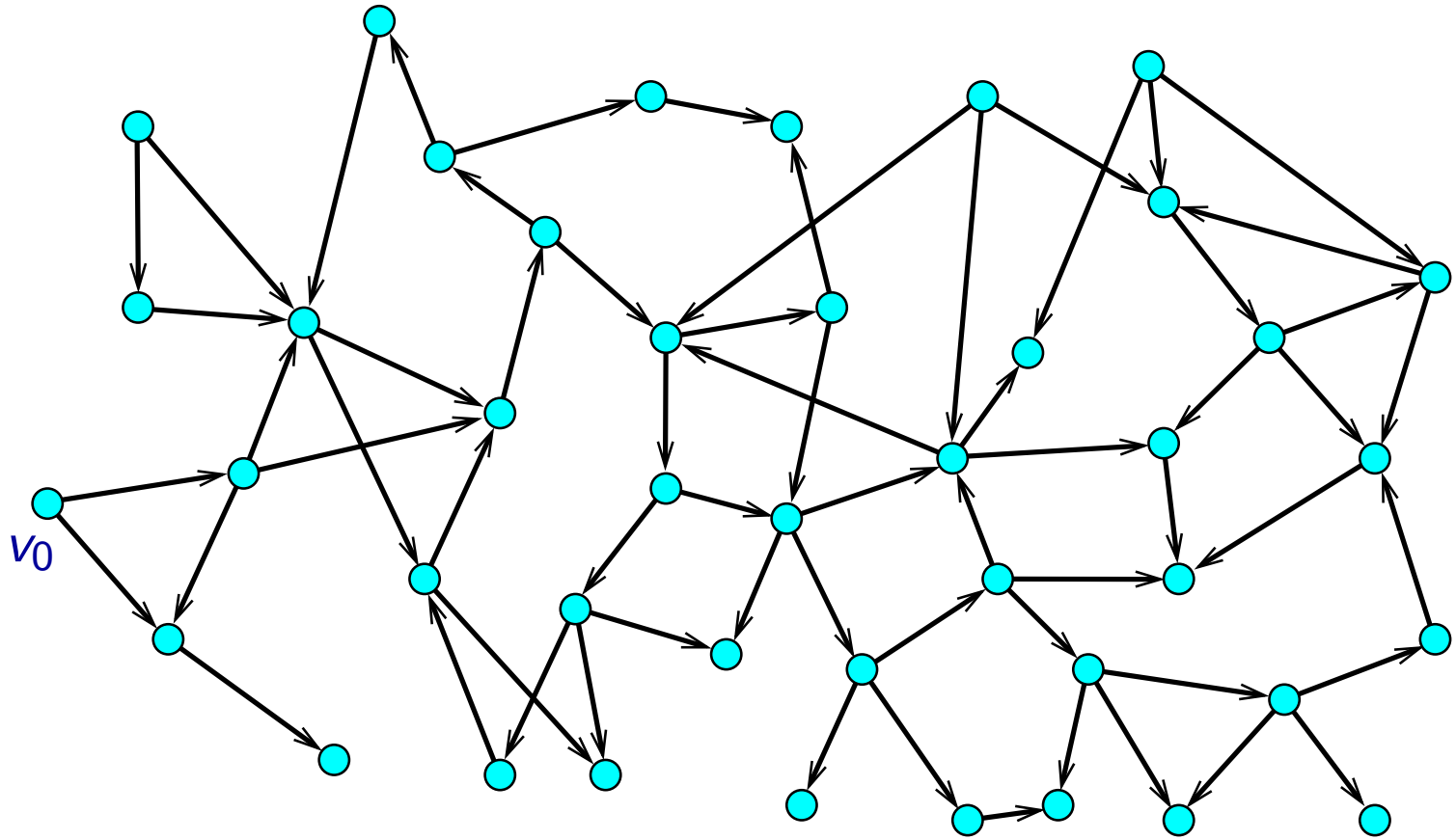
Reachability in a Graph

Let us assume a directed graph G with the set of nodes V the set of edges E.



Reachability in a Graph

Let us assume a directed graph G with the set of nodes V the set of edges E.



We would like to define the set of those nodes of G that are **reachable** from the given node v_0.

Reachability in a Graph

The set D, consisting of exactly those nodes that are reachable from the node v_0, can be defined by an inductive definition as follows:

- **Basis:** $v_0 \in D$
- **Inductive step:** If $u \in D$ and $(u, v) \in E$ then $v \in D$.

Lists

An inductive definition of a set can be sometimes formulated not in such a way that we separate some elements from an already defined set but rather is such a way that it gives a description how to **construct** elements of the given set.

Basically, we consider the existence of these elements as an additional assumption.

Lists

An inductive definition of a set can be sometimes formulated not in such a way that we separate some elements from an already defined set but rather is such a way that it gives a description how to **construct** elements of the given set.

