Foundations of Computer Science Comp109

University of Liverpool
Boris Konev
konev@liverpool.ac.uk
http://www.csc.liv.ac.uk/~konev/COMP109

Introduction

Comp109 Foundations of Computer Science

Information

Lecturer

- Prof Boris Konev
- Office: 1.15 Ashton building
- Email: konev@liverpool.ac.uk
- Course web page: http://www.csc.liv.ac.uk:/~konev/COMP109

~30 lectures + 2 class tests + 11 tutorials

Module aims

- To introduce the notation, terminology, and techniques underpinning the discipline of Theoretical Computer Science.
- To provide the mathematical foundation necessary for understanding datatypes as they arise in Computer Science and for understanding computation.
- To introduce the basic proof techniques which are used for reasoning about data and computation.
- To introduce the basic mathematical tools needed for specifying requirements and programs

Module outcomes

At the end of this module students should be able to:

- Understand how a computer represents simple numeric data types; reason about simple data types using basic proof techniques;
- Interpret set theory notation, perform operations on sets, and reason about sets;
- Understand, manipulate and reason about unary relations, binary relations, and functions;
- Apply logic to represent mathematical statement and digital circuit, and to recognise, understand, and reason about formulas in propositional and predicate logic;
- Apply basic counting and enumeration methods as these arise in analysing permutations and combinations.

Assessment

- Exam: 80%
  - Multiple-choice test
- Continuous Assessment: 20%
  - Assessment 1. Covers Parts 1-4
    - Class test
    - Your contribution during tutorials
  - Assessment 2. Covers Parts 5-7
    - Class test
    - Your contribution during tutorials

Lectures

We will have three lectures per week this term.
Your timetable is on *Liverpool Life*.

- Read the slides before (and after) the lecture.
- Take notes. (University is a lot different from school.)
- I will write on the slides.
- Notes often make no/little sense

  PDFs will appear on
  http://cgi.csc.liv.ac.uk/~konev/COMP109
  - These notes are not a replacement for your own notes!
- Please study as you go along.

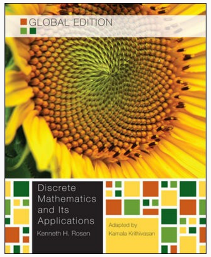
Tutorials

- The class will be divided into tutorial groups. You will be able to find out which group you are in from your personal timetable.
- Each tutorial group meets once a week.
- Problem sheets will become available on the module web page (https://intranet.csc.liv.ac.uk/~konev/COMP109).
  Try to solve the problems before your tutorial. Part of your continuous assessment mark will be based on your contribution during tutorials, including
  1. making reasonable attempts to solve the problems, and bringing these (in writing) to tutorials, and
  2. your contribution to group discussions in the tutorial group.

  You will hand your work in at the end of each tutorial and get it back the following week.

Core textbook

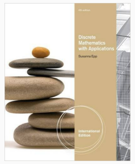
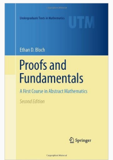
- K. Rosen. Discrete Mathematics and Its Applications, McGraw-Hill. 7th edition, 2012.

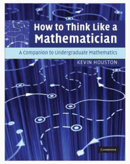


(any edition, including the US edition, is OK)

Recommended books

- E. Lehman, F. T. Leighton and A. R. Meyer Mathematics for Computer Science. **Free book**
- S. Epp. Discrete Mathematics with Applications, Cengage Learning. 4th edition, 2011.
- E. Bloch. Proofs and Fundamentals, Springer. 2nd edition, 2011
- K. Houston. How to Think Like a Mathematician, Cambridge University Press. 2009



Course contents

- Part 1. Number Systems and Proof Techniques
- Part 2. Set Theory
- Part 3. Functions
- Part 4. Relations
- Part 5. Propositional Logic & Digital Circuits
- Part 6. Combinatorics & Probability

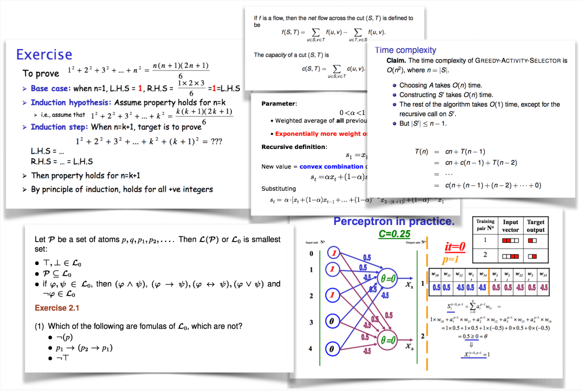
So, this is maths...

