Sets

Relations

A lot of maths has to do with *bookkeeping*. This section introduces some tools for keeping track of relations between objects and concepts.

Definition: *Relation*

A relation between the elements of the sets A and B is a set of ordered pairs $R \subset A \times B$.

- Intuitively, we think of the elements of the ordered pair as being connected or related in some way.
- For example, let H be the set of all humans and C be the set of all cars. Then

  $$R = \big\{(x, y) \in H \times C : x \text{ owns } y\big\}$$

  is a relation from humans to cars.
- The *order* makes a big difference in the relation: people own cars, not the other way around.

Relations

Example

Let H be the set of all humans and

- $R_P = \{(x, y) \mid x \text{ is a parent of } y\} \subset H \times H$.
- $R_A = \{(x, y) \mid x \text{ is an ancestor of } y\} \subset H \times H$.
- $R_S = \{(x, y) \mid x \text{ is a sibling of } y\} \subset H \times H$.
- $R_B = \{(x, y) \mid x \text{ is a brother of } y\} \subset H \times H$.

We sometimes write aRb as a shorthand for $(a, b) \in R$.

Special relations:

- The **identity relation** on A is

$$id_A = \{(a,a) : a \in A\} = \{(x,y) \in A \times A : x = y\}$$

  In words, the identity relation is "each element is related to itself, and nothing else".

- The **empty relation** from A to B is

$$\emptyset \subset A \times B$$

  e.g. let A be the set of humans and B be the set of dinosaurs. The relation between people and dinosaurs, xRy if "they were born at the same nanosecond" is probably empty. In other words, not one person is related to a dinosaur in this sense.

- The **universal relation** $R = A \times B$.
  "everything in A is related to everything in B"

Inverting relations

- The **inverse relation** of $R \subset A \times B$ is

$$R^{-1} = \{(b, a) \in B \times A : (a, b) \in R\}.$$

Check: The inverse relation of ancestor is descendant; the inverse relationship of parent is child; the inverse relationship of sibling is sibling.
Check: $(R^{-1})^{-1} = R$.

Special relations:

Example

- $R_1 = \{(a, b) \in \mathbb{Z} \times \mathbb{Z} \mid a \neq b\}$.
- $R_2 = \{(a, b) \in \mathbb{Z} \times \mathbb{Z} \mid a \leq b\}$.
- $R_3 = \{(a, b) \in \mathbb{Z} \times \mathbb{Z} \mid a \geq b\}$.
- $R_4 = \{(a, b) \in \mathbb{Z} \times \mathbb{Z} \mid a = b^2 + 1\}$.
- $R_5 = \{(a, b) \in \mathbb{Z} \times \mathbb{Z} \mid a < b^2 + 1\}$.
- $R_6 = \{(a, b) \in \mathbb{Z} \times \mathbb{Z} \mid a = b \text{ or } a = -b\}$.

Visualizing relations

The relation

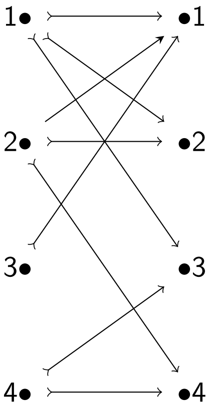
$$R = \{(1,1),(1,2),(2,1),(1,3),(2,2),(2,4),(3,1),(4,3),(4,4)\}$$

can be visualized using a table:

R	1	2	3	4
1	$\times$	$\times$	$\times$	
2	$\times$	$\times$		$\times$
3	$\times$			
4			$\times$	$\times$

Visualizing relations

Or a directed graph:



Composing relations

If $R \subset A \times B$ and $S \subset B \times C$ are two relations, then the **composite** relation $RS \subset A \times C$ is given by

$$RS = \{(a, c) : \exists b \in B \text{ such that } aRb \wedge bSc\}.$$

We often write $a(RS)c$ instead of $(a, c) \in RS$.
The definition looks complicated, but is actually just common sense.

Example

If R_B is the relation "is a brother of" and R_P is the relation "is a parent of", then $R_B R_P$ is the relation "is an uncle of". In this case, the definition says that Bill is an uncle of Jane if and only if there exists some person x such that Bill is a brother of x and x is a parent of Jane.