Basically, we consider the existence of these elements as an additional assumption.

Example: Let us assume that we have some arbitrary set A.

We would like to define a set *Lists* consisting of all **lists** whose elements are arbitrary elements from this set A.

Lists

An inductive definition of a set can be sometimes formulated not in such a way that we separate some elements from an already defined set but rather is such a way that it gives a description how to **construct** elements of the given set.

Basically, we consider the existence of these elements as an additional assumption.

Example: Let us assume that we have some arbitrary set A.

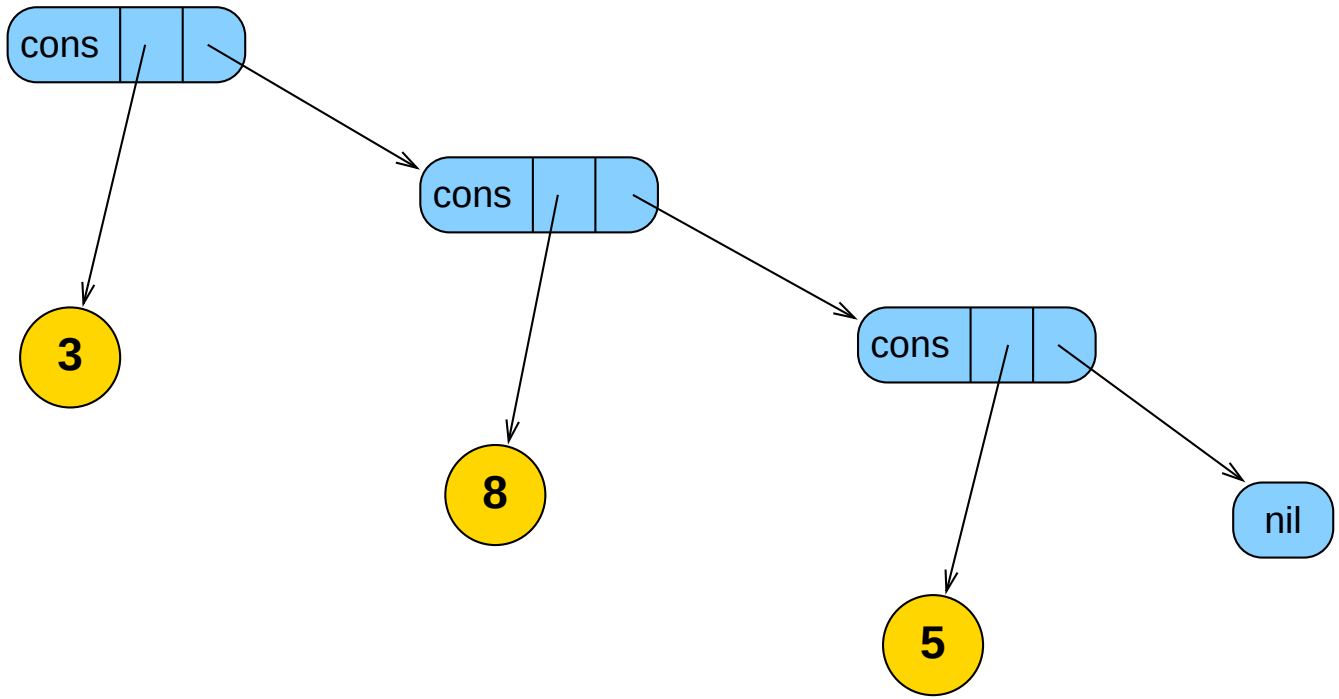
We would like to define a set *Lists* consisting of all **lists** whose elements are arbitrary elements from this set A.

This inductive definition of the set *Lists* defines two ways how to construct a list:

- **Basis:** There exists an element $nil \in Lists$.
- **Inductive step:** For each element $a \in A$ and each list $\ell \in Lists$ there exists a list $cons(a, \ell) \in Lists$.

Lists

An intuitive idea of a list consisting of elements $[3, 8, 5]$.
(We assume here that $A = \mathbb{N}$.)



$$cons(3, cons(8, cons(5, nil)))$$

Lists

The above given definition of the set *Lists* is incomplete. We must add assumptions that basically say that every list constructed by the two above given ways (*nil* and *cons*) is unique:

- For each $a \in A$ and $\ell \in \mathit{Lists}$ it holds that $\mathit{nil} \neq \mathit{cons}(a, \ell)$.
- For each $a, a' \in A$ and $\ell, \ell' \in \mathit{Lists}$ it holds that

$$\text{if} \quad \mathit{cons}(a, \ell) = \mathit{cons}(a', \ell') \quad \text{then} \quad a = a' \text{ and } \ell = \ell'.$$

Remark: In particular, this means that whenever we have an arbitrary list from the set *Lists* then we know that:

- it must be either *nil* or $\mathit{cons}(a, \ell)$ for some $a \in A$ and $\ell \in \mathit{Lists}$,
- if it is $\mathit{cons}(a, \ell)$ for some $a \in A$ and $\ell \in \mathit{Lists}$ then these elements a and ℓ are determined uniquely by this.