- The module does not depend upon A-level maths.
- You can get a first in this module even if you did badly at GCSE maths.
- To do well in this module, you have to work **hard**.

But Who Needs Maths?

You do!

Comp108, Comp 202, Comp226, Comp304, Comp305, Comp309,...



Datatypes

A datatype in a programming language is a set of values and the operations on those values. The datatype states

- the possible values for the datatype
- the operations that can be performed on the values
- the way that values are stored.

Number systems and datatypes

- The most basic datatypes
  - Natural Numbers
  - Integers
  - Rationals
  - Real Numbers
  - Prime Numbers

Number systems and proof techniques

- Proof Techniques
  - Finding a counter-example
  - Proof by contradiction
  - Proof by Induction

These are used, for example, to reason about data types and to reason about algorithms.

We use proof techniques, both to show that an algorithm is correct and to show that it is efficient.

Data collections

Most applications work with collections of data items

- Price list
- Phonebook
- Climate change data
- Stock exchange data
- ...

Sets

A set is a well-defined collection of objects. The objects in the set are called the elements or members of the set.

- The set containing the numbers 1, 2, 3, 4 and 5 is written $\{1, 2, 3, 4, 5\}$.
- The number 3 is an element of the set, that is, $3 \in \{1, 2, 3, 4, 5\}$.
- The number 6 is not an element of the set, that is, $6 \notin \{1, 2, 3, 4, 5\}$.
- The set {dog, cat, mouse} is a set with three elements: dog, cat and mouse.

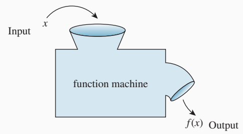
Young man, in mathematics you don't understand things. You just get used to them. (John von Neumann)

Some important sets

- $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ (the **natural** numbers)
- $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ (the **integers**)
- $\mathbb{Q} = \{p/q \mid p \text{ and } q \text{ are integers}, q \neq 0\}$ (the **rationals**)
- $\mathbb{R}$: (**real** numbers)

Functions

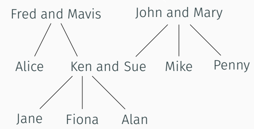
- A function is just a map from a set of inputs to a set of outputs.
  - This is exactly what an algorithm computes.
- Functions can also be used to determine how long algorithms take to run.



Examples:

- $y = x^2$
- $y = \sin(x)$
- first letter of your name

Family relations



Write down

- $R = \{(x, y) \mid$ x is a grandfather of y $\}$;

Relations and databases

Databases: Most databases store information as *relations* over *sets*. We need precise notation and terminology for sets and relations in order to talk about databases. Basic mathematical facts about relations and sets are required to understand how a database is designed and implemented.

Logic and specification languages

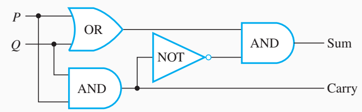
How can we specify what a program should do? Natural languages can be long-winded and ambiguous and are not appropriate for intricate problems.

A formal language without ambiguous statements is required.

Propositional and Predicate Logic are the most important formal languages for specifying programs.

Propositional logic and digital circuits

- Syntax: formulas and formal representations
- Semantics: interpretations and truth tables
- Logic and digital circuits
- Computer arithmetic
- Logical equivalence



Combinatorics

Combinatorics includes the study of counting and also the study of discrete structures such as graphs. It is essential for analysing the efficiency of algorithms.

Combinatorics

- Notation for sums and products, including the factorial function.
- Principles for counting permutations and combinations, for example, to enable you to solve the problem on the following slide.

Applications to discrete probability

The draw selects a set of six different numbers from $1, 2, \ldots, 49$. Each choice is equally likely.

You choose a set of six numbers in advance. If your numbers come up, you win the jackpot. What is the probability of this event?