- What is $R_P R_B$?
- The relation $R_P R_P$ ="grandparent".
  What is $R_A R_A$?
  What is $R_S R_S$?
- Check: $(RS)^{-1} = S^{-1}R^{-1}$.

Properties of relations

Relation R on a set A is

- **reflexive** if: $(a, a) \in R$ for all $a \in A$.
- **symmetric** if: $(a, b) \in R$ whenever $(b, a) \in R$.
- **antisymmetric** if: $(a, b) \in R$ and $(b, a) \in R$ implies $a = b$.
- **transitive** if: $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

Properties of relations

Example

- *reflexive*: $R_{Height} = \{(x, y) \in H \times H : x \text{ and } y \text{ are same height}\}$
  "I am the same height as myself"
- *not* reflexive: R_P (parent)
  "I am not my own parent"
- *transitive*: R_A (ancestor) and R_S (sibling)
  "my ancestor's ancestor is my ancestor"
- *not* transitive: R_P (parent)
  "my parent's parent is not my parent"

Properties of relations

Example

- *symmetric*: R_S
  "if I'm your sibling then you're my sibling"
- *not* symmetric: R_B (brother)
  "just because I'm your brother does not mean you're my brother – you could be my sister"
- *not* symmetric: R_A and R_P (ancestor and parent)
  "just because I'm your ancestor does not mean you're my ancestor"

Equivalence Relations

Definition: *Equivalence Relations*

An equivalence relation is a relation $R \subset A \times A$ that is

- reflexive: $a \sim a$ for all $a \in A$
- symmetric: if $a \sim b$ then $b \sim a$
- transitive: if $a \sim b$ and $b \sim c$, then $a \sim c$.

We often use $\sim$, instead of R, to denote equivalence relations.

- An equivalence relation is something "like" equality. Note that equality is reflexive (because $a = a$ for any a) symmetric (because if $a = b$ then $b = a$) and transitive (because if $a = b$ and $b = c$ then $a = c$).
- Equivalence relations give us a mathematically precise way to talk about relations between objects that are "approximately equal" in some sense or other. Examples are given below.

Practice Questions

Which of the following are equivalence relations:

1. For any set A, the identity relation id_A.
2. For any set A, the universal relation $A \times A$
3. Let H="humans" and R ="have the same age".
4. $\{(a, b) \in \mathbb{R} \times \mathbb{R} : a \leq b\}$
5. $\{(a, b) \in \mathbb{R} \times \mathbb{R} : a^2 = b^2\}$
6. $\{(a, b) \in \mathbb{Z} \times \mathbb{Z} : a + b \text{ is even}\}$
7. Let $\mathcal{P}$="all propositions".
   $R = \{(P, Q) \in \mathcal{P} \times \mathcal{P} : P \equiv Q\}$
8. The brother relation
9. The sibling relation

Equivalence classes

Definition: *Equivalence class*

Let $\sim$ be an equivalence relation on A. Then the set

$$[a]_\sim = \{x \in A : x \sim a\}$$

is called the $\sim$ equivalence class of a.

Describe the equivalence classes for:

1. the identity relation id_A.
2. the universal relation $A \times A$
3. Let H="humans" and R ="have the same age".
5. $\{(a, b) \in \mathbb{R} \times \mathbb{R} : a^2 = b^2\}$
6. $\{(a, b) \in \mathbb{Z} \times \mathbb{Z} : a + b \text{ is even}\}$
7. Let $\mathcal{P}$="all propositions".
   $R = \{(P, Q) \in \mathcal{P} \times \mathcal{P} : P \equiv Q\}$
9. The "pure-sibling+self" relation

What is this all about?

- Equivalence relations capture the notion of abstraction. Abstraction is just a fancy word for "ignoring some details and paying attention to others".
- For example, if all I care about is peoples' ages – not their name, gender, intelligence, etc – then two people of the same age are *identical* as far as I'm concerned. And thus I *group* people by age.
- Similarly, if all I care about is the absolute value of numbers (and not their sign) then the numbers 2 and -2 are *identical* to me and I will group them together.
- Writing down an equivalence relation is a mathematical way of say "I care about *this aspect of things* and *nothing else*". It follows automatically that you're grouping things together.

What is this all about?