Lists

Example: An inductive definition of function $\textsc{Length} : \mathit{Lists} \to \mathbb{N}$

- **Basis:** $\textsc{Length}(\mathit{nil}) = 0$
- **Inductive step:** For each $a \in A$ and $\ell \in \mathit{Lists}$ is

$$\textsc{Length}(\mathit{cons}(a, \ell)) = 1 + \textsc{Length}(\ell).$$

Lists

Example: An inductive definition of function
$\textsc{Append} : \mathit{Lists} \times \mathit{Lists} \to \mathit{Lists}$

- **Basis:** For each $\ell' \in \mathit{Lists}$ is
$$\textsc{Append}(\mathit{nil}, \ell') = \ell'$$
- **Inductive step:** For each $a \in A$ and $\ell \in \mathit{Lists}$ is
$$\textsc{Append}(\mathit{cons}(a, \ell), \ell') = \mathit{cons}(a, \textsc{Append}(\ell, \ell')).$$

Lists

Proposition

For every pair of lists $\ell, \ell' \in \mathit{Lists}$ it holds that

$$\textsc{Length}(\textsc{Append}(\ell, \ell')) = \textsc{Length}(\ell) + \textsc{Length}(\ell').$$

Proof: By induction on ℓ:

- **Basis** ($\ell = \mathit{nil}$):

$$\begin{aligned}\textsc{Length}(\textsc{Append}(\mathit{nil}, \ell')) &= \textsc{Length}(\ell') \\ &= 0 + \textsc{Length}(\ell') \\ &= \textsc{Length}(\mathit{nil}) + \textsc{Length}(\ell')\end{aligned}$$

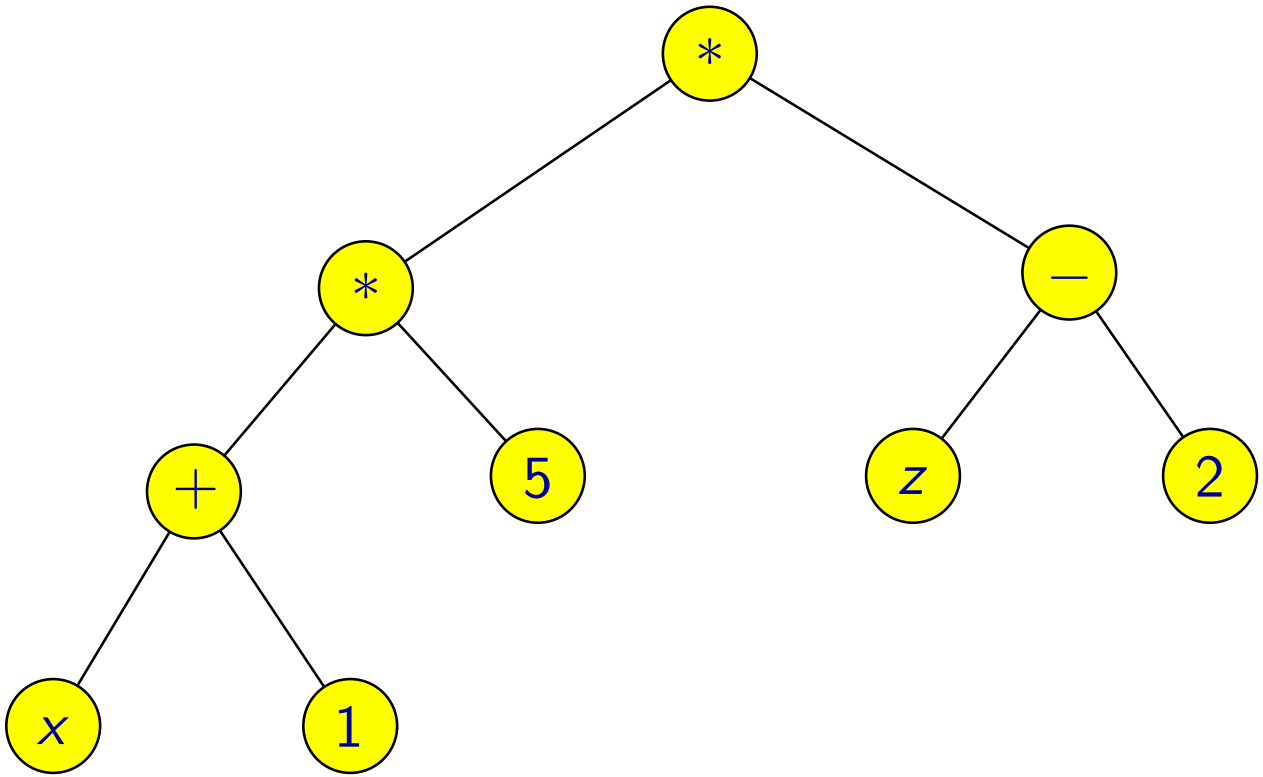
Lists

- **Inductive step** ($\ell = \mathit{cons}(a, \ell'')$ for some $a \in A$ and $\ell'' \in \mathit{Lists}$):

