CDM

Relations

Klaus Sutner

Carnegie Mellon University

30-relations 2017/12/15 23:22

1. Relations

- Operations and Properties
- Orders
- Equivalence Relations
- Closures

We have seen how to express general concepts (or properties) as sets: we form the set of all objects that "fall under" the concept (in Frege's terminology). Thus we can form the set of natural numbers, of primes, of reals, of continuous functions, of stacks, of syntactically correct C-programs and so on.

Another fundamental idea is to consider relationships between two or more objects. Here are some typical examples:

- divisibility relation on natural numbers,
- less-than relation on integers,
- greater-than relation on rational numbers,
- the "attends course" relation for students and courses,
- the "is prerequisite" relation for courses,
- the "is a parent of" relation for humans,
- the "terminates on input and produces output" relation for programs and inputs and outputs.

Let's focus on the binary case where two objects are associated, though not necessarily from the same domain.

The unifying characteristic is that we have some property (attribute, quality) that can hold or fail to hold of any two objects from the appropriate domains.

Standard notation: for a relation P and suitable objects a and b write $P(a, b)$ for the assertion that P holds on a and b.

Thus we can associate a truth value with $P(a, b)$: the assertion holds if a and b are indeed related by P, and is false otherwise.

For example, if P denotes the divisibility relation on the integers then $P(3, 9)$ holds whereas $P(3, 10)$ is false.

Later we will be mostly interested in relations where we can effectively determine whether $P(a, b)$ holds, but for the time being we will consider the general case.

Following our program of representing all important notions in terms of sets, here is a definition that expresses relations as sets.

Definition

A binary relation ρ from A to B is a subset of $A \times B$.

A is the domain of ρ, and B is the codomain of ρ.

We write $\mathsf{Rel}_{A,B}$ for the collection of all relations from A to B.

Notation

We will often write $\rho : A \to B$ instead of $\rho \subseteq A \times B$.

Also, it is customary to use infix notation $x\ \rho\ y$ or prefix notation $\rho(x, y)$ instead of the set-theoretic $(x, y) \in \rho$.

For example, the standard order on the naturals will be indicated by $2 < 5$ rather than $<(2, 5)$.

An important special case arises when the domain and codomain are the same.

Definition

A square relation (or an endorelation) is a binary relation with the same domain and codomain. The domain is called the carrier set or underlying set of the relation.

We write Rel_A for the collection of square relations on A. On occasion, we will simply refer to these relations as binary relations on A.

Example

The standard order relation $<$ on the natural numbers is a square relation.

Typical counterexamples: the relation "r is a root of polynomial $p(x)$" or "string s is an element of stack S".

Our definition is the standard one, but note that we could also represent a relation by a map

$$A \times B \to \{\mathsf{ff}, \mathsf{tt}\}$$

that associates pairs of objects with truth values. In other words, we identity a relation with its characteristic function. This is somewhat closer to common usage in programming languages and also has the advantage that we can draw pictures, at least for reasonably small domains.

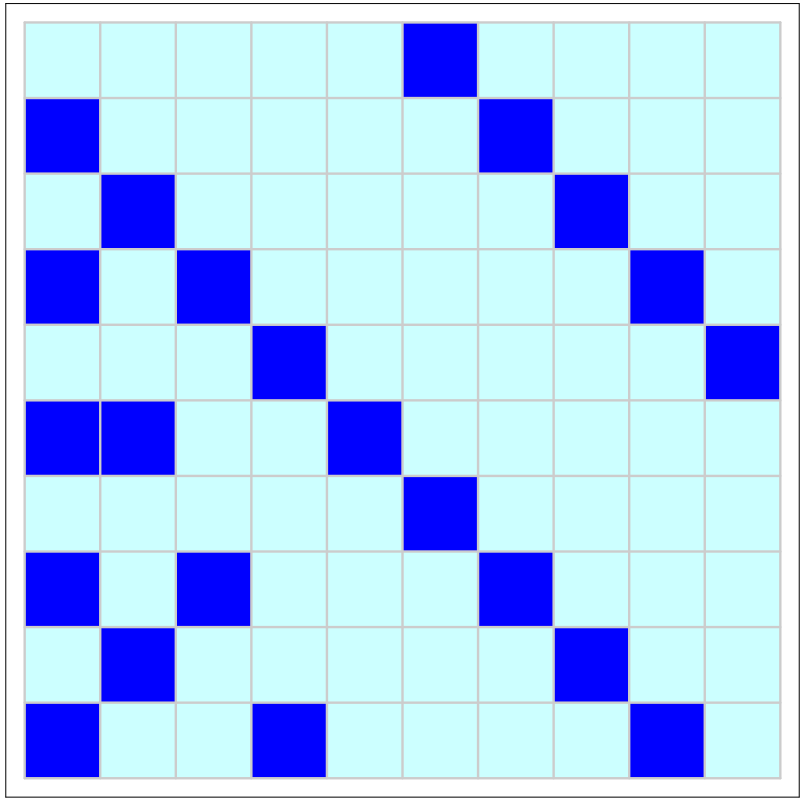
Suppose ρ is a relation on $[n]$ (or, more generally, a finite set of size n). We interpret ρ as a Boolean matrix, a 2-dimensional, n by n array $R : [n] \times [n] \to \mathbb{B}$ of Booleans such that

$$i \; \rho \; j \iff R[i, j] = \mathsf{tt}.$$

On occasion, these picture give some insight into properties of the relation.

A Random Relation

Here is the picture for a "random" relation ρ on $[10]$.



Of course, ρ is far from random, in fact

$$x \ \rho \ y \iff x \neq y \wedge (y+1 \mid x \vee x+5 \mid y)$$

Another way to think about $\rho \subseteq A \times B$ is as a set-valued map

$$R : A \to \mathfrak{P}(B)$$

that associates every object in the domain with a set of objects in the codomain in the obvious way:

$$R(a) = \{\, b \in B \mid a \;\rho\; b \,\}.$$

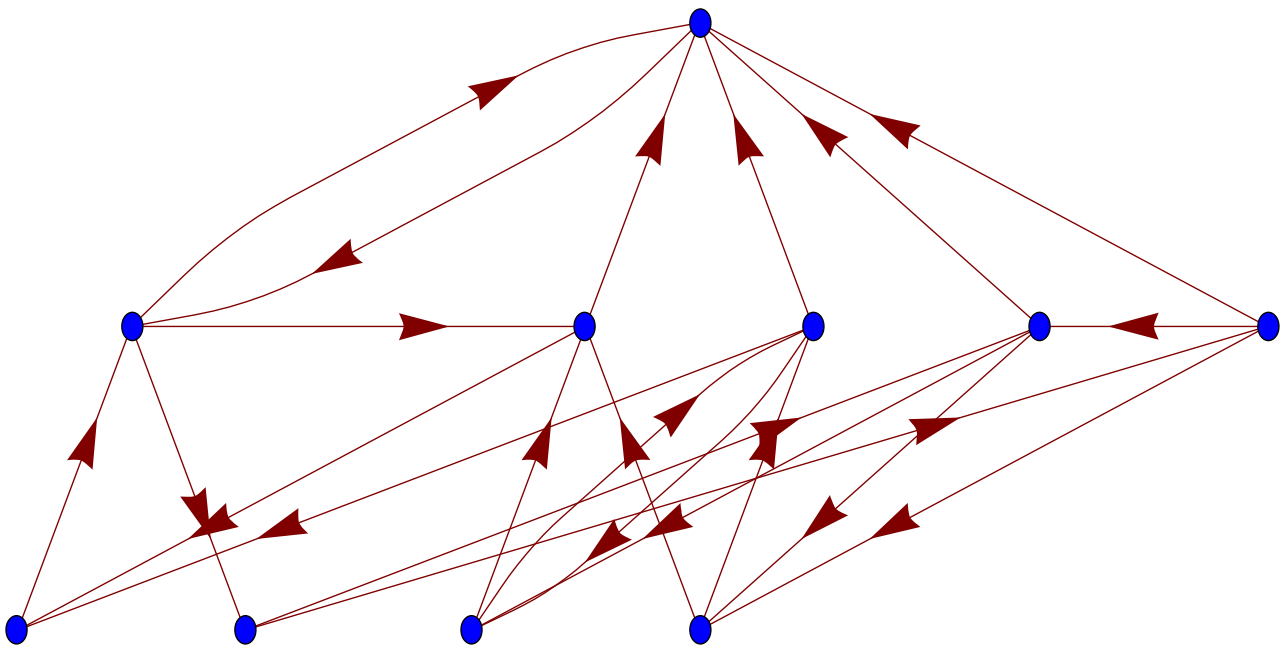
E.g., for the last relation we obtain

a	$R(a)$	a	$R(a)$
1	$\{6\}$	6	$\{1, 2, 5\}$
2	$\{1, 7\}$	7	$\{6\}$
3	$\{2, 8\}$	8	$\{1, 3, 7\}$
4	$\{1, 3, 9\}$	9	$\{2, 8\}$
5	$\{4, 10\}$	10	$\{1, 4, 9\}$

This "array of lists" representation is very useful in a number of algorithms.

Moreover, it can be visualized by a "shoelace" picture where we plot A in the left column, B in the right and connect a with b by a line if $a\ \rho\ b$.

The following picture is for the relation ρ from above.



Here is yet another way to think about a square relation $\rho \subseteq A \times A$.

Definition (Directed Graphs)

A directed graph (or digraph) is a structure $G = \langle V, E \rangle$ where

- V is a set of vertices (or nodes, points)
- $E \subseteq V \times V$ is a set of edges (or arcs, lines)

Edges are usually written as ordered pairs (u, v): $u \in V$ is the source, and $v \in V$ the target of the edge. Of course, a digraph is logically the same as a binary relation, except that the carrier set has been spelled out explicitly.

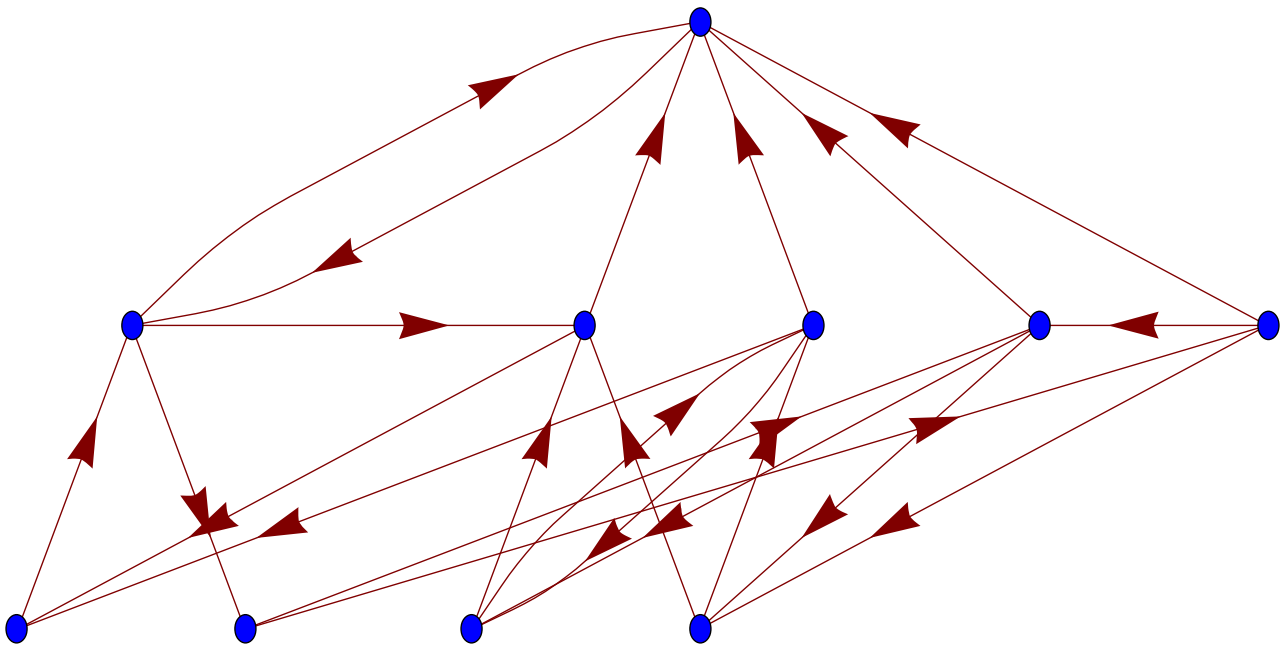
Because graph theory is an independent field of study, and historically has produced different results than the study of binary relations. Recall: Thinking about a problem just the right way is crucial in CS.