Definition: *Partition*

A partition of a set A is a collection $\mathcal{C}$ of subsets of A such that

- any two cells are **disjoint:** if $C_1, C_2 \in \mathcal{C}$, then
$$C_1 \cap C_2 = \emptyset$$
- Every element of A belongs to some cell. That is,
$$\forall a \in A \, \exists C \in \mathcal{C} \text{ such that } a \in C.$$

What is this all about?

Example

- $A = \{1, 2, 3, 4, 5, 6\}$ and $\mathcal{C} = \{\{1, 2, 3\}, \{4\}, \{5, 6\}\}$.
- $A = \mathbb{Z}$ and $\mathcal{C} = \{$even numbers, odd numbers$\}$.
- $A = \mathbb{Z}$ and $\mathcal{C} = \{\{0\}, \{-1, 1\}, \{-2, 2\}, \{-3, 3\}, \dots\}$.
- $A =$"humans" and $\mathcal{C} = \{$people $<$1 year, people 1-2 years, $\dots\}$
- $A =$"humans" and $\mathcal{C} = \{$people from Texas, everyone else$\}$

Partitions $\leftrightarrow$ equivalence relations

We can convert back and forth between partitions and equivalence relations.

- Partition $\to$ equivalence relations
  Check:
  If $\sim$ is an equivalence relation on A then

  $$\mathcal{C} = \{[a]_\sim : a \in A\}$$

  is a partition of A.
- Partition $\leftarrow$ equivalence relation
  Check:
  If $\mathcal{C}$ is a partition of A, then the relation

  $$a \sim b \text{ iff } \left(\exists C \in \mathcal{C} \text{ satisfying } a \in C \text{ and } b \in C\right)$$

  is an equivalence relation on A.

Partial orders

Equivalence relations were useful for expressing, precisely, that "things are the same in some respect". Partial orders are useful for expressing, precisely, that "one thing is *greater than* another in some respect".

Example

- The relation $\leq$ on $\mathbb{R}$ or $\mathbb{Z}$
- The subset relation $\subset$ on the powerset $\mathcal{P}(A) =$"set of all subsets of A".
- The "ancestor+self relation" on humans.

Visualizing partial orders: Hasse diagrams

Draw a graph where:

- vertices are labelled by elements of A
- if $a \preceq b$ and $a \neq b$ then draw a **below** b
- if $a \preceq b$ and there is no element c (different from a and b) satisfying $a \preceq c$ and $c \preceq b$, then draw an edge from a to b.

Total orders

Let $\preceq$ be a partial order on A. If $a \preceq b$ or $b \preceq a$ then we say a and b are comparable. Otherwise, they are incomparable.
A total order is an order where all elements are comparable.

Example

Total orders:

- $\leq$ on $\mathbb{Z}$ or $\mathbb{R}$
- humans, with relation $a \preceq b$ if b is taller than a (measured to picometers)

Not total orders:

- The subset relation on a powerset
- the "ancestor+self" relation on humans

Databases

An major theme in computer science is finding useful ways to represent information. A breakthrough[1] was the invention of relational databases by E.F. Codd in 1970.

Definition: *n-ary relations*

Let $A_1, \ldots, A_n$ be sets. An **n-ary relation** is a subset of $A_1 \times \cdots \times A_n$. The sets $A_1, \ldots, A_n$ are called the **domains** of the relation, and n is called its **degree**.

A database consists of records each of which is an n-tuples made of up fields. The fields are the entries of the n-tuples.

The relations used to represent databases are called tables, because the relations can often be represented as tables with "records $\leftrightarrow$ rows", "attributes $\leftrightarrow$ columns", and "fields $\leftrightarrow$ cells".

A domain in an n-ary relation is called a primary key if the value of the entry in this domain determines the n-tuple.

[1] Codd receive the Turing award for inventing relational databases; Larry Ellison made over $50 billion by commercializing them.