$$
\begin{aligned}
&\textsc{Length}(\textsc{Append}(\mathit{cons}(a, \ell''), \ell')) \\
&\qquad = \textsc{Length}(\mathit{cons}(a, \textsc{Append}(\ell'', \ell'))) \\
&\qquad = 1 + \textsc{Length}(\textsc{Append}(\ell'', \ell')) \\
&\qquad = 1 + (\textsc{Length}(\ell'') + \textsc{Length}(\ell')) \\
&\qquad = (1 + (\textsc{Length}(\ell'')) + \textsc{Length}(\ell') \\
&\qquad = (\textsc{Length}(\mathit{cons}(a, \ell'')) + \textsc{Length}(\ell')
\end{aligned}
$$

Abstract Syntax Trees

A structure of an arbitrary expression can be represented in a form of an **abstract syntax tree**.



Example: An abstract syntax tree for expression $((x + 1) * 5) * (z - 2)$

Abstract Syntax Trees

We would like to define a set of all well-formed abstract syntax trees of arithmetic expressions.

Let us assume that the following sets are already defined:

- $Var = \{x, y, z, \ldots\}$
- $Num = \{0, 1, 2, \ldots\}$
- $UnOp = \{-, \ldots\}$
- $BinOp = \{+, -, *, /, \ldots\}$

The set $Expr$ of all well-formed abstract syntax trees:

- For each $x \in Var$ there is a tree $var(x) \in Expr$.
- For each $n \in Num$ there is a tree $num(n) \in Expr$.
- For each $op \in UnOp$ and $e \in Expr$ there is a tree $unop(op, e) \in Expr$.
- For each $op \in BinOp$ and $e_1, e_2 \in Expr$ there is a tree $binop(op, e1, e2) \in Expr$.

Abstract Syntax Trees

- For each $x \in \mathit{Var}$, $n \in \mathit{Num}$, $\mathit{op}_1 \in \mathit{UnOp}$, $\mathit{op}_2 \in \mathit{BinOp}$, $e, e_1, e_2 \in \mathit{Expr}$ it holds that
  - $\mathit{var}(x) \neq \mathit{num}(n)$
  - $\mathit{var}(x) \neq \mathit{unop}(\mathit{op}_1, e)$
  - $\mathit{var}(x) \neq \mathit{binop}(\mathit{op}_2, e_1, e_2)$
  - $\mathit{num}(n) \neq \mathit{unop}(\mathit{op}_1, e)$
  - $\mathit{num}(n) \neq \mathit{binop}(\mathit{op}_2, e_1, e_2)$
  - $\mathit{unop}(\mathit{op}_1, e) \neq \mathit{binop}(\mathit{op}_2, e_1, e_2)$
- For each $x, x' \in \mathit{Var}$ it holds that if $\mathit{var}(x) = \mathit{var}(x')$ then $x = x'$.
- For each $n, n' \in \mathit{Num}$ it holds that if $\mathit{num}(n) = \mathit{num}(n')$ then $n = n'$.
- For each $\mathit{op}, \mathit{op}' \in \mathit{UnOp}$ and $e, e' \in \mathit{Expr}$ it holds that if $\mathit{unop}(\mathit{op}, e) = \mathit{unop}(\mathit{op}', e')$ then $\mathit{op} = \mathit{op}'$ and $e = e'$.
- For each $\mathit{op}, \mathit{op}' \in \mathit{BinOp}$ and $e_1, e_2, e_1', e_2' \in \mathit{Expr}$ it holds that if $\mathit{binop}(\mathit{op}, e_1, e_2) = \mathit{binop}(\mathit{op}', e_1', e_2')$ then $\mathit{op} = \mathit{op}'$, $e_1 = e_1'$, and $e_2 = e_2'$.

Natural Numbers

In a similar way we can also define the set of **natural numbers**.

The definition of the set Nat:

- **Basis:** There exists an element $zero \in Nat$.
- **Inductive step:** For each element $x \in Nat$ there exists an elementent $succ(x) \in Nat$.

We also assume that:

- For each $x \in Nat$ it holds that $succ(x) \neq zero$.
- For each $x, x' \in Nat$ it holds that if $succ(x) = succ(x')$ then $x = x'$.