Reading mathematics[1]

- Read with a purpose
- Choose a book at the right level
- Read with pen and paper at hand
- Don't read it like a novel
- Identify what is important
- Stop periodically to review
- Read statements first—proofs later
- Do the exercises and problems
- Reflect
- Write a summary

[1] *How to think like a mathematician* by K. Houston.

Appendix: Greek letters

Alpha	α A	Iota	ι I	Sigma	σ Σ
Beta	β B	Kappa	κ K	Tau	τ T
Gamma	γ Γ	Lambda	λ Λ	Upsilon	υ Υ
Delta	δ Δ	Mu	μ M	Phi	ϕ Φ
Epsilon	ϵ E	Nu	ν N	Chi	χ X
Zeta	ζ Z	Omicron	o O	Psi	ψ Ψ
Eta	η E	Pi	π Π	Omega	ω Ω
Theta	θ Θ	Rho	ρ R		

Foundations of Computer Science Comp109

University of Liverpool
Boris Konev
konev@liverpool.ac.uk
http://www.csc.liv.ac.uk/~konev/COMP109

Part 1. Number Systems and Proof Techniques

Comp109 Foundations of Computer Science

Reading

- S. Epp. *Discrete Mathematics with Applications*
  Chapter 4, Sections 5.2 and 5.3.
- E. Bloch. *Proofs and Fundamentals*
  Chapter 2, Section 6.3.
- K. Rosen. *Discrete Mathematics and Its Applications*
  Section 5.1.

Contents

- The most basic datatypes
  - Natural Numbers
  - Integers
  - Rationals
  - Real Numbers
  - Prime Numbers
- Proof Techniques
  - Direct proof and disproof
    - Disproof by counterexample
    - Existence proof
    - Generalising from the generic particular
    - ...
  - Indirect Proof
    - Proof by contradiction
    - ...
  - Proof by mathematical induction

What is a number?

The natural numbers

$$0, 1, 2, 3, \ldots$$

Key property: Any natural number can be obtained from 0 by applying the operation $S(n) = n + 1$ some number times.

Examples: $S(0) = 1$.

$S(S(0)) = 2$.

$S(S(S(0))) = 3$.

Prime numbers

A prime number is a integer greater than 1 which has exactly two divisors that are positive integers: 1 and itself.

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, \ldots$$

Every natural number greater than 1 can be written as a unique product of prime numbers.

Examples: $6 = 2 \times 3$. $15 = 3 \times 5$. $1400 = 2^3 \times 5^2 \times 7$.

Example: prime and composite numbers

1. Is 1 prime?
2. Is every integer greater than 1 either prime or composite?
3. Write the first six prime numbers.
4. Write the first six composite numbers.

Beyond naturals

The Integers $\ldots, -2, -1, 0, 1, 2, \ldots$

The Rational Numbers all numbers that can be written as $\frac{m}{n}$ where m and n are integers and n is not 0.

Reminder: Algebraic manipulation

Solving and computing

Mathematics underpins STEM subjects. In many cases, we are concerned with solving and computing

The quadratic equation $2x^2 + 6x + 7 = 0$ has roots α and β.

Write down the value of $\alpha + \beta$ and the value of $\alpha\beta$.

Complete the table of values for $y = 3 - x^2$

x	-3	-2	-1	0	1	2	3
y		-1	2		2		-6

Work out $\frac{1}{3} \times \frac{1}{5}$

Find the general solution, in degrees, of the equation

$$2\sin(3x + 45°) = 1$$

5 miles = 8 kilometres

Which is longer, 26 miles or 45 km?

Statements

Which of the following are true?

- "26 miles is longer than 45 km."
- An integer doubled is larger than the integer.
- The sum of any two odd numbers is even.

The moral of the story

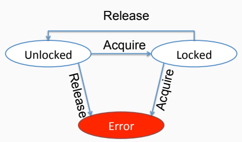
- We can't believe a statement just because it appears to be true.

We need a proof that the statement is true or a proof that it is false.
Do we care?