Database Example

A database of student records has fields corresponding to $A_1 =$ Names, $A_2 =$ ID_numbers, $A_3 =$ Majors, and $A_4 =$ GPAs.
A sample of six records in the student database is the 4-ary relation below:

$$R = \Big\{(\text{Hollande}, 231455, \text{Software Engineering}, 2.99),\\ (\text{Jiabao}, 888323, \text{Physics}, 3.90),\\ (\text{Modi}, 102147, \text{Software Engineering}, 3.45),\\ (\text{Obama}, 453876, \text{Mathematics}, 3.49),\\ (\text{Putin}, 678543, \text{Mathematics}, 3.45),\\ (\text{Rousseff}, 786576, \text{Psychology}, 3.88)\Big\}$$

Database Example

Table: **Students**

Name	ID_number	Major	GPA
Hollande	231455	Software Engineering	2.99
Jiabao	888323	Physics	3.90
Modi	102147	Software Engineering	3.45
Obama	453876	Mathematics	3.49
Putin	678543	Mathematics	3.45
Rousseff	786576	Psychology	3.88

Operations on databases: Selection

Definition: *Selection*

Let R be an n-ary operation and C be a condition that elements in R might satisfy. Then the selection operator s_C maps the n-ary relation R to the n-ary relation of all tuples from R that satisfy C.
Selection is just picking rows from a table.

Example

Suppose the condition is $C =$"student is a maths major". Applying s_C to our table yields:

Table: **Maths majors**

Name	ID_number	Major	GPA
Obama	453876	Mathematics	3.49
Putin	678543	Mathematics	3.45

Operations on databases: Projection

Definition: *Projection*

The projection operator $P_{i_1,i_2,\ldots,i_m}$ maps the n-tuple $(a_1,\ldots,a_n)$ to the m-tuples $(a_{i_1},a_{i_2},\ldots,a_{i_m})$ where $m \leq n$. Projection is just throwing away columns of a table.

Example

Applying the projection $P_{1,2}$ yields:

Table: **Enrollments**

Student	Major	Course
Harper	Biology	BIOL132
Harper	Biology	BIOL111
Harper	Biology	BIOL114
Abe	Network Engineering	NWEN241
Abe	Network Engineering	NWEN242
Merkel	Mathematics	MATH301
Merkel	Mathematics	MATH313

Operations on databases: Projection

Example

Table: **Name and Majors**

Name	Major
Harper	Biology
Abe	Network Engineering
Merkel	Mathematics

Operations on databases: Join

Definition: *Join*

Let R be a relation of degree m and S a relation of degree n. The **join** $J_p(R, S)$ where $p \leq m$ and $p \leq n$ is a relation of degree $m + n - p$ that consists of all $(m+n-p)$-tuples $(a_1, a_2, \ldots, a_{m-p}, c_1, \ldots, c_p, b_1, \ldots, b_{n-p})$, where the m-tuple $(a_1, a_2, \ldots, a_{m-p}, c_1, \ldots, c_p)$ belongs to R and the n-tuple $(c_1, c_2, \ldots, c_p, b_1, \ldots, b_{n-p})$ belongs to S.
Join combines two tables, without double-counting the overlap.

Operations on databases: Join

Example

Applying the join J_2 gives:

Table: **Teaching_assignments**

Lecturer	Department	Course_number
Avila	Biology	BIOL132
Avila	Biology	BIOL111
Bhargava	Network Engineering	NWEN241
Bhargava	Network Engineering	NWEN242
Hairer	Mathematics	MATH301
Hairer	Mathematics	MATH313
Mirzakhani	Physics	PHYS114
Mirzakhani	Physics	PHYS122

Operations on databases: Join

Example

Table: **Class_schedule**

Department	Course_number	Room	Time
Biology	BIOL132	MY101	13:00
Biology	BIOL111	MC103	14:00
Network Engineering	NWEN241	MY101	14:00
Network Engineering	NWEN242	KK301	11:00
Mathematics	MATH301	HM101	11:00
Mathematics	MATH313	KK302	16:00
Physics	PHYS114	HM101	10:00
Physics	PHYS122	MY101	09:00

Operations on databases: Join

Example

Table: **Teaching_schedule**

Lecturer	Department	Course_number	Room	Time
Avila	Biology	BIOL132	MY101	13:00
Avila	Biology	BIOL111	MC103	14:00
Bhargava	Network Engineering	NWEN241	MY101	14:00
Bhargava	Network Engineering	NWEN242	KK301	11:00
Hairer	Mathematics	MATH301	HM101	11:00
Hairer	Mathematics	MATH313	KK302	16:00
Mirzakhani	Physics	PHYS114	HM101	10:00
Mirzakhani	Physics	PHYS122	MY101	09:00

Functions

- When mathematicians first started working with functions, they thought of them in terms of formulas.
- For example, the function $f(x) = x^2 + 1$ is defined by its formula.
- However, it steadily became useful to work in increasingly abstract settings, until eventually mathematicians converged on the modern definition of a function, which is that a function is a special kind of relation.
- This more abstract perspective takes some getting used to.
- The payoff is that it is much more flexible than the old fashioned formula-based definition of a function, and can be used in many more settings that arise in computer science.