Natural Numbers

Basically this means that the individual natural numbers are the elements denoted by the following expressions:

$$
\begin{array}{ll}
0 & \mathit{zero} \\
1 & \mathit{succ}(\mathit{zero}) \\
2 & \mathit{succ}(\mathit{succ}(\mathit{zero})) \\
3 & \mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{zero}))) \\
4 & \mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{zero})))) \\
5 & \mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{zero}))))) \\
6 & \mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{zero})))))) \\
7 & \mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{succ}(\mathit{zero}))))))) \\
 & \vdots
\end{array}
$$

Natural Numbers

An inductive definition of function $\textsc{Plus} : \mathit{Nat} \times \mathit{Nat} \to \mathit{Nat}$ — the induction proceeds here by the second argument:

- **Basis:** For each $x \in \mathit{Nat}$ it holds that
$$\textsc{Plus}(x, \mathit{zero}) = x$$
- **Inductive step:** For each $x, y \in \mathit{Nat}$ it holds that
$$\textsc{Plus}(x, \mathit{succ}(y)) = \mathit{succ}(\textsc{Plus}(x, y))$$

Natural Numbers

An inductive definition of function $\textsc{Plus} : \mathit{Nat} \times \mathit{Nat} \to \mathit{Nat}$ — the induction proceeds here by the second argument:

- **Basis:** For each $x \in \mathit{Nat}$ it holds that
$$\textsc{Plus}(x, \mathit{zero}) = x$$
- **Inductive step:** For each $x, y \in \mathit{Nat}$ it holds that
$$\textsc{Plus}(x, \mathit{succ}(y)) = \mathit{succ}(\textsc{Plus}(x, y))$$

Example:
$$\begin{aligned}\textsc{Plus}(5,3) &= \mathit{succ}(\textsc{Plus}(5,2)) \\ &= \mathit{succ}(\mathit{succ}(\textsc{Plus}(5,1))) \\ &= \mathit{succ}(\mathit{succ}(\mathit{succ}(\textsc{Plus}(5,0)))) \\ &= \mathit{succ}(\mathit{succ}(\mathit{succ}(5))) = 8\end{aligned}$$

Natural Numbers

Proposition

For each $x, y, z \in \mathit{Nat}$ it holds that

$$\textsc{Plus}(x, \textsc{Plus}(y, z)) = \textsc{Plus}(\textsc{Plus}(x, y), z)$$

Proof: By induction on z.

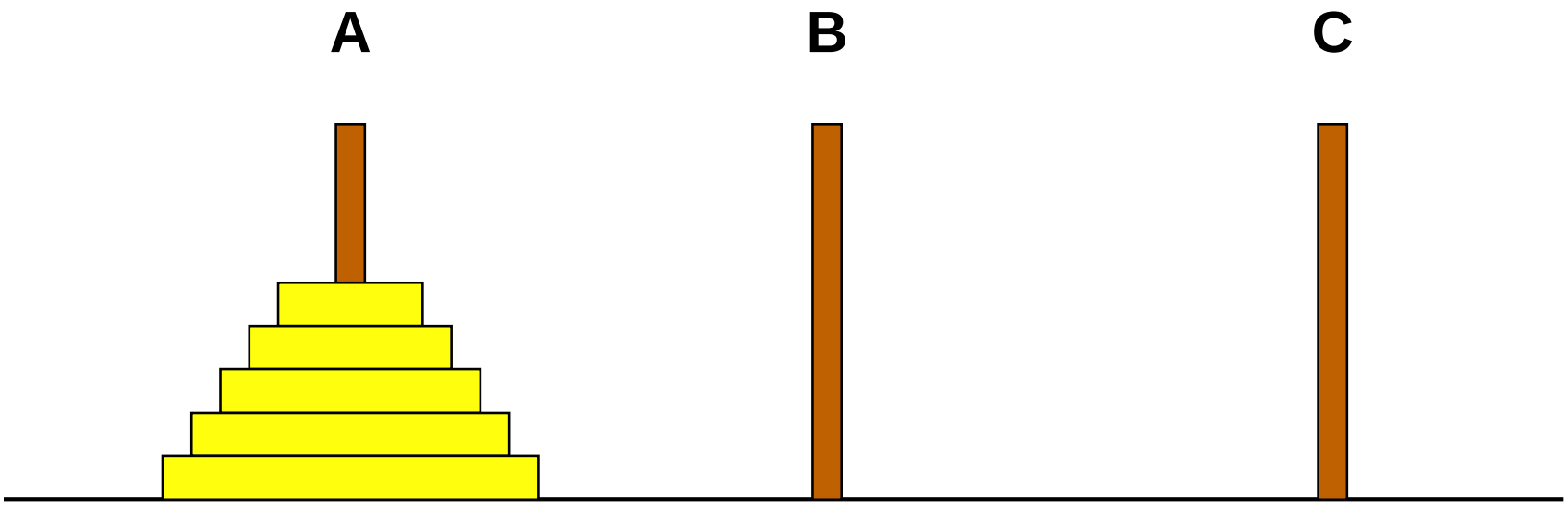
- **Basis:**

$$\textsc{Plus}(x, \textsc{Plus}(y, \mathit{zero})) = \textsc{Plus}(x, y) = \textsc{Plus}(\textsc{Plus}(x, y), \mathit{zero})$$