Most notably, the vertices and edges in a graph are often labeled. In particular edge labels are very important in automata theory.

Another important difference is that researchers in graph theory have spent considerable time and effort on finding ways to represent graphs as sets of points in some suitable geometric space.

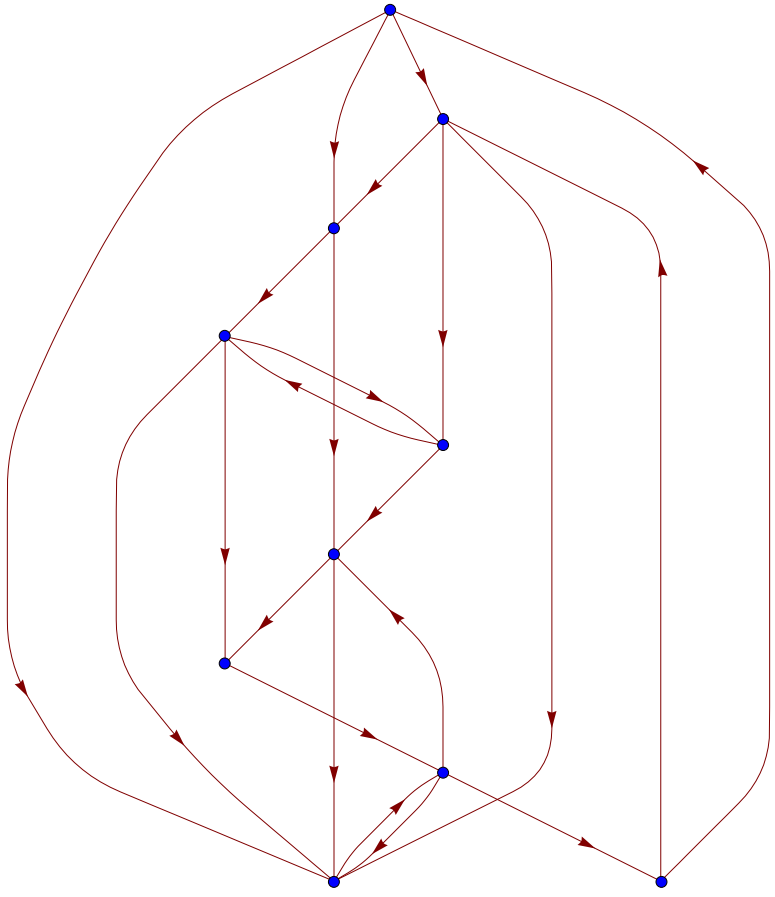
For example, we might use points in the plane or in 3-space, with edges indicated by lines or curves.

Identifying points and edges with geometric constructs is very important in visualization – but it is very hard to do right by algorithm (and also difficult to do by hand if the graphs are large).

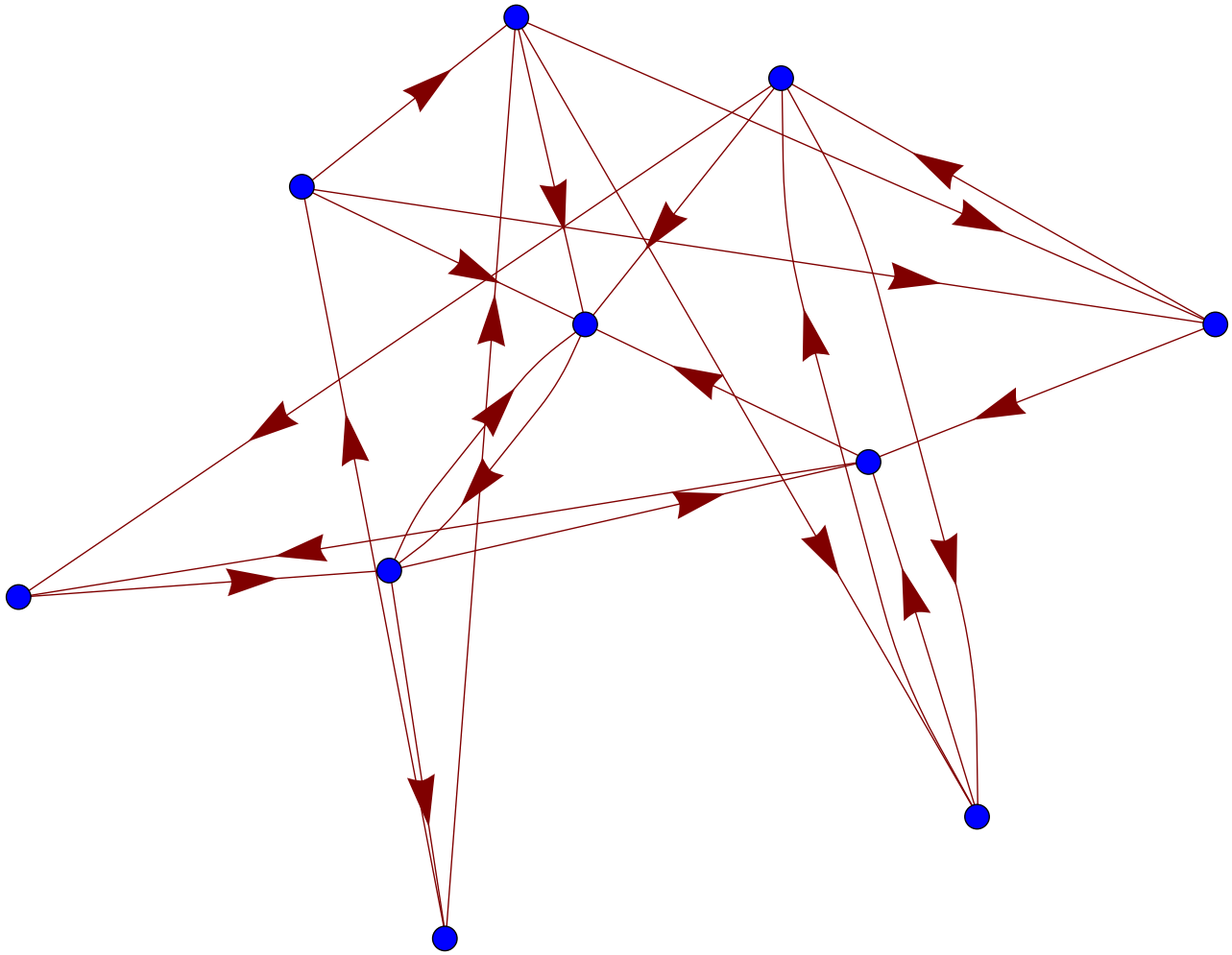


This particular layout was generated automatically.

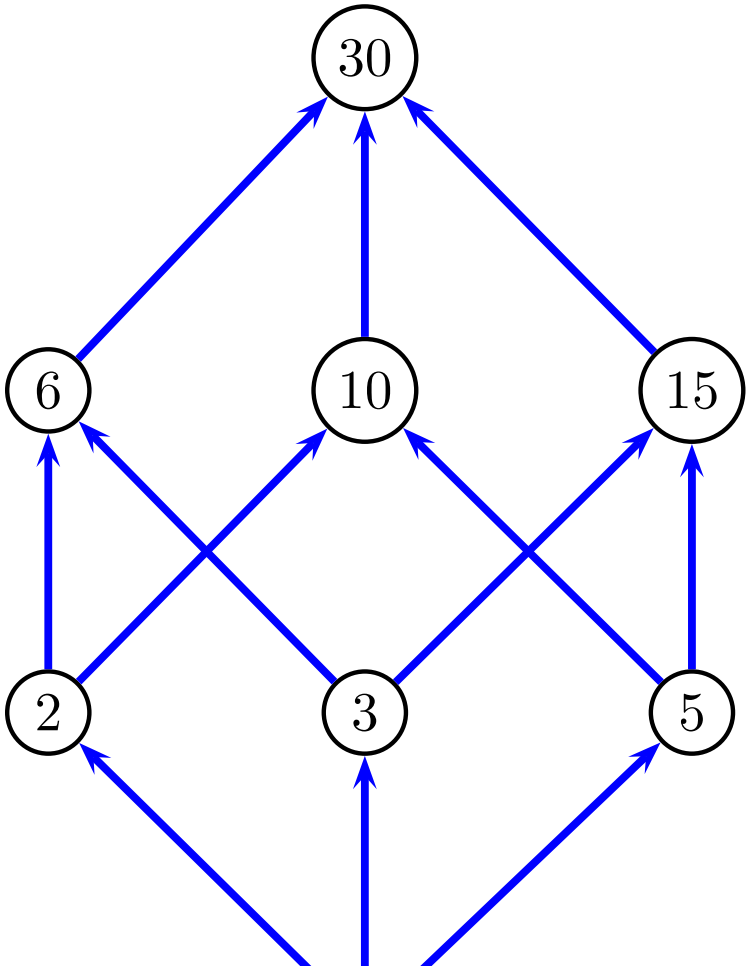
Careful, though. The drawing represents the graph, but it is by no means canonical: there are many alternative ways to draw the same graph.



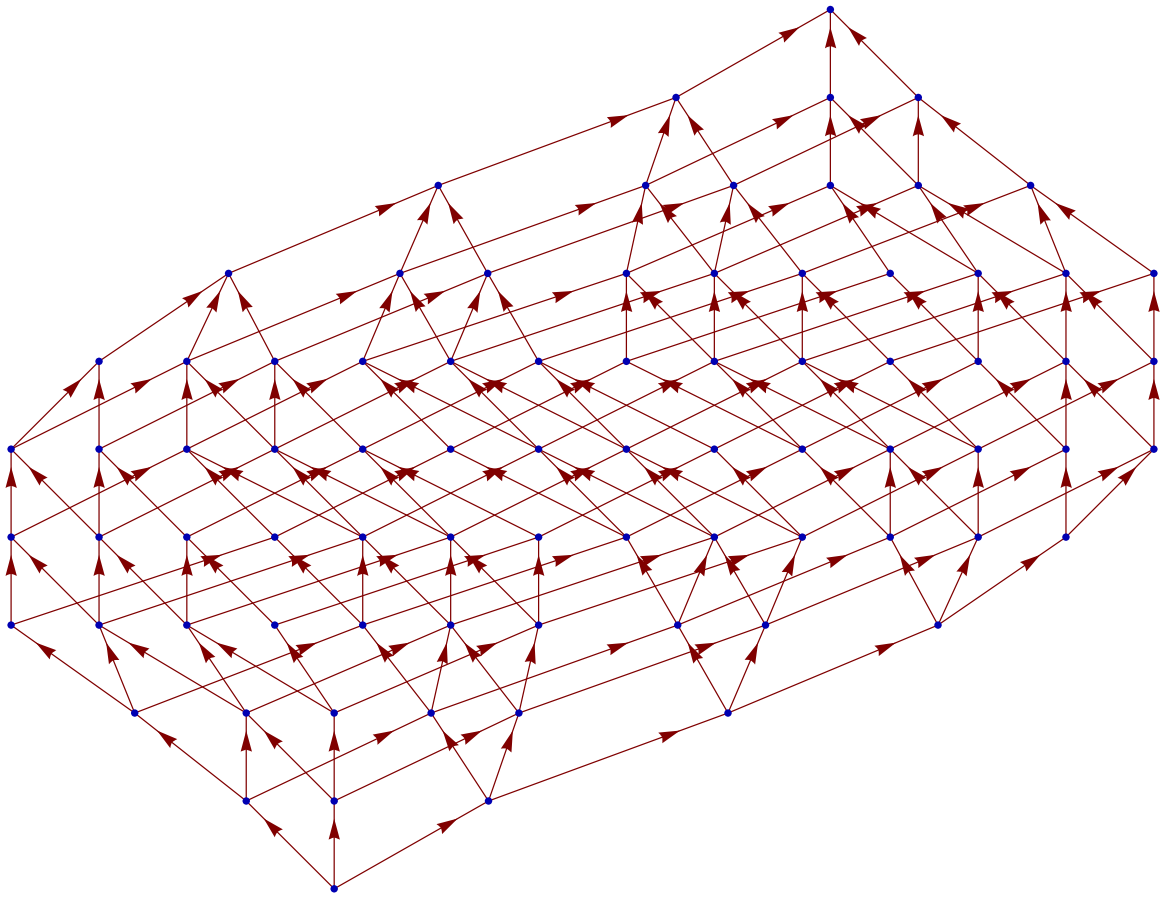
Another picture for the same graph.



