

Relations

Binary Relations

A **binary relation** between arbitrary sets A and B is a subset of the binary product $A \times B$.

If $R \subseteq A \times A$, we often say that R is a binary relation **on** A .

Notation

loves(John, Mary) or ‘John loves Mary’ rather than
(John, Mary) \in loves

$2 < 5$

`a `f` b` in Haskell

Example

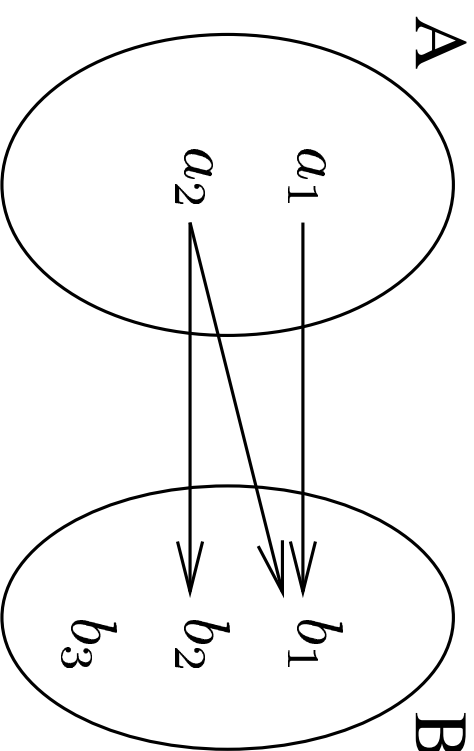
For $A = \{a, b\}$, there are sixteen binary relations on A :

$$\begin{array}{ll} \emptyset & \{(a, b), (b, a)\} \\ \{(a, a)\} & \{(a, b), (b, b)\} \\ \{(a, b)\} & \{(b, a), (b, b)\} \\ \{(b, a)\} & \{(a, a), (a, b), (b, a)\} \\ \{(b, b)\} & \{(a, a), (a, b), (b, b)\} \\ \{(a, a), (a, b)\} & \{(a, a), (b, a), (b, b)\} \\ \{(a, a), (b, a)\} & \{(a, b), (b, a), (b, b)\} \\ \{(a, a), (b, b)\} & \{(a, a), (a, b), (b, a), (b, b)\} \end{array}$$

Diagram

Let $A = \{a_1, a_2\}$, $B = \{(b_1, b_2, b_3)\}$ and $R = \{(a_1, b_1), (a_2, b_1), (a_2, b_2)\}$.

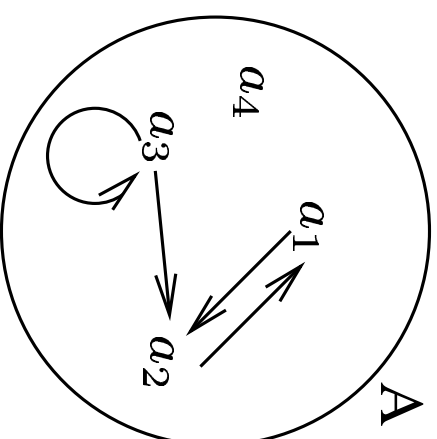
Relation R is illustrated by the diagram



This representation will be used in this course.

Directed Graph

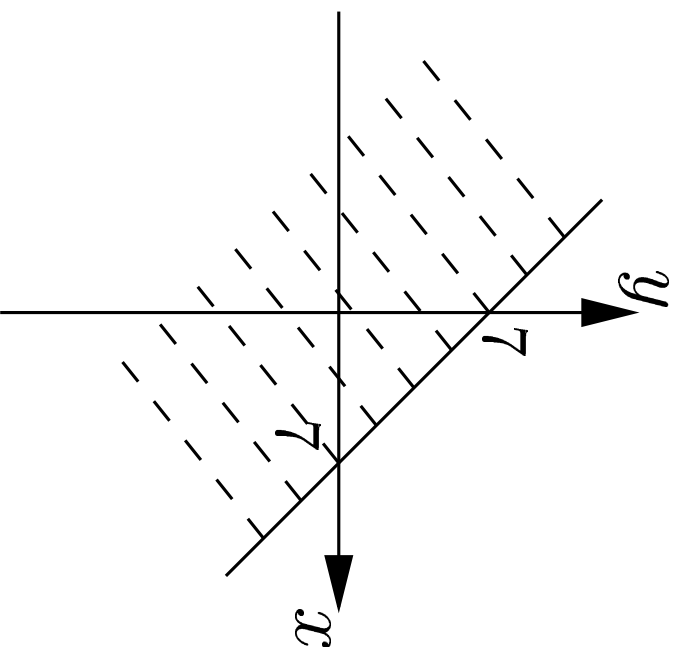
Let R be a binary relation on $A = \{a_1, a_2, a_3, a_4\}$ with $R = \{(a_1, a_2), (a_2, a_1), (a_3, a_2), (a_3, a_3)\}$. The directed graph of this relation is