- **Inductive step:**

$$\begin{aligned}
\textsc{Plus}(x, \textsc{Plus}(y, \mathit{succ}(z))) &= \textsc{Plus}(x, \mathit{succ}(\textsc{Plus}(y, z))) \\
&= \mathit{succ}(\textsc{Plus}(x, \textsc{Plus}(y, z))) \\
&= \mathit{succ}(\textsc{Plus}(\textsc{Plus}(x, y), z)) \\
&= \textsc{Plus}(\textsc{Plus}(x, y), \mathit{succ}(z)))
\end{aligned}$$

Towers of Hanoi



Task: To move the disks from A to B in such a way that:

- In one moment we can move only one disk.
- It is not allowed to place a bigger disk on top of a smaller disk.

Towers of Hanoi

$n = 1$:

$A \to B$

Towers of Hanoi

$n = 1$:

$A \to B$

$n = 2$:

$A \to C$

$A \to B$

$C \to B$

Towers of Hanoi

$n = 1$:

$A \to B$

$n = 2$:

$A \to C$
$A \to B$
$C \to B$

$n = 3$:

$A \to B$
$A \to C$
$B \to C$
$A \to B$
$C \to A$
$C \to B$
$A \to B$

Towers of Hanoi

$n = 1$:

$A \to B$

$n = 2$:

$A \to C$
$A \to B$
$C \to B$

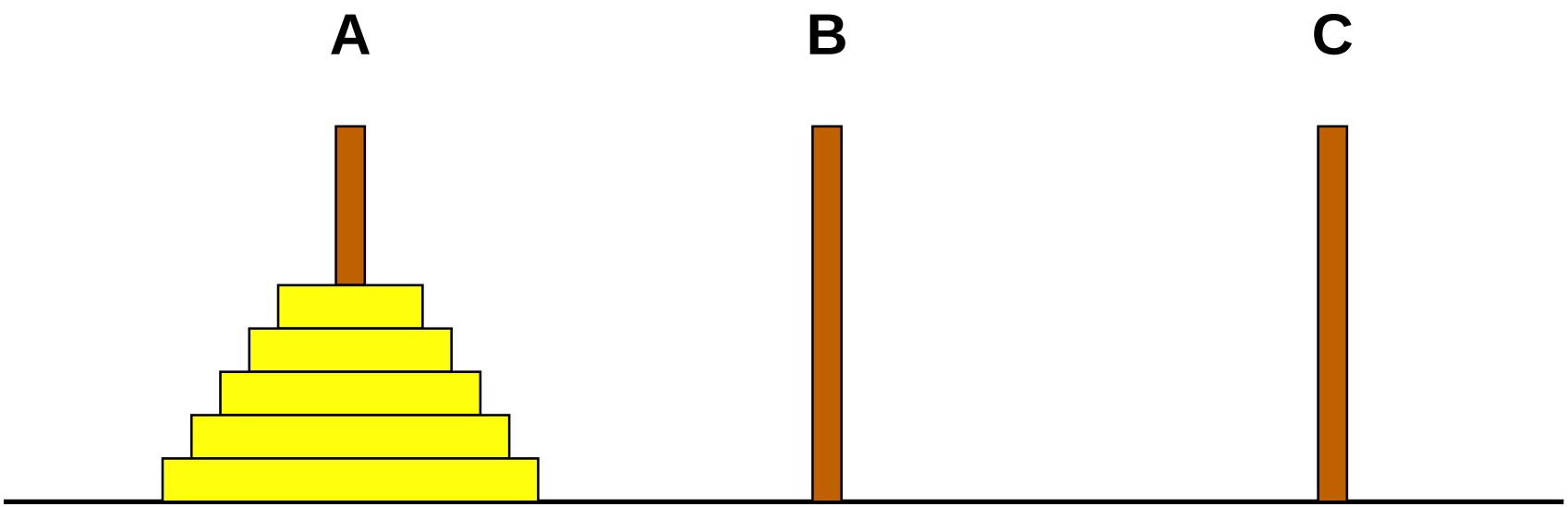
$n = 3$:

$A \to B$
$A \to C$
$B \to C$
$A \to B$
$C \to A$
$C \to B$
$A \to B$

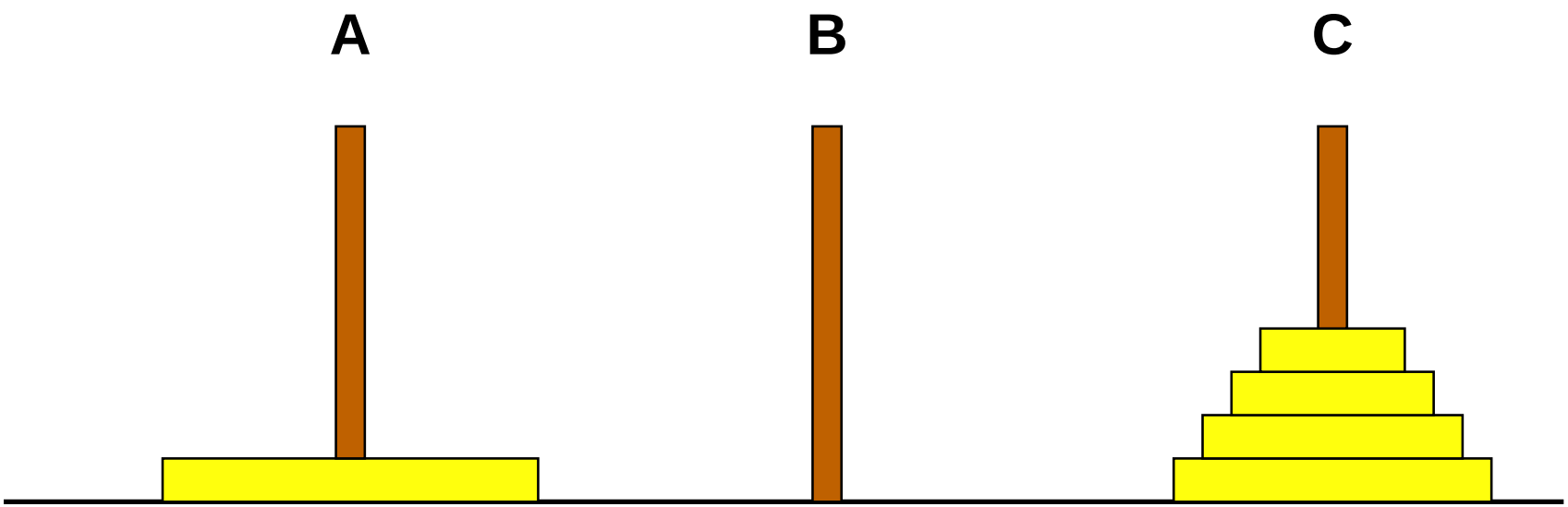
$n = 4$:

$A \to C$
$A \to B$
$C \to B$
$A \to C$
$B \to A$
$B \to C$
$A \to C$
$A \to B$
$C \to B$
$C \to A$
$B \to A$
$C \to B$
$A \to C$
$A \to B$
$C \to B$