Careful selection of the vertex coordinates can help to convey a lot of information about the structure of the relation. The next picture was constructed painstakingly by hand.



Here is the automatically generated picture for the divisor lattice of 148176.



We focus on square relations on ground set A:

Definition

- $I_A = \{ (x, x) \mid x \in A \}$, the identity or diagonal relation.
- $U_A = A \times A$, the universal relation.
- $\emptyset_A = \emptyset$, the empty relation.

Exercise

What would the pictures for these relations look like in the various models above?

Exercise

Are there any other natural relations on A that one might reserve special notation for? Why?

Lemma

If $|A| = n$ and $|B| = m$, then there are 2^{nm} relations from A to B.

Proof. The collection of all relations from A to B is simply the powerset of $A \times B$. The latter has size mn, and our claim follows. □

Another way to think about this: we can represent each relation ρ by an n by m Boolean matrix. There are $n \cdot m$ bits, each can be on or off: 2^{nm} possibilities.

This result also holds in the infinite case (but requires a little cardinal arithmetic to state and prove).

Exercise

Count relations based on the set-valued function representation.

- Relations

2. Operations and Properties

- Orders

- Equivalence Relations

- Closures

Since relations express properties (of pairs of objects) it is natural to consider logical connections between them.

Definition

Let ρ and σ be two relations from A to B. ρ is finer than σ (or σ is coarser that ρ) if $x\ \rho\ y$ implies $x\ \sigma\ y$.

In symbols:

$$\rho \sqsubseteq \sigma$$

If we think of relations as subsets of the Cartesian product $A \times B$ this simply says that $\rho \subseteq \sigma$. Of course, most relations are not comparable in this sense: neither one will be finer than the other.

Exercise

Which of the knight's move relations should be expect to be finer than another? Look at the pictures, but don't rely on them.

Definition

Let ρ and σ be two relations from A to B.

The join of ρ and σ is defined by $x\ (\rho \sqcup \sigma)\ y \iff x\ \rho\ y \vee x\ \sigma\ y$.

The meet of ρ and σ is defined by $x\ (\rho \sqcap \sigma)\ y \iff x\ \rho\ y \wedge x\ \sigma\ y$.

The negation or complement of ρ is defined by $x\ \rho^{-}\ y \iff \neg x\ \rho\ y$.

Note that negation is an involution: $\rho^{--} = \rho$.

Example

Consider the usual order relations on $\mathbb{N}$. Then

$$\leq\ =\ <\ \sqcup\ I_{\mathbb{N}}$$
$$<\ =\ \leq\ \sqcap\ I_{\mathbb{N}}^{-}$$

If we think of relations from A to B as subsets of the Cartesian product the Boolean operations on relations translate into Boolean operations on sets.

$$\rho \sqcup \sigma = \rho \cup \sigma$$
$$\rho \sqcap \sigma = \rho \cap \sigma$$
$$\rho^{-} = A \times B - \rho$$

But note that this is a representation issue: if we were to implement the relations as digraphs or Boolean matrices things work out differently.

Exercise

Explain negation ρ^{-} in terms of $\sqsubseteq$.

Definition

Let $\rho : A \to B$ be a relation. The converse of ρ is a relation from B to A defined by

$$x \; \rho^c \; y \iff y \; \rho \; x.$$

Thus the domain/codomain of ρ^c is the codomain/domain of ρ. Don't confuse this operation with complement.

Note that the converse operation is an involution: $(\rho^c)^c = \rho$.

In the set-theoretic interpretation we simply reverse the pairs:

$$\rho^c = \{ (y, x) \mid (x, y) \in \rho \}$$

Clearly, ρ is symmetric iff $\rho^c = \rho$.

The converse is also written ρ^{-1} on occasion, the same notation used for inverse functions. We will avoid this notation since the converse of any relation ρ is always well-defined, the inverse for functions is not.

Exercise

Show that $(\rho \sqcap \sigma)^c = \rho^c \sqcap \sigma^c$.

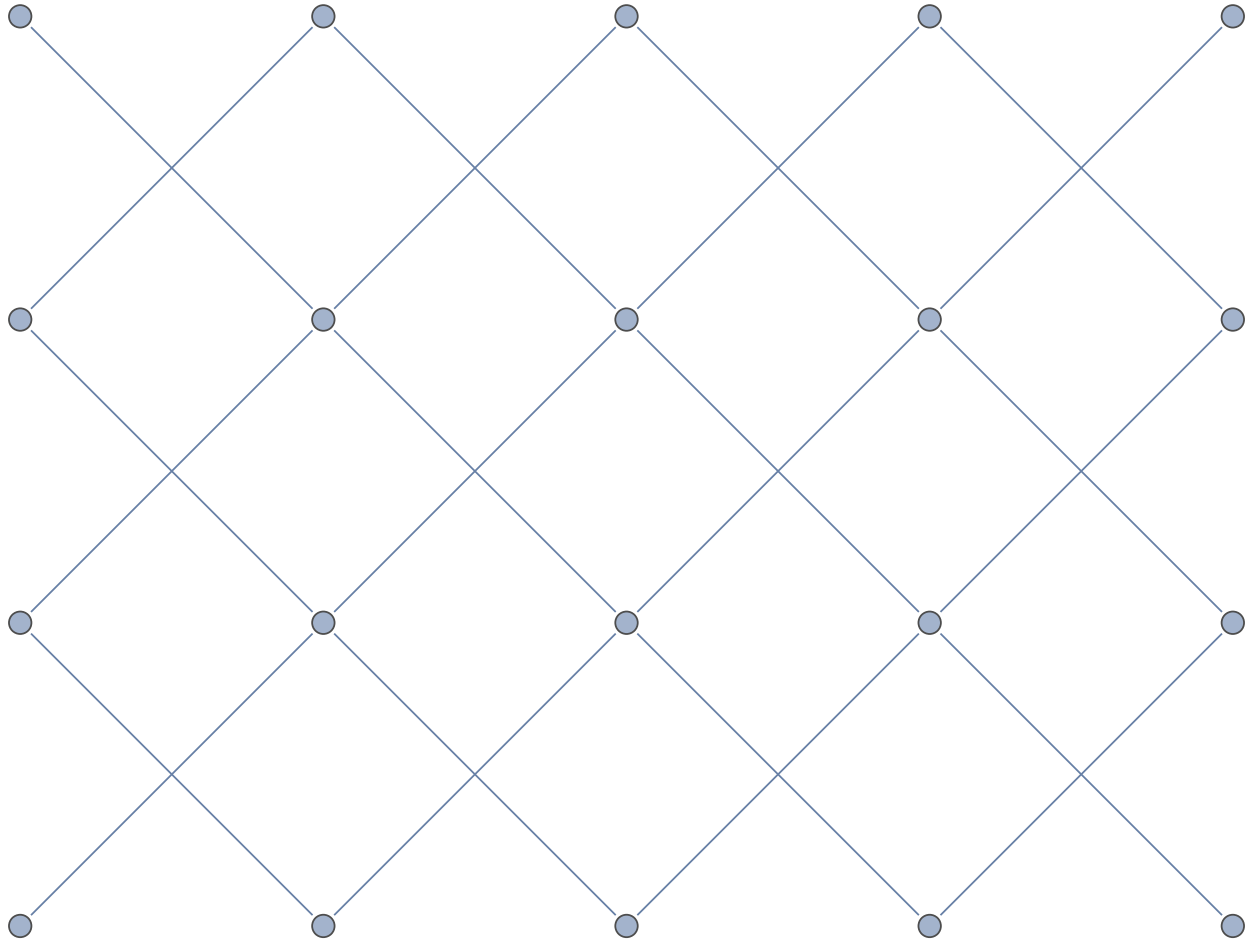
Definition

Let $\rho : A \to A$ and $\sigma : B \to B$ be two relations. Their direct product $\rho \times \sigma$ is a relation from $A \times B$ to $A \times B$ defined by

$$(a, b)\ \rho \times \sigma\ (a', b') \iff a\ \rho\ a' \wedge b\ \sigma\ b'.$$

This operation is also called the tensor product or even the Kronecker product. The reason for the latter terminology is that

$$\mathsf{bm}(\rho \times \sigma) = \mathsf{bm}(\rho) \otimes \mathsf{bm}(\sigma).$$



$P_5 \times P_4$

Kronecker Product

Suppose we have an $n \times m$ matrix A and an $n' \times m'$ matrix B over some suitable algebraic structure.