Notice that the direction of the arrows matters. This representation will be used in Discrete maths 2.

Special representation

The relation R defined by $x R y$ if and only if $x + y \leq 7$ can be represented by



Matrix Representation

Let $A = \{a_1, a_2\}$, $B = \{b_1, b_2, b_3\}$ and $R = \{(a_1, b_1), (a_2, b_2)\}$ as before. The matrix representation of R is

$$\begin{pmatrix} T & F & F \\ T & T & F \end{pmatrix}$$

where T stands for True and F for False.

This representation can be generalised to arbitrary finite sets A and B , as discussed in the notes.

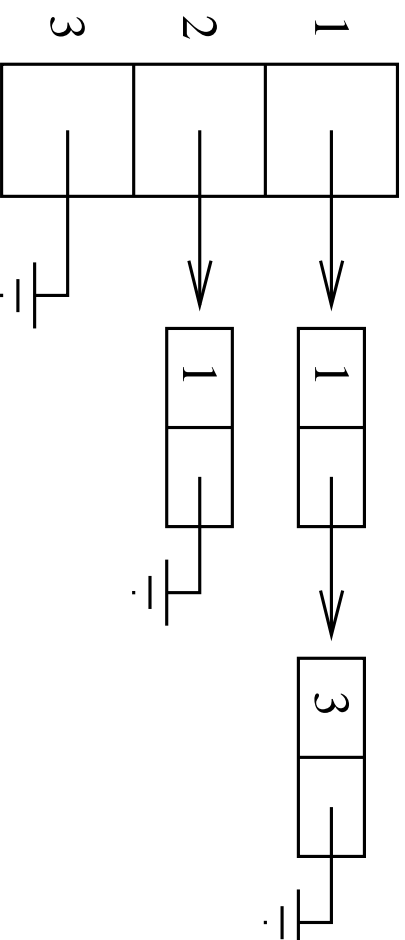
You do not need to remember this representation.

Adjacency List

We can store a relation using an array.

With a **sparse** relation, we can instead use an **adjacency list**.

Consider the binary relation $R = \{(1, 1), (1, 3), (2, 1)\}$ on set $\{1, 2, 3\}$. The adjacency list representation is



You do not need to remember this representation.

***n*-ary relation**

A **n-ary relation** between sets A_1, \dots, A_n is a subset of a n -ary product $A_1 \times \dots \times A_n$.

The definition of a 2-ary relation is the same as that of a binary relation.

A **unary relation**, or **predicate**, over set A is a 1-ary relation: that is, a subset of A .

Examples

1. The set $\{x \in \mathcal{N} : x \text{ is prime}\}$ is a unary relation on \mathcal{N} .
2. The set $\{(x, y, z) \in \mathcal{R}^3 : \sqrt{x^2 + y^2 + z^2} = 1\}$ is a 3-ary relation on the real numbers, which describes the surface of the unary sphere with centre $(0, 0, 0)$.

Basic Relation Operators

Let $R, S \subseteq A_1 \times A_2$. Define the relations $R \cup S$, $R \cap S$ and \bar{R} , all with type $A_1 \times A_2$, by

Relation Union

$(a_1, a_2) \in R \cup S$ iff $(a_1, a_2) \in R$ or $(a_1, a_2) \in S$;

Relation Intersection

$(a_1, a_2) \in R \cap S$ if and only if $(a_1, a_2) \in R$ and $(a_1, a_2) \in S$;

Relation Complement

$(a_1, a_2) \in \bar{R}$ iff $(a_1, a_2) \in A_1 \times A_2$ and $(a_1, a_2) \notin R$.