Example: Drivers behaviour[1]

```
do {
    KeAcquireSpinLock();
    nPacketsOld = nPackets;
    if (request) {
        request = request->Next;
        KeReleaseSpinLock();
        nPackets++;
    }
} while (nPackets != nPacketsOld);
KeReleaseSpinLock();
```

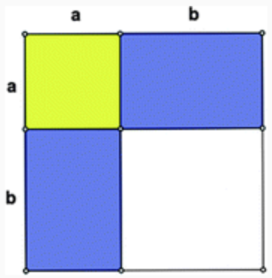
Does this code obey the locking rules?



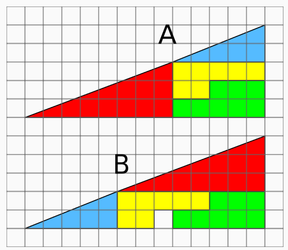
You don't need to understand the actual code!

[1] from Microsoft presentations on Static Driver Verifier (part of Visual Studio)

Historical detour: Visual proofs



Visual proof of $(a+b)^2 = a^2 + 2ab + b^2$



Visual "proof" of 32.5 = 31.5

Proofs

- A mathematical proof is as a **carefully reasoned argument** to convince a sceptical listener (often yourself) that a given statement is true.
- Both discovery and proof are integral parts of problem solving. When you think you have discovered that a certain statement is true, try to figure out why it is true.
- If you succeed, you will know that your discovery is genuine. Even if you fail, the process of trying will give you insight into the nature of the problem and may lead to the discovery that the statement is false.

Example: Odd and even numbers

Definition
An integer n is **even** if, and only if, n equals twice some integer.
An integer n is **odd** if, and only if, n equals twice some integer plus 1.

Symbolically, if n is an integer, then
n is even $\Leftrightarrow \exists$ an integer k such that $n = 2k$.
n is odd $\Leftrightarrow \exists$ an integer k such that $n = 2k + 1$.

Notice the use of $\Leftrightarrow$ $\exists$ $\forall$.

Example: Properties of odd and even numbers

Use the definitions of even and odd to justify your answers to the following questions.

Definition
n is even $\Leftrightarrow \exists$ an integer k such that $n = 2k$.
n is odd $\Leftrightarrow \exists$ an integer k such that $n = 2k + 1$.

1. Is 0 even?
2. Is 301 odd?

Example: Properties of odd and even numbers

Definition
n is even $\Leftrightarrow \exists$ an integer k such that $n = 2k$.
n is odd $\Leftrightarrow \exists$ an integer k such that $n = 2k + 1$.

3. If a and b are integers, is $6a^2b$ even?
4. If a and b are integers, is $10a + 8b + 1$ odd?
5. Is every integer either even or odd?

Existence proofs

Statements of the form $\exists x\, Q(x)$

Examples:

1. Prove the following: $\exists$ an even integer n that can be written in two ways as a sum of two prime numbers.
2. Suppose that r and s are integers. Prove the following: $\exists$ an integer k such that $22r + 18s = 2k$.

Constructive proof

- One way to prove
$$\exists x\, Q(x)$$
is to find an x in that makes $Q(x)$ true.

Proving universal statements

The vast majority of mathematical statements to be proved are universal. In discussing how to prove such statements, it is helpful to imagine them in a standard form:
$$\forall x \text{ if } P(x) \text{ then } Q(x)$$

For example,

- If a and b are integers then $6a^2b$ is even.

Proving universal statements: The method of exhaustion

Some theorems can be proved by examining relatively small number of examples.

- Prove that $(n + 1)^3 \geq 3^n$ if n is a positive integer with $n \leq 4$.
  - $n = 1$
  - $n = 2$
  - $n = 3$
  - $n = 4$
- Prove for every natural number n with $n < 40$ that $n^2 + n + 41$ is prime.

Motivating example: "Mathematical trick"

Pick any number, add 5, multiply by 4, subtract 6, divide by 2, and subtract twice the original number. The answer is 7.

Step	Visual Result	Algebraic Result
Pick a number.	□	x
Add 5.	□|||||	$x + 5$
Multiply by 4.	□||||| □||||| □||||| □|||||	$(x + 5) \cdot 4 = 4x + 20$
Subtract 6.	□|| □|| □||||| □|||||	$(4x + 20) - 6 = 4x + 14$
Divide by 2.	□|| □|||||	$\frac{4x + 14}{2} = 2x + 7$
Subtract twice the original number.	|| |||||	$(2x + 7) - 2x = 7$

Generalising from the Generic Particular

The most powerful technique for proving a universal statement is one that works regardless of the choice of values for x.