Their Kronecker product is the $nn' \times mm'$ matrix $C = A \otimes B$ defined by (we assume 0-indexing):

$$C(i,j) = A(i \operatorname{div} n, j \operatorname{div} m) \cdot B(i \bmod n', j \bmod m')$$

For example, for A 2×2 and B 3×3 we get a 6×6 matrix consisting of 4 blocks:

$$\left(\begin{array}{c|c} a_{0,0}B & a_{0,1}B \\ \hline a_{1,0}B & a_{1,1}B \end{array}\right) =$$

$$\left(\begin{array}{ccc|ccc} a_{0,0}b_{0,0} & a_{0,0}b_{0,1} & a_{0,0}b_{0,2} & a_{0,1}b_{0,0} & a_{0,1}b_{0,1} & a_{0,1}b_{0,2} \\ a_{0,0}b_{1,0} & a_{0,0}b_{1,1} & a_{0,0}b_{1,2} & a_{0,1}b_{1,0} & a_{0,1}b_{1,1} & a_{0,1}b_{1,2} \\ a_{0,0}b_{2,0} & a_{0,0}b_{2,1} & a_{0,0}b_{2,2} & a_{0,1}b_{2,0} & a_{0,1}b_{2,1} & a_{0,1}b_{2,2} \\ \hline a_{1,0}b_{0,0} & a_{1,0}b_{0,1} & a_{1,0}b_{0,2} & a_{1,1}b_{0,0} & a_{1,1}b_{0,1} & a_{1,1}b_{0,2} \\ a_{1,0}b_{1,0} & a_{1,0}b_{1,1} & a_{1,0}b_{1,2} & a_{1,1}b_{1,0} & a_{1,1}b_{1,1} & a_{1,1}b_{1,2} \\ a_{1,0}b_{2,0} & a_{1,0}b_{2,1} & a_{1,0}b_{2,2} & a_{1,1}b_{2,0} & a_{1,1}b_{2,1} & a_{1,1}b_{2,2} \end{array}\right)$$

Kronecker Product Properties

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C$$

$$A \otimes (B + C) = A \otimes B + A \otimes C$$

$$(B + C) \otimes A = B \otimes A + C \otimes A$$

$$(A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D)$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

$$\mathsf{rank}(A \otimes B) = \mathsf{rank}(A)\ \mathsf{rank}(B)$$

Definition

Let $\rho : A \to A$ and $\sigma : B \to B$ be two relations. Their Cartesian product $\rho \oplus \sigma$ is a relation from $A \times B$ to $A \times B$ defined by

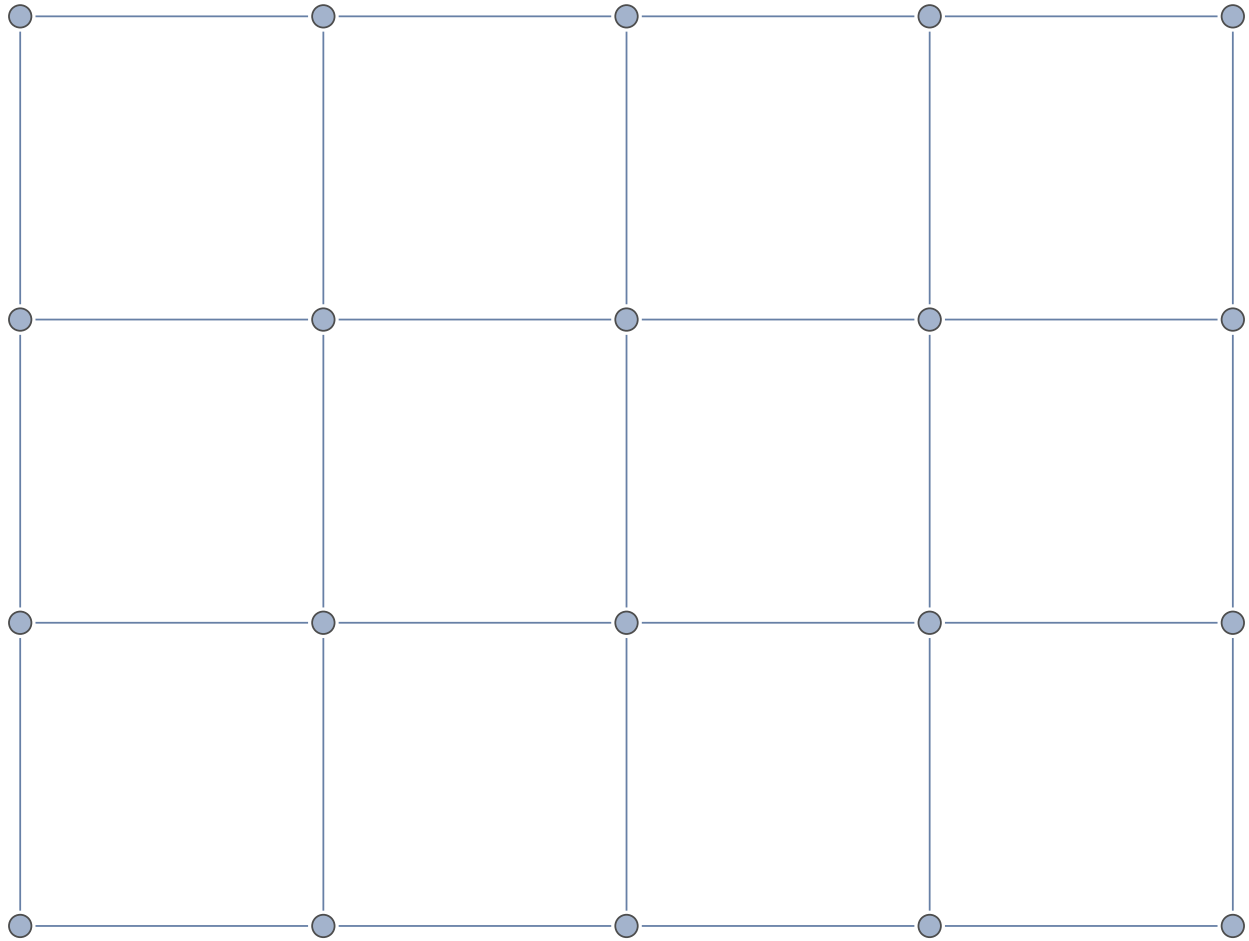
$$(a, b)\ \rho \oplus \sigma\ (a', b') \iff (a\ \rho\ a' \wedge b = b') \vee (a = a' \wedge b\ \sigma\ b').$$

As with the direct product, the carrier set here is the Cartesian set product $A \times B$, but this operation is not categorical.

In terms of matrices we have

$$\mathsf{bm}(\rho \oplus \sigma) = \mathsf{bm}(\rho) \otimes I + I \otimes \mathsf{bm}(\sigma).$$

where the identity matrices are chosen properly for the sum to work.



$P_5 \oplus P_4$

Definition

Suppose $\rho : A \to B$ and $\tau : B \to C$ are relations. The (relational) composition of ρ and τ is defined to be the relation $\sigma = \rho \bullet \tau : A \to C$ where

$$x \; \sigma \; y \iff \exists \, z \in B \, (x \; \rho \; z \wedge z \; \tau \; y).$$

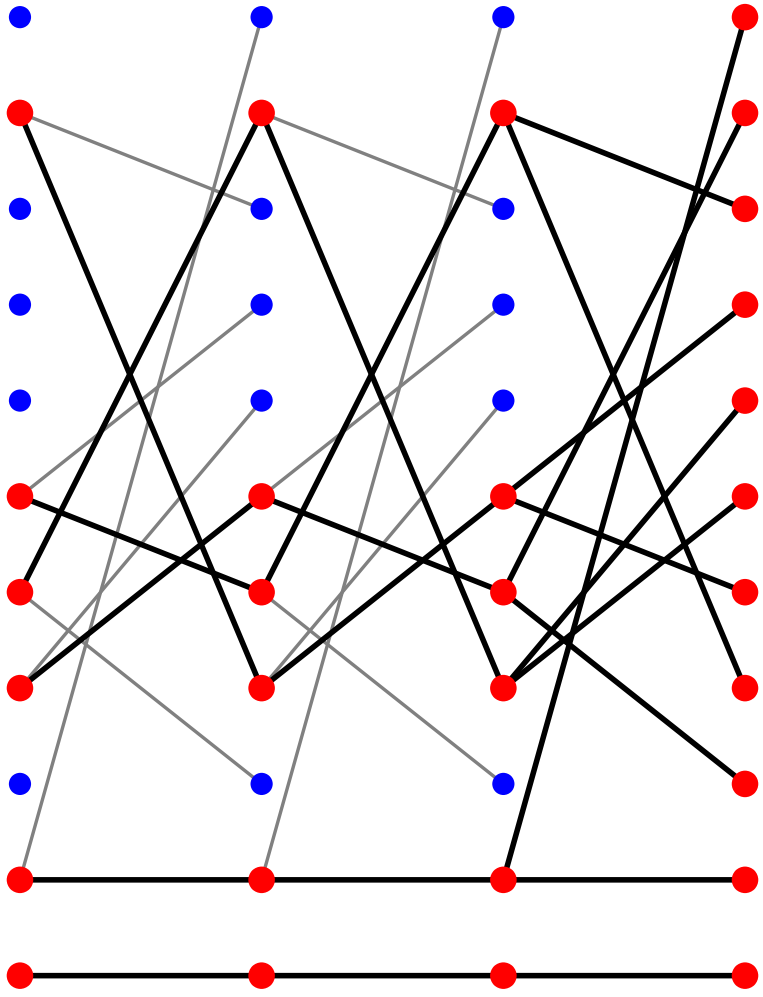
The intermediate element $z \in B$ is a witness (for $x \; \sigma \; y$).

Computationally composition is much more interesting than the previous operations. To test whether $x \; \sigma \; y$ holds we have to conduct a search for the witness z such that $x \; \rho \; z \; \tau \; y$.

Finding good algorithms to compute relational composition is a non-trivial problem; more later.

Visualizing Composition

Composing $x \ \rho \ y \iff x^2 = y \bmod 11$.



Example

$\tau =$ "parent of"

$\tau \bullet \tau =$ "grandparent of"

$\tau^c \bullet \tau =$ "sibling of"

Example

$\tau =$ divisibility on $\mathbb{N}$: "x divides y"

Then $\tau \bullet \tau = \tau$

τ^c is "x is a multiple of y"

Example

$\tau =$ "less than" on $\mathbb{N}$

$\tau^c = >$, and $\tau \cup \tau^c = I_{\mathbb{N}}^-$ (inequality)

$\tau \bullet \tau \subset \tau$.

But for $A = \mathbb{Q}, \mathbb{R}$ we have $\tau \bullet \tau = \tau$.

Lemma

Suppose $\rho : A \to B$, $\sigma : B \to C$, *and* $\tau : C \to D$ *are relations.*

1. $\rho \bullet (\sigma \bullet \tau) = (\rho \bullet \sigma) \bullet \tau$.
2. $\rho \bullet I_B = I_A \bullet \rho = \rho$.
3. $\rho \bullet \emptyset = \emptyset \bullet \rho = \emptyset$.
4. $(\rho \bullet \sigma)^c = \sigma^c \bullet \rho^c$.

Proof. Tedious but rather straightforward application of definitions and a little logic. E.g., the first claim is established like so:

$$\begin{aligned} x\,(\rho \bullet (\sigma \bullet \tau))\,y &\iff \\ \exists\, u\,\big(x\ \rho\ u\,(\sigma \bullet \tau)\,y\big) &\iff \\ \exists\, u, v\,\big(x\ \rho\ u\ \sigma\ v\ \tau\ y\big) &\iff \\ \exists\, v\,\big(x\,(\rho \bullet \sigma)\,v\ \tau\ y\big) &\iff \\ x\,((\rho \bullet \sigma) \bullet \tau)\,y & \end{aligned}$$

□

By the lemma, the set of all binary relations on A, with relational composition as operation, and the identity relation as neutral element, forms a monoid:

$$\langle \mathsf{Rel}_A, \bullet, I_A \rangle$$

Hence one can study the structure of this monoid from an algebraic point of view. A more ambitious project would be to add other operations such as meet and join to the structure.

Exercise

Give a Cayley table for the monoid $\mathsf{Rel}_{\{0,1\}}$.

Exercise

Is this monoid commutative? Is this monoid a group?

Definition

Let ρ be a relation on A. The powers ρ^i, $i \geq 0$, of ρ are defined by

$$\rho^0 = I_A$$
$$\rho^{k+1} = \rho \bullet \rho^k$$

Composition on the left is arbitrary, we could also have chosen composition on the right.