Functions

Definition: *Domain* and *Range*

Let R be a relation from A to B. The domain of R is the set

$$\text{dom}(R) = \{a \in A : aRb \text{ for some } b \in B\} \subset A$$

The range of R is

$$\text{ran}(R) = \{b \in B : aRb \text{ for some } a \in A\} \subset B$$

Intuitively, the *domain* is the set of inputs to the relation, and the *range* is the set of outputs.

Example

What are the domain and range of the relations $R_2 = \{(x, y) \in \mathbb{R} \times \mathbb{R} : y = x^2\}$ and $R_{-1} = \{(x, y) \in \mathbb{R} \times \mathbb{R} : y = \frac{1}{x}\}$?

Functions

Definition: *Function*

A relation f on sets A and B is a function if

- the domain of f is A and
- if $(a, b) \in f$ and $(a, c) \in f$ then $b = c$.

Instead of $(a, b) \in f$, we normally write $f(a) = b$ for functions. Similarly, instead of $f \subset A \times B$, we normally write $f : A \to B$ for functions.

Key idea:

A function f describes a process that transforms an input a into an output b. Functions have the following useful properties:

- every input can be processed
  ("the domain of f is A")
- for any input, the process produces just one output
  ("if $(a, b) \in f$ and $(a, c) \in f$ then $b = c$")

Practice Questions

Which are the following relations are functions? Why or why not?

- Parent
- Mother
- Sibling
- Eldest sibling
- $R = \{(x, y) \in \mathbb{R} \times \mathbb{R} : x^2 + y^2 = 1\}$
- $R = \{(x, y) \in \mathbb{R} \times \mathbb{R} : y = \frac{1}{x}\}$
- $R = \left\{(x, y) \in \mathbb{R} \setminus \{0\} : y = \frac{1}{x}\right\}$
- $R = \{(x, y) \in \mathbb{R} \times \mathbb{R}_+ : x^2 + y^2 = 1\}$
- $R = \{(x, y) \in \mathbb{R} \times \mathbb{R} : y = \sqrt{x}\}$

Important: whether or not a relation is a function depends on the domain!

Codomain

If $f : A \to B$ is a function, then B is its **codomain**.

Definitions

A function $f : A \to B$ is

- onto or surjective if for all $y \in B$ there exists an $x \in A$ such that $f(x) = y$.
- one-to-one or injective if for all $x, y \in A$, if $f(x) = f(y)$ then $x = y$.
- bijective if it is one-to-one and onto.

Note: a function is surjective iff its codomain is equal to its range. Why?

Examples

Example

- *Neither injective nor surjective:*
  $f : \mathbb{R} \to \mathbb{R} : x \mapsto x^2$
- *Injective (but not surjective):*
  $f : \mathbb{Z} \to \mathbb{Z} : x \mapsto 2x$
  $f : \mathbb{R} \to \mathbb{R} : x \mapsto e^x$
- *Surjective (but not injective):*
  $f : \mathbb{R} \to \mathbb{R} : x \mapsto x^3 - x + 1$
- *Bijective:*
  $f : \mathbb{R} \to \mathbb{R} : x \mapsto x^3$
  $f : \mathbb{R} \to \mathbb{R} : x \mapsto 2x$
  $f : \mathbb{R}_+ \to \mathbb{R} : x \mapsto \log x$

Inverse functions

Definition: *Inverse*

The inverse of function $f : A \to B$ is the relation

$$f^{-1} = \{(y, x) \in B \times A : f(x) = y\}$$

Example

Is the inverse of a function a function? Give a proof or counterexample.

Theorem

The inverse of a bijection is a bijection (and therefore also a function).

Composing functions

Definition: *Composition*

If $f : A \to B$ and $g : B \to C$ are functions then the composition of f and g, denoted by $g \circ f : A \to C$, is defined by

$$g \circ f(a) = g(f(a)).$$

Theorem

If f and g are bijections, then so is $g \circ f$.