Contrast the difference between the relation and set operators.

Example

Let R and S be binary relations on $\{1, 2, 3, 4\}$ such that

$$R = \{(1, 2), (2, 3), (3, 4), (4, 1)\}$$

$$S = \{(1, 2), (2, 1), (3, 4), (4, 3)\}$$

$$R \cup S = \{(1, 2), (2, 3), (3, 4), (4, 1), (2, 1), (4, 3)\}$$

$$R \cap S = \{(1, 2), (3, 4)\}$$

$$\overline{R} = \{(1, 1), (1, 3), (1, 4), (2, 1), (2, 2), (2, 4), (3, 1), (3, 2), (3, 3), (4, 2), (4, 3), (4, 4)\}$$

More Operators

Identity Relation

Given any set S , the **identity** on A , written id_A , is a binary relation on A defined by $\text{id}_A = \{(a, a) : a \in A\}$.

Inverse relation

Let $R \in A \times B$ denote an arbitrary binary relation. The **inverse** of R , written R^{-1} , is defined by $a R^{-1} b$ if and only if $b R a$.

Example

Let R be a binary relation on $\{1, 2, 3, 4\}$ such that

$$R = \{(1, 2), (2, 3), (3, 4), (4, 1)\}.$$

$$R^{-1} = \{(2, 1), (3, 2), (4, 3), (1, 4)\}.$$

$$(R^{-1})^{-1} = R.$$

Notice the difference between R^{-1} and \bar{R} given earlier.

Composition of Relations

Given $R \subseteq A \times B$ and $S \subseteq B \times C$, then the **composition** of R with S , written $R \circ S$, is defined (for arbitrary a and c) by

$$a R \circ S c \text{ if and only if } \exists b \in B. (a R b \wedge b S c)$$

The notation $R \circ S$ may be read as ‘ R composed with S ’ or ‘ R circle S ’.

The relation $R \circ S$ is only defined if the types of R and S match.

Contrast with the Haskell notation for functional composition:

$$(g \cdot f) \ x = g \ (f \ x) \quad \text{Very confusing!}$$

Example

Let R and S be binary relations on $\{1, 2, 3, 4\}$ such that

$$R = \{(1, 2), (2, 3), (3, 4), (4, 1)\}$$

$$S = \{(1, 2), (2, 1), (3, 4), (4, 3)\}$$

$$R \circ S = \{(1, 1), (2, 4), (3, 3), (4, 2)\}$$

$$S \circ R = \{(1, 3), (2, 2), (3, 1), (4, 4)\}$$

Equalities between Relations

1. If $R \subseteq A \times B$, then $\text{id}_A \circ R = R = R \circ \text{id}_B$.
2. \circ is associative: that is, for arbitrary relations $R \subseteq A \times B$ and $S \subseteq B \times C$ and $T \subseteq C \times D$, then

$$R \circ (S \circ T) = (R \circ S) \circ T.$$

The proof of part 1 is left to you.

Notation

$x R y$ $S z$ is shorthand for $x R y$ and $y S z$.

Proof of part 2

Let (x, u) be an arbitrary member of $(R \circ S) \circ T$.

$$\begin{aligned}x (R \circ S) \circ T u & \text{ iff } \exists z. x (R \circ S) z T u \\ & \text{ iff } \exists z. ((\exists y. x R y S z) \wedge z T u) \\ & \text{ iff } \exists z, y. (x R y S z \wedge z T u) \\ & \text{ iff } \exists z, y. (x R y \wedge y S z T u) \\ & \text{ iff } \exists y. (x R y \wedge \exists z. y S z T u) \\ & \text{ iff } \exists y. (x R y \wedge y S \circ T u) \\ & \text{ iff } x R \circ (S \circ T) u\end{aligned}$$

We have shown that $(R \circ S) \circ T = R \circ (S \circ T)$.

Negative Results

Let R be a binary relation on A . Then

1. $R \neq R^{-1}$;
2. composition is not commutative;
3. $R \circ R^{-1} \neq \text{id}_A$.