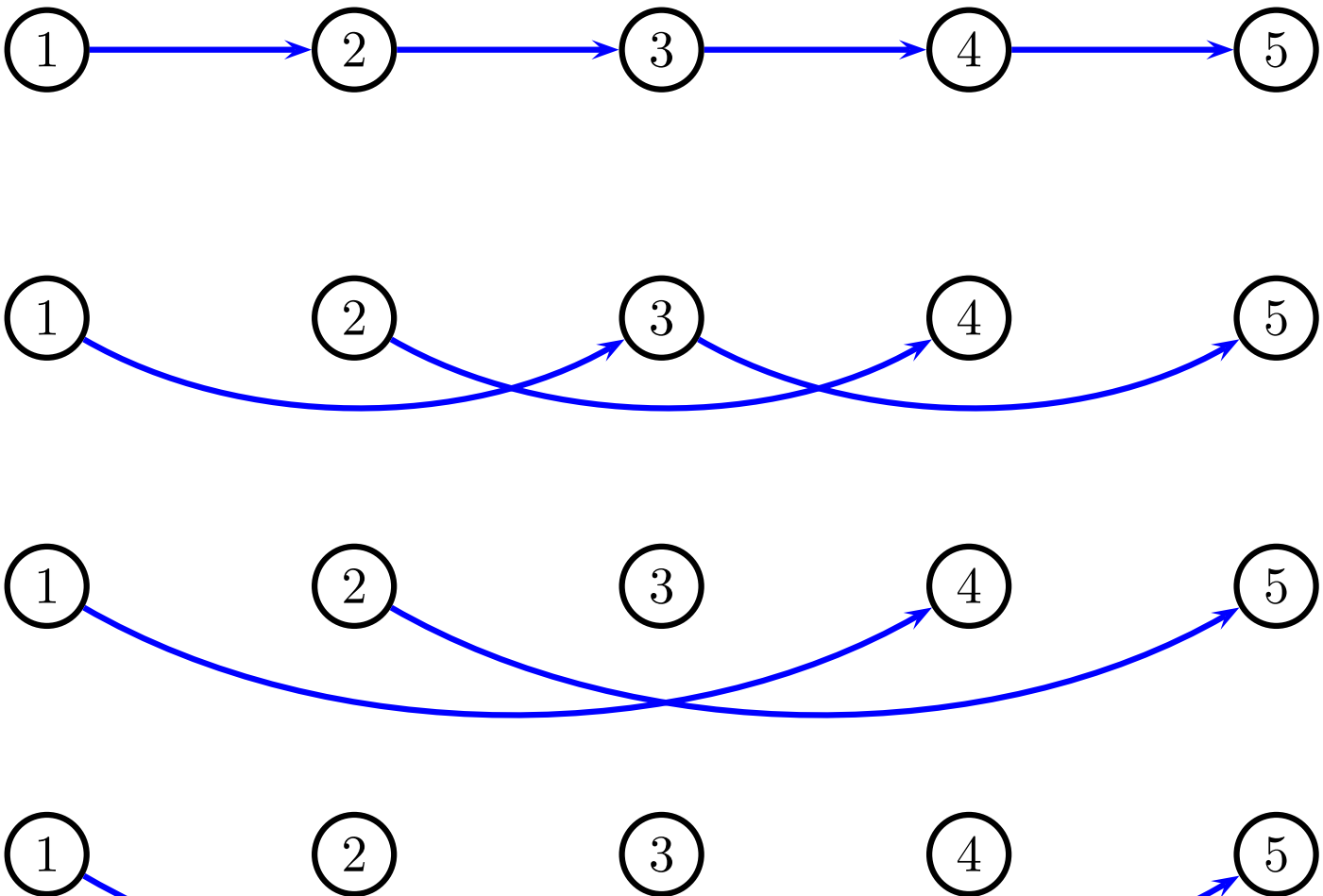
Thus $x\ \rho^k\ y$ if, and only if, there is a ρ-chain of length k from x to y:

$$\exists\, x_0, x_1, \ldots, x_k\ (x = x_0\ \rho\ x_1\ \rho\ x_2\ \rho\ \ldots x_{k-1}\ \rho\ x_k = y)$$

In terms of the monoid or relations this is just exponentiation.

Power Example

The picture below shows the powers of the relation $x \ \rho \ y \iff y = x + 1$ on $[5]$. It is easy to see that in this case, $\rho^k = \emptyset$ for all $k \geq 5$, so only ρ through ρ^4 are shown.



Let ρ be a square relation on A.

Definition

A ρ-chain is a sequence $a_0, a_1, \ldots, a_n$ of elements in A such that $a_i \ \rho \ a_{i+1}$ for all $i = 0, \ldots, n-1$. a_0 is the source of the chain, a_n its target and n its length.

This is often expressed informally in infix notation

$$a_0 \ \rho \ a_1 \ \rho \ \ldots \ \rho \ a_{n-1} \ \rho \ a_n$$

Computationally, the most interesting problem related to chains is to check if there is a chain from some given source to a given target. This is mostly of interest for finite carrier sets where we have a reasonable representation of the relation.

Exercise

What is the longest chain in the divisibility relation (excluding equality) on a natural number n?

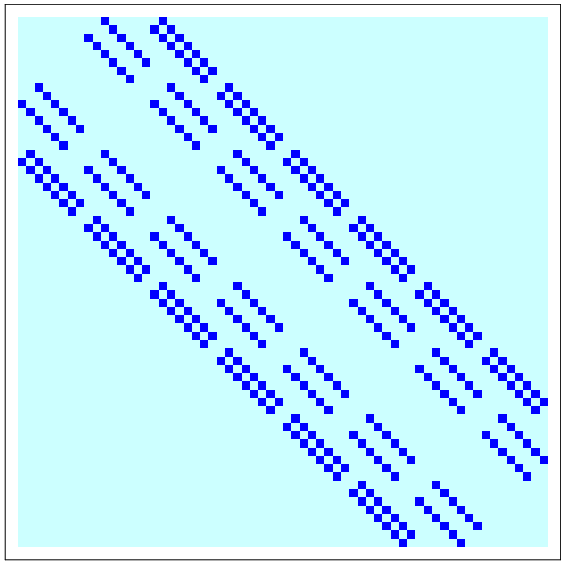
Here is a situation where the Boolean matrix pictures are somewhat interesting.

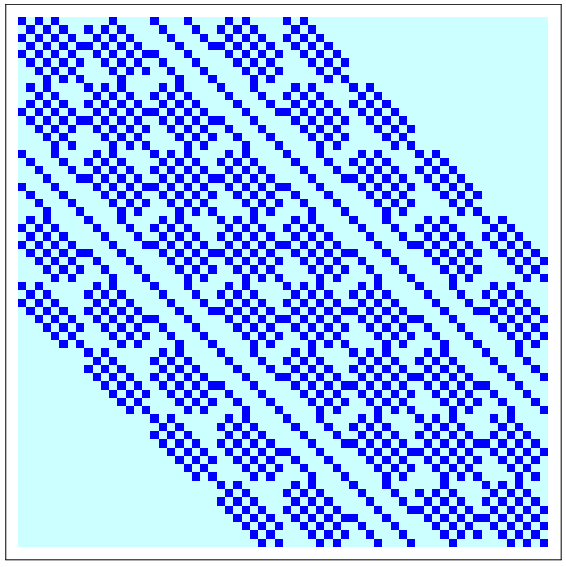
Consider the carrier set $A = [8] \times [8]$ which may be identified with the squares of a chess board. Define two squares to be related by ρ if a knight can move from one to the other.

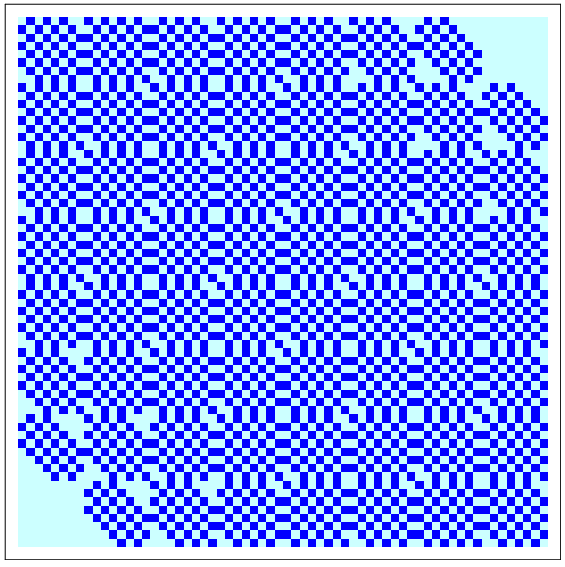
Can a knight starting at square $(1, 1)$ reach all squares on a chessboard?

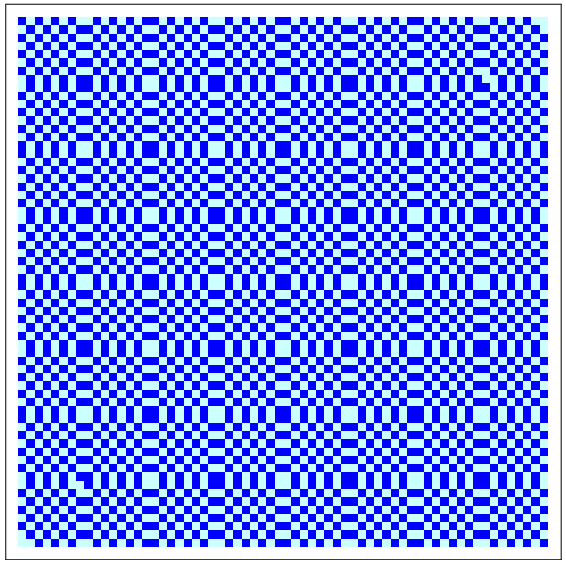
We can flatten the chessboard in row-major order so that the Boolean matrix representation of ρ will be 64 by 64.

The following pictures show reachability by a chain of knight moves, for lengths 1, 2, 3 and 4.









Needless to say, chains are just the relation theoretic version of paths in graph theory.

Thus, if the relation models a communication network (direct links from station to station), a chain/path corresponds to the possibility of sending a message from one station to another (hopping).

There are standard algorithms in graph theory to tackle the chain/path existence problem, most notably depth-first-search and breadth-first-search.

Theorem

The path existence problem in a finite digraph can be solved in time linear in the number of nodes plus the number of edges.

Note that the space requirement is also linear in the size of the vertex set, though.

There is a short list of basic properties of relations that can be combined to define important types of relations such as orders and equivalence relation. Let ρ be a binary relation on A.

Definition

property	$\forall\, x\,, y, z \in A$
reflexive	$x\ \rho\ x$
irreflexive	$\neg(x\ \rho\ x)$
symmetric	$x\ \rho\ y \Rightarrow y\ \rho\ x$
asymmetric	$\neg(x\ \rho\ y \wedge y\ \rho\ x)$
antisymmetric	$x\ \rho\ y \wedge y\ \rho\ x \Rightarrow x = y$
transitive	$x\ \rho\ y \wedge y\ \rho\ z \Rightarrow x\ \rho\ z$

Exercise

Give an example and a counterexample for each property.

The logical definitions from above are standard, but one could also use the calculus of relations to pin down these properties.

property	condition
reflexive	$I \sqsubseteq \rho$
irreflexive	$I \sqcap \rho = \emptyset$
symmetric	$\rho^c \sqsubseteq \rho$
asymmetric	$\rho \sqcap \rho^c = \emptyset$
antisymmetric	$\rho \sqcap \rho^c \sqsubseteq I$
transitive	$\rho \bullet \rho \sqsubseteq \rho$

An interesting fact here is that this characterizations is entirely equational in terms of meet since we can express $\sqsubseteq$ equationally as follows:

$$\rho \sqsubseteq \sigma \iff \rho \sqcap \sigma = \rho.$$

- equal-to, subset-of and divides are reflexive
- less-than, proper-subset-of and parent-of are irreflexive
- equal-to and relatively-prime are symmetric
- less-than and parent-of are asymmetric
- less-than-or-equal, subset-of and divides are antisymmetric
- equal-to, subset-of, divides and ancestor-of are transitive
- parent-of and relatively-prime are not transitive

Note that there are corresponding decision problems: how do we check whether are relation is, say, transitive?