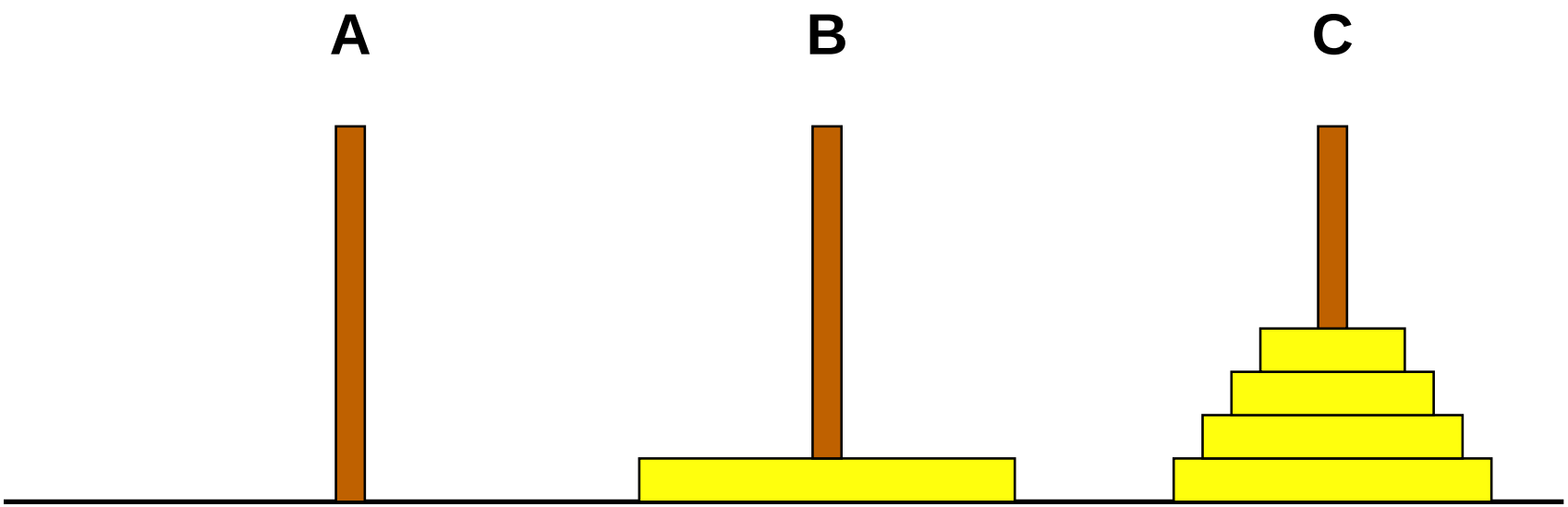
Towers of Hanoi



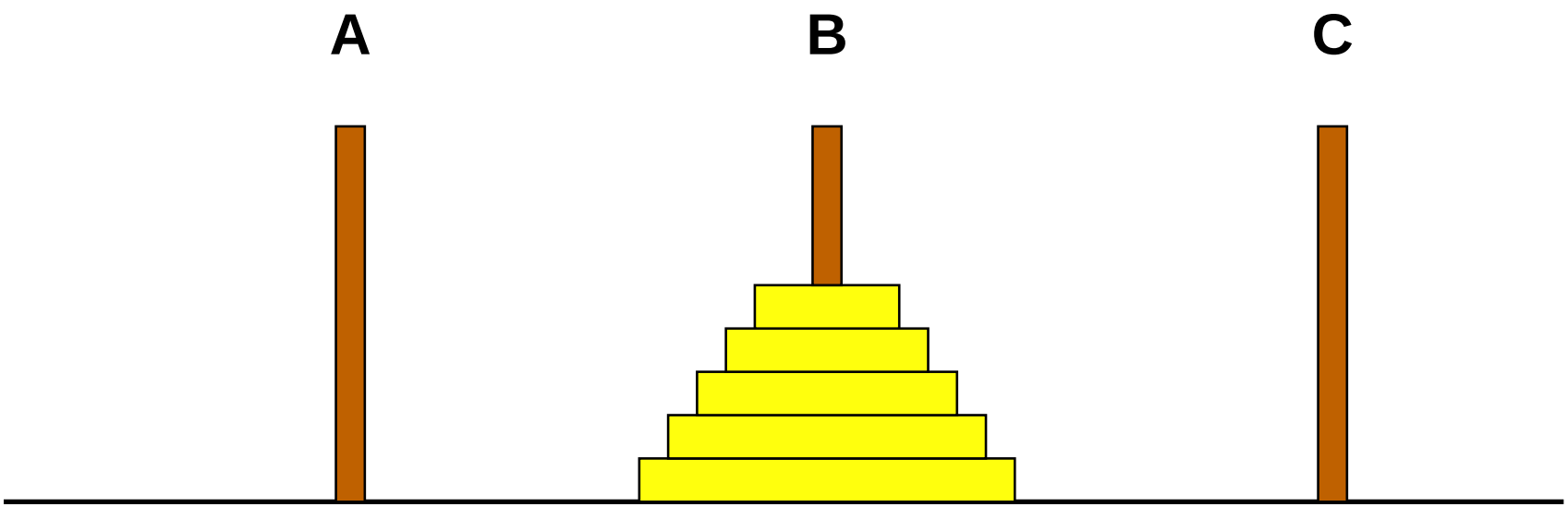
Towers of Hanoi



Towers of Hanoi



Towers of Hanoi



Towers of Hanoi

```
void hanoi(int n, char src, char dst, char tmp)
{
    if (n == 0) return;
    hanoi(n-1, src, tmp, dst);
    printf("%c -> %c\n", src, dst);
    hanoi(n-1, tmp, dst, src);
}

int main()
{
    int n;
    scanf("%d", &n);
    hanoi(n, 'A', 'B', 'C');
    return 0;
}
```

Towers of Hanoi

Let $P(n)$ denote the number of moves performed by the algorithm for n disks.

Proposition

$P(n) = 2^n - 1$.

Proof:

- For $n = 0$: $P(n) = 0 = 2^0 - 1$
- For $n > 0$: We can assume that $P(n-1) = 2^{n-1} - 1$.

$$\begin{aligned} P(n) &= 2P(n-1) + 1 = \\ &= 2(2^{n-1} - 1) + 1 = \\ &= 2 \cdot 2^{n-1} - 2 + 1 = \\ &= 2^n - 1 \end{aligned}$$

Towers of Hanoi

Proposition

To move n disks requires at least $2^n - 1$ moves.

Proof:

By induction.

So the given algorithm finds the optimal solution.

Remark

Question: How long it takes to move 64 disks if moving one disk takes 1 s ?

Towers of Hanoi

Proposition

To move n disks requires at least $2^n - 1$ moves.

Proof:

By induction.

So the given algorithm finds the optimal solution.

Remark

Question: How long it takes to move 64 disks if moving one disk takes 1 s ?

Answer: 18446744073709551615 s, i.e., about 585 billions of years.