Proof The way to prove that a property does not hold is to provide a **counter-example**.

A counter-example to part 1 is $R = \{(a, b)\} \subseteq \{a, b\} \times \{a, b\}$.

A counter-example to part 2 is $R = \{(a, a)\}$ and $S = \{(a, b)\}$ where $A = \{a, b\}$. [Part 3 is left as an exercise.](#)

Application to Relational Databases

Here are two example databases:

name	address	number	number	grade
...
Brown, B	5 Lawn Rd.	105	105	A
Jackson, B.	1 Oak Dr.	167		
Smith, J.	9 Elm St.	156	156	A
Walker, S.	4 Ash Gr.	189	189	C
...

Join operator

name	address	number	grade
...
Brown, B	5 Lawn Rd.	105	A
Smith, J	9 Elm St.	156	A
Walker, S	4 Ash Gr.	189	C
...

Notice that candidate 167 did not sit the exam, and so therefore does not appear in the join.

Projection operator

name	address	grade
...
Brown, B	5 Lawn Rd.	A
Smith, J	9 Elm St.	A
Walker, S	4 Ash Gr.	C
...

Select operator

name	address	grade
...
Brown, B	5 Lawn Rd.	A
Smith, J	9 Elm St.	A
...

Question

What is the connection between relational composition and the operators for relational databases?

Equivalence Relations: Motivation

Two Haskell functions $f : A \rightarrow B$ and $g : A \rightarrow B$ are equivalent, written $f = g$, if and only if, for all terms a in type A , then if $f a$ terminates then $g a$ terminates and $f a = g a$, and if $f a$ does not terminate then $g a$ does not terminate.

The following properties are satisfied:

- **reflexivity** $\forall f. f = f$;
- **symmetry** $\forall f_1, f_2. f_1 = f_2 \Rightarrow f_2 = f_1$;
- **transitivity** $\forall f_1, f_2, f_3. f_1 = f_2 \wedge f_2 = f_3 \Rightarrow f_1 = f_3$.

Properties of Relations

Let R be a binary relation on A . Then

1. R is **reflexive** if and only if $\forall x \in A. x R x$;
2. R is **symmetric** if and only if $\forall x, y \in A. x R y \Leftrightarrow y R x$;
3. R is **transitive** iff $\forall x, y, z \in A. x R y \wedge y R z \Rightarrow x R z$.

Examples

1. The equality relation on sets is reflexive, symmetric and transitive.
2. The relations \leq and \subseteq are reflexive and transitive, but not symmetric.
3. The relation $<$ on numbers is transitive, but not reflexive or symmetric.

Proposition

Let R be a binary relation on A .

1. The relation R is reflexive if and only if $\text{id}_A \subseteq R$.
2. The relation R is symmetric if and only if $R = R^{-1}$.
3. The relation R is transitive if and only if $R \circ R \subseteq R$.

The proof is easy and is left as an exercise.

Equivalence Relations

An equivalence relation is a bit like a weak equality: $a R b$ means that a and b are in some sense indistinguishable.

Let A be a set and R a binary relation on A .

The relation R is an **equivalence relation** if and only if R is reflexive, symmetric and transitive.

We sometimes just say that R is an **equivalence**.

Examples of Equivalence Relations

1. Given $n \in \mathcal{N}$ for $n \neq 0$, the binary relation R on \mathcal{Z} defined by $a R b$ iff n divides into $(b - a)$.
2. The binary relation S on the set $\mathcal{Z} \times \mathcal{N}$ defined by $(z_1, n_1) S (z_2, n_2)$ iff $z_1 \times n_2 = z_2 \times n_1$.
3. The identity relation $\text{id}_A : A \times A$.
4. Given a set **Student** and a map $\text{age} : \text{Student} \rightarrow \mathcal{N}$, the relation ' s_1 sameage s_2 iff $\text{age}(s_1) = \text{age}(s_2)$ '.
5. The logical equivalence between formulae, given by $A \equiv B$ if and only if $\vdash A \leftrightarrow B$.

Equivalence classes

Let R be an equivalence relation on A . For any $a \in A$, the **equivalence class** of a with respect to R , denoted $[a]_R$, is defined as

$$[a]_R = \{x \in A : a R x\}.$$

We write $[a]$ instead of $[a]_R$ when the relation R is apparent.