At least for finite carrier sets one would like to have efficient algorithms.

Exercise

Show that $(\rho \bullet \sigma)^c = \sigma^c \bullet \rho^c$.

Exercise

Show that $\rho \bullet \rho^c \bullet \rho \sqsubseteq \rho$ *iff* $\rho \bullet \rho^c \bullet \rho = \rho$.

Exercise

Suppose ρ *is reflexive. Show that* $\rho \bullet \rho^c \bullet \rho \sqsubseteq \rho$ *iff* ρ *is also symmetric and transitive.*

Exercise

Show that $\rho \bullet (\sigma \sqcup \tau) = \rho \bullet \sigma \sqcup \rho \sqcup \tau$.

Exercise

Show that $\rho \bullet (\sigma \sqcap \tau) \sqsubseteq \rho \bullet \sigma \sqcup \rho \sqcup \tau$. *Find an example that demonstrates that equality does not hold in general.*

- Relations
- Operations and Properties

3. Orders

- Equivalence Relations
- Closures

The relations $\leq$ on $\mathbb{N}$, $\leq$ on $\mathbb{R}$ and $\subseteq$ on $\mathfrak{P}(\mathbb{N})$ are similar in a sense, they organize the elements of the domain into "smaller" versus "larger" elements.

What are the crucial properties that make them similar?

Definition

Let ρ be a relation on A.

ρ is a preorder (or quasi-order) if it is both reflexive and transitive.

ρ is a partial order if it is a preorder and antisymmetric.

ρ is an order (or total order or linear order) if it is a partial order and all elements of A are comparable with respect to ρ:

$$\forall\, x, y \in A\, (x\ \rho\ y \vee y\ \rho\ x).$$

Equivalently, ρ is total if $\rho \sqcup \rho^c = U_A$.

Often it is more convenient to consider the strict (i.e. irreflexive) version of an order, obtained by setting

$$x \; \rho' \; y \iff x \neq y \wedge x \; \rho \; y.$$

Notation: $\leq$ versus $<$, $\subseteq$ versus $\subset$. One sometimes refers to the reflexive versions as weak orders.

Thus, a strict preorder is any irreflexive and transitive relation. Note that any such relation is automatically asymmetric.

A strict total order has the additional trichotomy property:
$\forall\, x, y \in A\, (x \; \rho \; y \vee x = y \vee y \; \rho \; x)$.

Also, the converse of an order is often useful: we write $\geq$ and $>$ versus $\leq$ and $<$.

Exercise

Show that the converse of a (strict) pre/partial/total order is again a (strict) pre/partial/total order.

The operation $\rho' = \rho - I$ turns a partial order ρ into a strict partial order ρ'.

But if ρ is just a preorder, ρ' need not be transitive.

Exercise

Prove these claims.

Another piece of terminology: a structure

$$\langle A, < \rangle$$

consisting of a set A and a partial order $<$ (or a weak version) is often called a poset.

This is mostly an acknowledgment that partial orders are so important that one should have a nice, compact, generally accepted name for them (just like groups or fields).

The study of posets has turned out to be of major importance for the semantics of programming languages.

Examples

- The usual order relations on numbers are total orders.
- Subset-of and divides are partial orders.
- Ordering polynomials by their degree produces a preorder.
- Cardinality of a set is a preorder (and has full comparability).
- Lies-north-east-of in the plane is a partial order.
- Lexicographic order on words is a total order.
- The substring order is a partial order on strings:
  $x \preceq y \iff \exists\, u, v\, (y = uxv)$.
- The element relation $\in$ is a total order on the class of ordinals but only a partial order on the class of all sets.

A standard problem is to lift a given order $\leq$ on A to $A \times A$. One plausible definition is

Definition

The product order of $\leq$ with itself is defined by

$$(x, y) \leq_2 (x', y') \iff x \leq x' \wedge y \leq y'.$$

Example

On the plane $\mathbb{R} \times \mathbb{R}$ this is the "north-east-of" relation.

Thus, the product order is partial, even though $\langle \mathbb{R}; \leq \rangle$ is a total order. Can we manufacture a total order?

How about

$$(x, y) \trianglelefteq (x', y') \iff x \leq x' \vee (x = x' \wedge y \leq y').$$

Exercise

Show that $\trianglelefteq$ is indeed a total order.

It is slightly harder to order all sequences over A, not just pairs.

The usual order on characters $a < b < c < \ldots < y < z$ can be lifted to words over this alphabet.

Recall that we write $|u|$ for the length of a sequence u.

Definition

Consider two sequences u and v over A.

1. Lexicographic Order (Dictionary Order)
   $u \prec v$ lexicographically if u is a proper prefix of v or if $u = xay$, $v = xbz$ and $a < b$, where x, y, z are sequences, $a, b \in A$.
2. Length-Lex Order
   $u \prec v$ in the length-lex order if $|u| < |v|$ or $|u| = |v|$ and u precedes v lexicographically.
3. Length Order
   $u \prec v$ in the length order if $|u| < |v|$.

Lexicographic order and length-lex order are bonified total orders on finite sequences, but length order is only a preorder.

On words over the alphabet $\{0, 1\}$, length-lex order produces

$$\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \ldots$$

This is crucially important for induction arguments: we finish off all words of length k before working on length $k + 1$.

But lexicographic order for these few words would be

$$\varepsilon, 0, 00, 000, 001, 010, 011, 1, 11$$

This is much less natural if one tries to systematically work through these words.

Lexicographic order is used in the C++ library STL to overload `operator<` for stacks.

Talking about induction, here is the key requirement for inductive arguments and constructions.

Definition

A binary relation ρ on A is well-founded if every non-empty subset of A contains a ρ-minimal element. If ρ is also a strict total order, it is said to be well-ordered.

So for a well-founded relation $\langle A, < \rangle$ we have

$$\emptyset \neq X \subseteq A \Rightarrow \exists\, a \in X\, \forall\, z \in A\, (z < a \rightarrow z \notin X).$$

Equivalently (using the axiom of choice), there are no infinite descending chains in $<$:

$$a_0 > a_1 > a_2 > \ldots > a_n > \ldots$$

Definition

Let $\langle A, < \rangle$ be a well-founded relation. A subset $X \subseteq A$ is inductive if $\forall\, x < y\, (x \in X) \to y \in X$.

The fundamental result that enables us to perform induction is the following.

Theorem

Let $\langle A, < \rangle$ be well-founded and $X \subseteq A$ inductive. Then $X = A$.

So we can perform induction on the natural numbers with the standard order, on words in length-lex order, on ordinals with the standard order, on inductively defined data structures such as trees and lists, and so on.

Exercise

Show that we can also perform induction in a suitable partial order. In fact, not even transitivity is needed, just the well-ordering property from above.

The classic example of a well-ordering is $\langle \mathbb{N}, < \rangle$, the standard order on the natural numbers.

But other number systems such $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ all fail to be well-founded. Do not confuse this with the Well-Ordering Principle which says that every set can be well-ordered: they can be, but the standard order does not do the job.

In fact, when Zermelo first proposed his well-ordering principle in 1904 he met with considerable resistance: any well-order of the reals seemed artificial and contrived.

For computer science, recursive data-types produce important examples of well-founded relations. Many operations on data structures such as lists, words and trees can be explained in terms of induction along these well-founded relations.

Another important application is termination: if the steps in a computation can be described in terms of a well-founded relation, the computation must be finite. This is the idea behind the totality proof for Ackermann's function.

Lemma

The direct (pointwise) product of two well-founded relation is well-founded. The lexicographic product of two well-founded relation is well-founded.

Exercise

Prove the lemma. How about well-founded partial orders and well-orderings?

Suppose A is some set equipped with an order $<$.

Informally, a multiset over A is a set that allows multiple occurrences of an element, as in $\{a, a, a, b, b\}$. Technically, a multiset can be interpreted as a map $X : A \to \mathbb{N}$ where $X(a)$ indicates the number of occurrences of element a. A finite multiset is a set with finite support.

Write $\mathcal{M}(A)$ for the collection of all finite multisets over A.

We can define a partial order $X \prec Y$ on $\mathcal{M}(A)$ as follows: remove some elements a from X, and replace them by elements $b < a$. Thus

$$\emptyset \prec \{1, 1, 1, 2, 2, 2\} \prec \{1, 1, 3\} \prec \{1, 4\}$$

More precisely,

$$Y \prec X \iff \exists\, U, V\, (U \neq \emptyset \wedge Y = (X - U) \cup V \wedge \forall\, v \in V\, \exists\, u \in U\, (v < u))$$

A little experimentation with $\mathcal{M}(\mathbb{N})$ shows that $\prec$ appears to be well-founded: while the number of elements potentially increases in the step $Y \prec X$, these elements are smaller, so ultimately the well-ordering on $\mathbb{N}$ wins out.

Theorem

Let $<$ be an order on A and $\prec$ the corresponding order on $\mathcal{M}(A)$. Then $<$ is well-founded if, and only if, $\prec$ is well-founded.

The proof requires König's lemma on finitely branching infinite trees.

- Relations
- Operations and Properties
- Orders

4. Equivalence Relations

- Closures

Equality is one of the most important relations. Clearly, equality is reflexive, symmetric and transitive:

- $x = x$,
- $x = y$ implies $y = x$, and
- $x = y$, $y = z$ implies $x = z$.

It is natural to consider weaker versions of equality: coarser relations that maintain these three properties.

These relations formalize the notion of two objects being "the same" in some sense – without necessarily being identical.

Definition

A square relation ρ on A is an equivalence relation if ρ is symmetric, reflexive, and transitive.

Suppose ρ equivalence relation. The equivalence class of $a \in A$ is:

$$[a]_\rho = \{\, x \in A \mid a \;\rho\; x \,\}.$$

$A/\rho = \{\, [a]_\rho \mid a \in A \,\}$ is the collection of all such classes, the quotient of A by ρ. The cardinality of A/ρ is the index of ρ.

Recall that for a binary relation ρ on some set A we have

property	$\forall\, x\,, y, z \in A$
reflexive	$x \;\rho\; x$
symmetric	$x \;\rho\; y \rightarrow y \;\rho\; x$
transitive	$x \;\rho\; y \wedge y \;\rho\; z \rightarrow x \;\rho\; z$

Examples

- equality is an equivalence relation
- same-weight is an equivalence relation on dumbbells
- same-parents is an equivalence relation on humans
- same-cardinality is an equivalence relation on sets
- same-area is an equivalence relation on polygons
- same-input-output is an equivalence relation on programs

Note that these are all of the form

$$x \; \rho \; y \iff f(x) = f(y)$$

for some suitable function f.

Congruence modulo m on the integers.

$$x = y \pmod{m} \iff m \text{ divides } x - y$$

Is an equivalence relation on $\mathbb{Z}$ where

$$[a] = \ldots, a - 2m, a - m, a, a + m, a + 2m, \ldots$$

Note that the equivalence relation $\pmod{m}$ has index m.

This simple equivalence is the foundation for a lot of cryptographic schemes, including RSA.

Was first studied in great detail by C. F. Gauss (1777–1855) in his work on number theory.

What can we say about equivalence classes?

The equivalence classes of I_A and U_A are trivial:

$$\{a\} = [a]_= \quad \text{and} \quad A = [a]_{U_A}$$

In general, we always have

$$a \in [a]_\rho$$

by reflexivity.

Usually write $[a]$ instead of $[a]_\rho$ unless there is any danger of confusion.

The following dichotomy is the single-most important property of equivalence classes.

Lemma

Let ρ be an equivalence relation on A. For all $a, b \in A$:

$$[a] = [b] \quad \textit{or} \quad [a] \cap [b] = \emptyset.$$

Proof.