The set of equivalence classes is called the **quotient set** A/R .

Examples

1. Given $n \in \mathcal{N}$ for $n \neq 0$, the binary relation R on \mathcal{Z} defined by $a R b$ iff n divides into $(b - a)$. The set \mathcal{Z}/R represents the integers modulo n .
2. The binary relation S on the set $\mathcal{Z} \times \mathcal{N}$ defined by $(z_1, n_1) S (z_2, n_2)$ iff $z_1 \times n_2 = z_2 \times n_1$. The set $(\mathcal{Z} \times \mathcal{N})/S$ is the usual way of representing the rational numbers.

Proposition

The set of equivalence classes $\{[a] : a \in A\}$ forms a **partition** of A : that is,

- each $[a]$ is non-empty;
- the classes *cover* A : that is, $A = \bigcup_{a \in A} [a]$;
- the classes are disjoint:

$$\forall a, b \in A. [a] \cap [b] \neq \emptyset \Rightarrow [a] = [b].$$

Proof

Given $a \in A$, then $a R a$ by reflexivity and so $a \in [a]$.

$a \in \bigcup_{a \in A} [a]$, and hence the classes cover A .

Suppose $[a] \cap [b] \neq \emptyset$, and let $x \in [a] \cap [b]$.

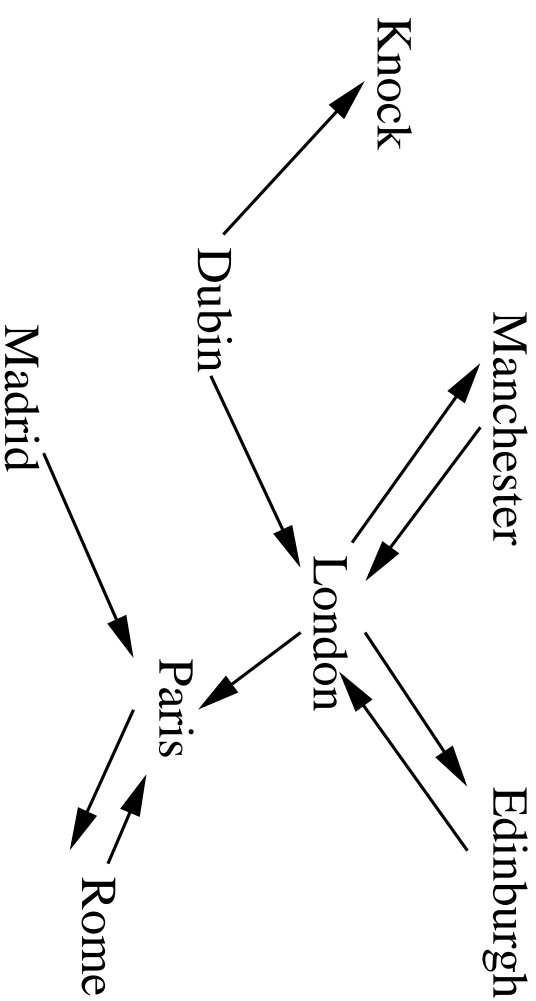
This means that $a R x$ and $b R x$, and hence $x R b$ by symmetry.

Given any $v \in [b]$, observe that $b R v$. Now $a R x$, $x R b$ and $b R v$, so $a R v$ by transitivity. Hence $v \in [a]$ and $[b] \subseteq [a]$.

$[a] \subseteq [b]$ using a similar argument, so $[a] = [b]$.

Transitive Closure: Example

Define a set City of cities and a binary relation R on City such that $a R b$ iff there is a direct flight from a to b . **For example**



Define the relation R^+ by $a R^+ b$ if and only if there is a trip from a to b . We would like to calculate R^+ from R .

Transitive Closure: Example continued

$a R^+ b$ iff there is a path of length n from a to b , for **some** $n \geq 1$.

$a R^n b$ if and only if there is a path of length n from a to b .

An equivalent definition is $a R^+ b$ iff $a R^n b$ for some $n \geq 1$.

The relation is called the **transitive closure** of R .

It is the smallest transitive relation containing R .

Transitive Closure: Example continued (again)

Another way of defining R^n is

$$R^1 = R$$

$$R^2 = R \circ R$$

$$R^3 = R \circ R^2 = R^2 \circ R, \quad \text{since } \circ \text{ is associative}$$

...

$$R^n = R \circ R^{n-1} = R \circ \dots \circ R, \quad n \text{ times}$$

...

$$R^+ = R \cup R^2 \cup \dots \cup R^n \cup \dots = \bigcup_{n \geq 1} R^n$$

Example

Program modules can import other modules.

You have seen this already in the Haskell course.

They can also depend indirectly on modules via some chain of importation:

M depends on M' if M imports M'' and M'' imports M' .

The relation ‘depends’ is the transitive closure of the relation ‘imports’.

Example

Two people are related iff

1. either one is the parent of the other; or
2. they are married; or
3. there is a chain of such relationships joining them directly.

Consider a universal set **People**, and three relations **Married**, **Parent** and **Relative**.

The relation **Relative** is defined using the transitive closure:

$$\text{Relative} = ((\text{Parent} \cup \text{Parent}^{-1}) \cup \text{Married})^+$$

Transitive Closure

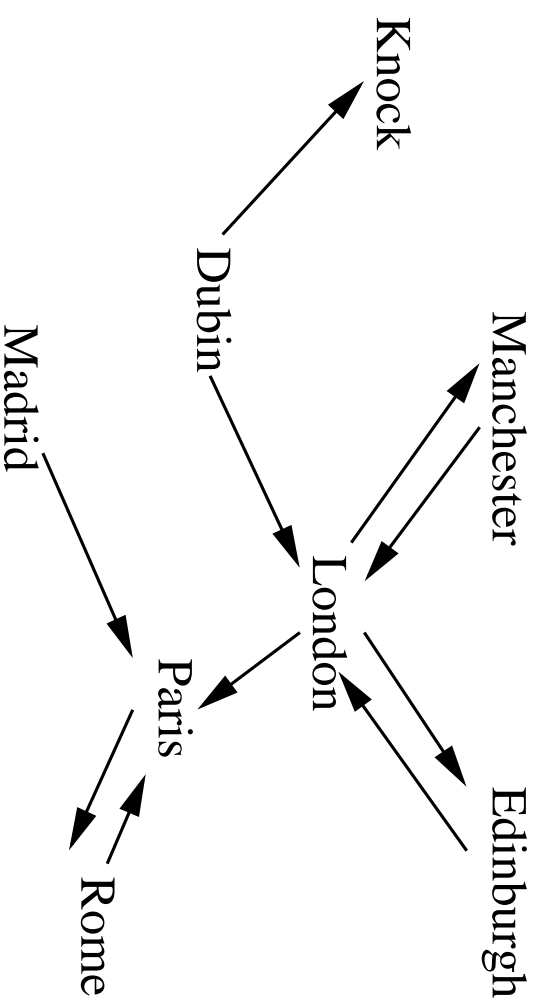
Definition

Let R be a binary relation on A . The **transitive closure** of R , written $t(R)$ or R^+ , is $\bigcup_{n \geq 1} R^n$: that is,

$$R \cup R^2 \cup R^3 \cup R^4 \cup \dots$$

The transitive closure of a binary relation always exists.

Back to the Example



Exercise

Construct the transitive closure.

Computing the Transitive Closure

To calculate R^+ , compute successively

$$R, R \cup R^2, R \cup R^2 \cup R^3, \dots$$

The relation $R \cup R^2 \cup \dots \cup R^n$ represents all paths of length between 1 and n .

If R is **finite**, the process will come to an end.

Suppose the set on which R is defined has n elements.

$$\text{Then } R^+ = R \cup R^2 \cup \dots \cup R^n.$$

Exercise Find such an R with $R^+ \neq R \cup R^2 \cup \dots \cup R^{n-1}$.

Algorithm

We may describe our procedure by the following Kenya-like algorithm:

Input R

$S := R$

$T := R$

$S := R \circ S$

while not $S \subseteq T$ **do**

$T := T \cup S$

$S := R \circ S$

od

Output T

There are many ways of improving the algorithm.

See **Warshall's algorithm** described in Discrete Maths 2.