If $c \in [a] \cap [b]$, then by symmetry for any $z \in [a]$:

$$z \ \rho \ a \ \rho \ c \ \rho \ b$$

Hence $z \in [b]$ by transitivity. But then $[a] \subseteq [b]$.

By a symmetric argument, $[b] \subseteq [a]$, done. □

Definition

A partition of a set A is a set $P \subseteq \mathfrak{P}(A)$ of subsets of A such that

- $X \neq \emptyset$ for all $X \in P$,
- $\bigcup P = A$, and
- $X \neq Y \in P \rightarrow X \cap Y = \emptyset$.

The sets $X \in P$ are called the blocks of the partition.

This is really just different terminology, not a new idea: a block is nothing but an equivalence class.

By the lemma, the classes form a partition.

Lemma

Equivalence relations correspond exactly to partitions.

Proof.

If ρ is an equivalence relation on A the equivalence classes $[a]$ of ρ produce a partition by the last lemma.

Now let P be a partition of A. Define

$$x \, \rho \, y \iff \exists \, X \in P \, (x, y \in X).$$

Clearly ρ is reflexive and symmetric.

Transitivity follows from the lemma. □

??????????

Consider the chess board $C = [8] \times [8]$.

Define a relation ρ on C by: $x \; \rho \; y$ iff a knight can move from x to y (sequence of single moves).

Claim

ρ is an equivalence relation.

There is exactly one class: $[(1,1)]_\rho = C$. Same for a rook, king, queen.

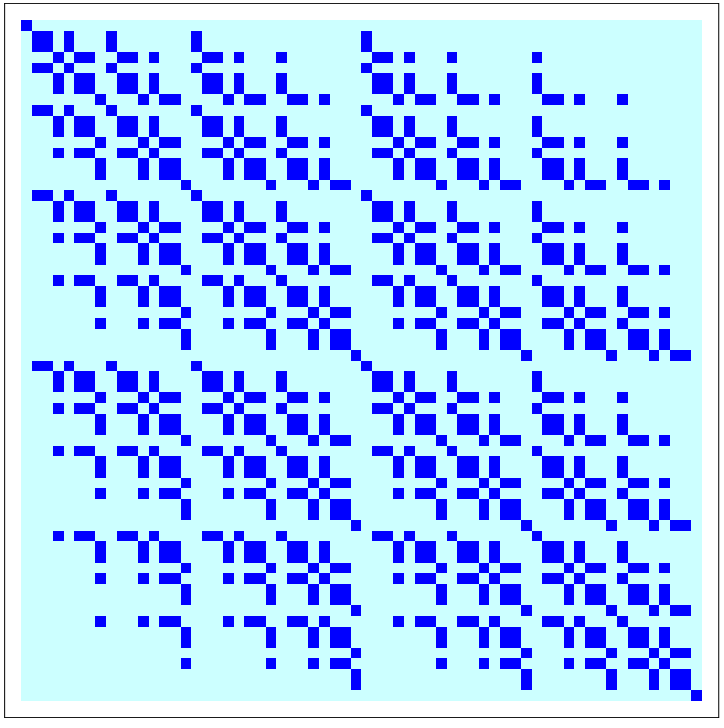
But for a bishop there are two equivalence classes: the black squares and the white squares.

Exercise

How many equivalence classes for a pawn? Only consider the basic move $(i,j) \to (i,j+1)$, no conversion. What's wrong?

Consider as carrier set the collection of all binary lists of length 6 (organized in natural order). Here is a picture of the relation "list x has the same number of 1's as list y.

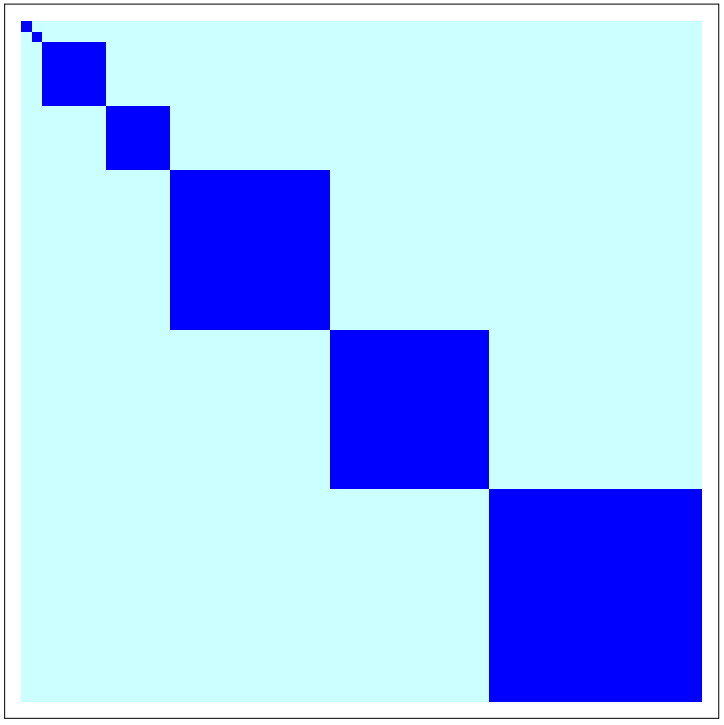
Note that by definition this is an equivalence relation.



The picture clearly shows reflexivity and symmetry. But how about transitivity?

Transitivity is just about impossible to see from the last picture.

But, if we reorder the carrier set so that equivalent elements are all consecutive, the picture becomes very clear.



Exercise

What are the sizes of these blocks?

Picture 1 uses the carrier set in natural order, but in picture 2 the carrier set is reordered so all elements of a block in the equivalence relation are contiguous.

natural	reordered
000000	000000
000001	111111
000010	000001
000011	000010
000100	000100
000101	001000
...	...
111101	110010
111110	110100
111111	111000

The important point here is that the relation is unchanged, only its representation in terms of the Boolean matrix has changed.

Example: Equidecomposability

Here is a more interesting equivalence relation.

Definition

Two polygons P and Q are equidecomposable if P can be cut up into finitely many triangles which can be reassembled to form Q.

We will not give a precise definition of what we mean by reassemble and appeal to common (geometric) sense.

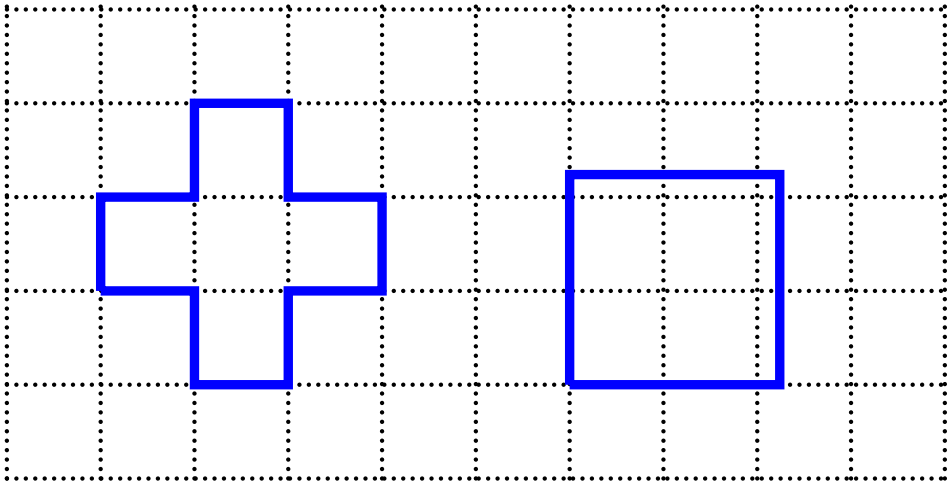
Lemma

Equidecomposability is an equivalence relation.

Proof. Reflexivity and symmetry is obvious. But transitivity is not: need the fact that the intersection of two polygons is another polygon (or a set of polygons) and that all polygons can be triangulated. □

A Picture

Given two triangulations of the same polygon we can construct a finer one: just triangulate all the polygonal regions produced by the intersection of the two given triangulations.



The square on the right is $\sqrt{5}$ by $\sqrt{5}$, so both polygons have area 5.

Exercise

Show that the two polygons are equidecomposable. Use as few cuts as possible.

It is clear that any two equidecomposable polygons must have the same area: cutting them up and reassembling the pieces does not change the area.

But it is somewhat surprising that the opposite direction also holds.

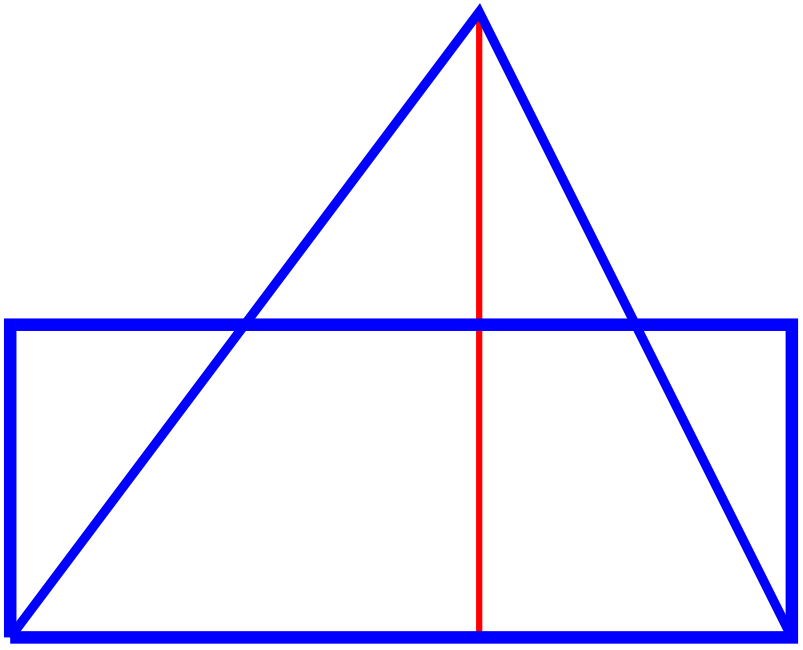
Theorem (Bolyai-Gerwin Theorem)

Let P and Q be two polygons of equal area. Then P and Q are equidecomposable.

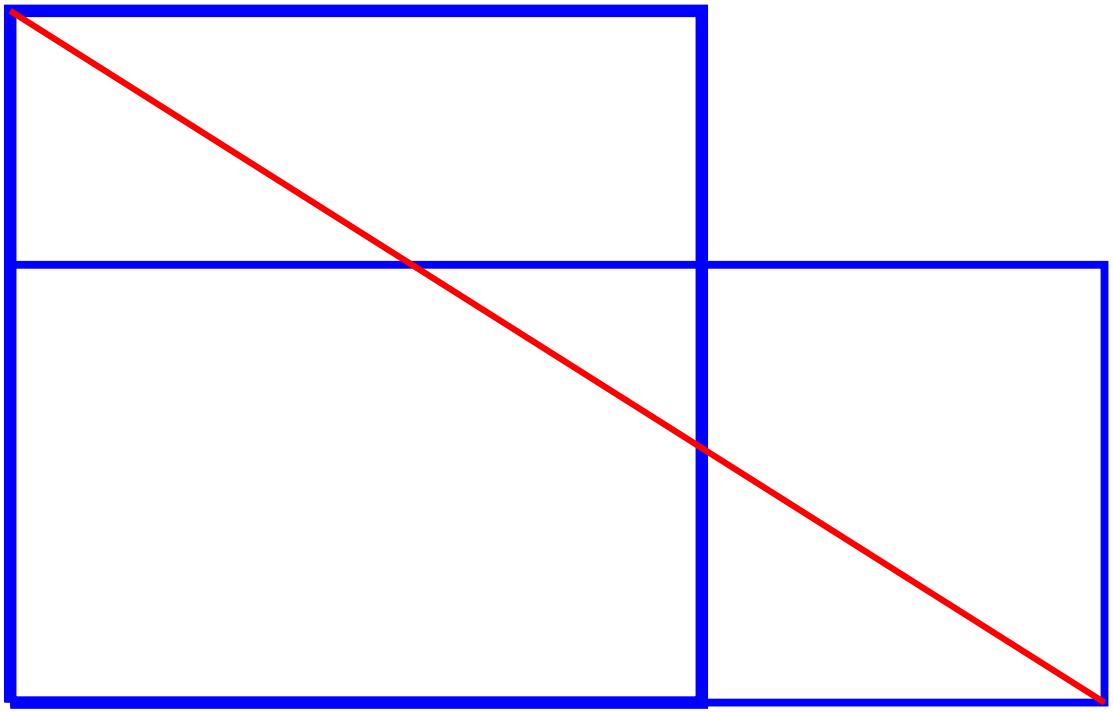
We sketch the proof below. It is an excellent exercise to turn this into a real proof.

To prove the Bolyai-Gerwin theorem it suffices to show that every polygon P is equidecomposable with a square: there is exactly one square for each area. Since we can triangulate any polynomial, we start with triangles.

Here is a method to convert triangles to rectangles of equal area:

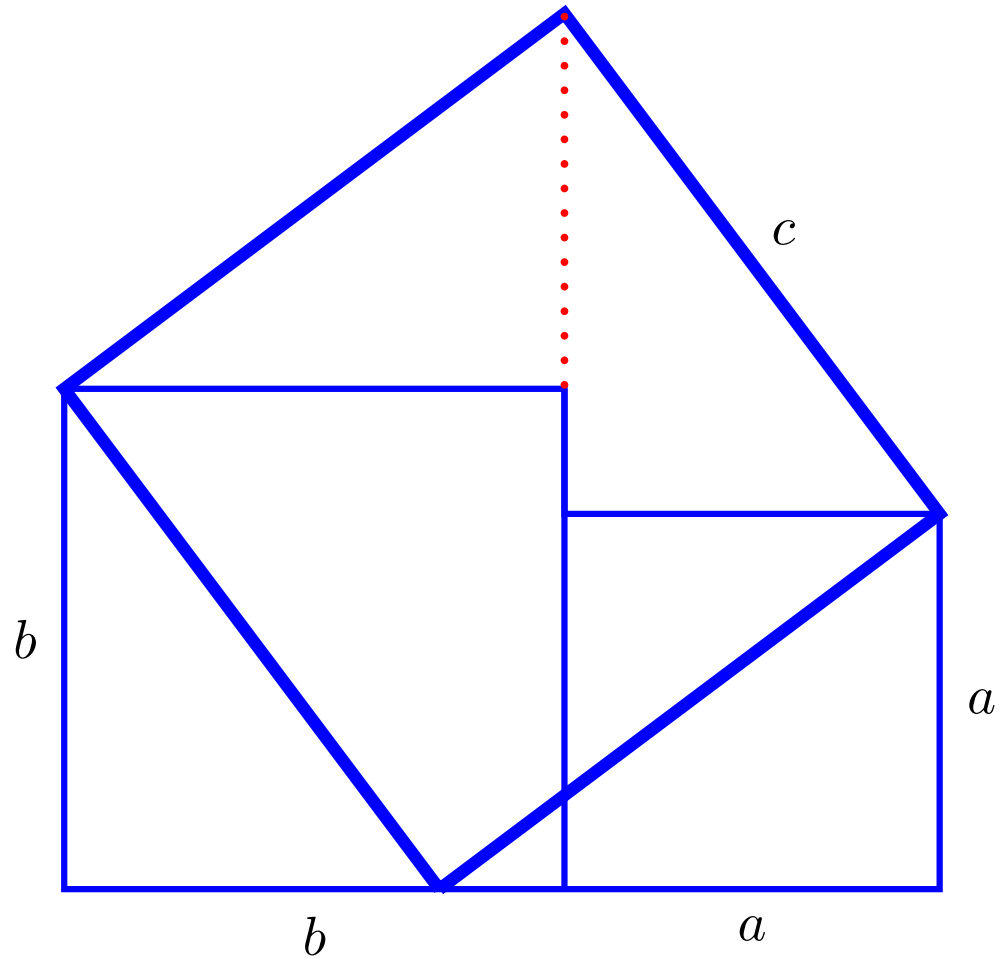


In the second step we convert rectangles to squares.



This requires that the longer side of the rectangles is ≤ 4 times the shorter one. What if not?

Lastly, we have to add squares and produce new squares. We exploit Pythagoras's theorem: $a^2 + b^2 = c^2$.



Sorting algorithms in the STL can use orders supplied by user: provide a comparison operation $cmp(x, y)$. Must be a "strict weak ordering."

```
sort( A.begin(), A.end(), cmp );
```

[1] $cmp(x, x)$ is false.

[2] If $cmp(x, y)$ and $cmp(y, z)$, then $cmp(x, z)$.

[3] Define $equiv(x, y)$ to be $!(cmp(x, y)||cmp(y, x))$. If $equiv(x, y)$ and $equiv(y, z)$, then $equiv(x, z)$.

What on earth does this really mean?

Let's write ρ for the relation cmp, and τ for $equiv$.

Note that ρ is transitive and irreflexive by [1] and [2].

Claim

τ is an equivalence relation.

First off, $\tau = (\rho \cup \rho^c)^-$. Transitivity of τ is given.

Reflexivity follows from the fact that both ρ and ρ^c are irreflexive.

Symmetry follows from $\rho \cup \rho^c$ being symmetric, and symmetry being preserved by complementation. □

So, the specification really says: incomparable elements have to be equivalent, we don't care which order they appear in.

Otherwise the sorting algorithm will not work.

- Sorting complex numbers according to their absolute value is fine.
- Sorting lists according to their length is fine.
- Sorting subsets according to cardinality will work.
- Sorting an array of subsets according to $x \subseteq y$ will not work.

- Relations
- Operations and Properties
- Orders
- Equivalence Relations

5. Closures

A frequent computational problem is to enlarge a given a relation ρ on A a bit until it has some property P.

Definition

Let ρ be a relation on A and P a property of relations. The P-closure of ρ is the $\sqsubseteq$-least relation σ such that $\rho \sqsubseteq \sigma$ and σ has property P.

Important cases are: reflexive closure, symmetric closure, transitive closure; combinations thereof.

Notation:
The transitive reflexive closure is often written $\rho^\star$. The transitive closure is written ρ^+.

How do we know that the, say, transitive closure exists? We have to argue that a relation with certain properties actually exists.

Claim

If $\rho \sqsubseteq \sigma_i$ and σ_i is transitive, $i = 1, 2$, then $\sigma_1 \sqcap \sigma_2$ is again transitive.

This claim is true even for an infinite family $(\sigma_i)_{i \in I}$.

But then, abstractly, the closure is

$$\sigma = \bigcap \{\, \tau \sqsubseteq A \times A \mid \rho \sqsubseteq \tau \wedge \tau \text{ has property } P \,\}$$

The set on the right can never be empty: U_A is in there.

Of course, this is useless for computational purposes: we need algorithms, not general abstract nonsense.

A fine point: set on the right contains the relation whose existence we worry about (impredicative definition).

In the same way we can define reflexive closure, symmetric closure, transitive closure and reflexive transitive closure.

Notation:

$$\mathrm{eqcl}(\rho), \mathrm{rcl}(\rho), \mathrm{scl}(\rho), \mathrm{tcl}(\rho), \mathrm{rtcl}(\rho)$$

Lemma

All these operations F are idempotent: $F(F(\rho)) = F(\rho)$.

Lemma

- $\mathrm{scl}(\mathrm{rcl}(\rho)) = \mathrm{rcl}(\mathrm{scl}(\rho))$
- $\mathrm{rtcl}(\rho) = \mathrm{tcl}(\mathrm{rcl}(\rho)) = \mathrm{rcl}(\mathrm{tcl}(\rho))$
- $\mathrm{eqcl}(\rho) = \mathrm{tcl}(\mathrm{scl}(\mathrm{rcl}(\rho)))$

One can think of ρ as a succinct representation of σ: we have ρ, and we want to compute σ. Sometimes, that's easy.

Write $\mathrm{rcl}(\rho)$ and $\mathrm{scl}(\rho)$ for reflexive and symmetric closure, and $\mathrm{rscl}(\rho)$ for the reflexive symmetric closure.

$$\mathrm{rcl}(\rho) = \rho \sqcup I_A$$
$$\mathrm{scl}(\rho) = \rho \sqcup \rho^c$$
$$\mathrm{rscl}(\rho) = \rho \sqcup \rho^c \sqcup I_A$$

Example

On the natural numbers we have: $\mathrm{rcl}(<) = \leq$, $\mathrm{scl}(<) = \neq = U_A - I_A$, $\mathrm{rscl}(<) = U_{\mathbb{N}}$

Exercise

Show that $\mathrm{rscl}(\rho) = \mathrm{rcl}(\mathrm{scl}(\rho)) = \mathrm{scl}(\mathrm{rcl}(\rho))$. *Show that* $\mathrm{rcl}(\mathrm{rcl}(\rho)) = \mathrm{rcl}(\rho)$.

Here is another example: equivalence as closure.

Suppose you need to implement equivalence relations on $[n]$ where n is, say, about 1000. There could be as many as 10^6 pairs that are related. Clearly, we do not wish to store so much information.

What do we really need to know?

The property "is an equivalence relation" is yet another closure property. Get another closure operation: $ec(\rho)$:

$$ec(\rho) = tc(rsc(\rho)) = (I \sqcup \rho \sqcup \rho^c)^+$$

But the size of ρ may well be $O(n)$, while the size of $ec(\rho)$ is $O(n^2)$.

Powers and transitive closure are very closely connected. We could have defined the closures like so:

$$\mathrm{rtcl}(\rho) = \bigsqcup_{k \geq 0} \rho^k = I_A \sqcup \rho \sqcup \rho^2 \sqcup \rho^3 \sqcup \ldots$$

$$\mathrm{tcl}(\rho) = \bigsqcup_{k > 0} \rho^k = \rho \sqcup \rho^2 \sqcup \rho^3 \sqcup \ldots$$

Of course, this definition is a bit intimidating since it uses an infinitary operator.

But note that unlike with infinite series in calculus there is no problem with convergence here: no matter what the individual terms are, the big join will always exist.

This characterization simply says that $x \ \mathrm{tcl}(\rho) \ y$ iff

$$\exists\, k \geq 0 \, \exists\, x_0, x_1, \ldots, x_k \ (x = x_0 \ \rho \ x_1 \ \rho \ \ldots \ \rho \ x_{k-1} \ \rho \ x_k = y)$$

Note that length of the chains is actually bounded since A is finite.

Exercise

One can always assume $k \leq n - 1$ *where* $n = |A|$.

We will exploit this fact in the section on algorithms for relations.

Aside: k-ary Relations

We have limited our discussion to binary relations, but of course we can also study relations on k objects where $k \geq 1$. We can define relations in general to be subsets

$$\rho \subseteq A_1 \times A_2 \times \ldots \times A_k$$

of general Cartesian products.

The case $k = 1$ is somewhat uninteresting, since unary relations are essentially the same as sets. However, k-ary relations for $k > 2$ do appear, in particular when dealing with databases.

For example, we might be interested in a relation on the product of

- A_1: students
- A_2: departments
- A_3: degree programs
- A_4: courses
- A_5: grades
- A_6: status indicator

We won't pursue this matter here, google for relational data bases.