To show that every x satisfies a certain property, suppose x is a particular but arbitrarily chosen and show that x satisfies the property.

Method of direct proof

- Express the statement to be proved in the form
  "$\forall x,\ \text{if } P(x) \text{ then } Q(x)$."
  (This step is often done mentally.)
- Start the proof by supposing x is a particular but arbitrarily chosen element for which the hypothesis $P(x)$ is true.
  (This step is often abbreviated "Suppose P(x).")
- Show that the conclusion $Q(x)$ is true by using definitions, previously established results, and the rules for logical inference.

Prove that the sum of any two even integers is even

Prove that every integer is rational

Prove that the sum of any two rational numbers is rational

Prove that the product of any two rational numbers is rational

Prove that the double of a rational number is rational

Prove for all integers n, if n is even then n^2 is even

Prove by cases: Combine generic particulars and proof by exhaustion

Statement: For all integers n, $n^2 + n$ is even

Case 1: n is even

Case 2: n is odd

How about

Prove for all integers m and n, if $m^2 = n^2$ then $m = n$?

Disproving universal statements by counterexample

To disprove a statement means to show that it is false. Consider the question of disproving a statement of the form

$$\forall x,\ \text{if } P(x) \text{ then } Q(x).$$

Showing that this statement is false is equivalent to showing that its negation is true. The negation of the statement is existential:

$$\exists x \text{ such that } P(x) \text{ and not } Q(x).$$

Is it true that for every positive integer n, $n^2 \geq 2n$?

Indirect proofs

- In a direct proof you start with the hypothesis of a statement and make one deduction after another until you reach the conclusion.
- Indirect proofs are more roundabout. One kind of indirect proof, argument by contradiction, is based on the fact that either a statement is true or it is false but not both.
- So if you can show that the assumption that a given statement is not true leads logically to a contradiction, impossibility, or absurdity, then that assumption must be false: and, hence, the given statement must be true.

Motivating example: Trial and error

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Use proof by contradiction to show that there is no greatest integer

Use proof by contradiction to show that there is no smallest positive rational number

Use proof by contradiction to show that no integer can be both even and odd

Use proof by contradiction to show that there is no greatest prime number

Let $f(x) = 2x + 5$. Prove that if $x \neq y$ then $f(x) \neq f(y)$

Direct proof

Proof by contradiction

When to use indirect proof

- Many theorems can be proved either way. Usually, however, when both types of proof are possible, indirect proof is clumsier than direct proof.
- In the absence of obvious clues suggesting indirect argument, try first to prove a statement directly. Then, if that does not succeed, look for a counterexample.
- If the search for a counterexample is unsuccessful, look for a proof by contradiction

The real numbers

All (decimal) numbers — distances to points on a number line.

Examples.

- -3.0
- 0
- 1.6
- $\pi = 3.14159\ldots$

A real number that is not rational is called irrational.

But are there any irrational numbers?

Proving that $\sqrt{2}$ is not a rational number

Proof by contradiction.

- If $\sqrt{2}$ were rational then we could write it as $\sqrt{2} = x/y$ where x and y are integers and y is not 0.
- By repeatedly cancelling common factors, we can make sure that x and y have no common factors so they are not both even.
- Then $2 = x^2/y^2$ so $x^2 = 2y^2$ so x^2 is even. This means x is even, because the square of any odd number is odd.

the proof continued

- Let $x = 2w$ for some integer w.
- Then $x^2 = 4w^2$ so $4w^2 = 2y^2$ so $y^2 = 2w^2$ so y^2 is even so y is even.
- This contradicts the fact that x and y are not both even, so our original assumption, that $\sqrt{2}$ is rational, must have been wrong.

Prove that $1 + 3\sqrt{2}$ is irrational

Mathematical induction

- Mathematical induction is one of the more *recently* developed techniques of proof in the history of mathematics.
- It is used to check conjectures about the outcomes of processes that occur repeatedly and according to definite patterns.
- In general, mathematical induction is a method for proving that a property defined for integers n is true for all values of n that are greater than or equal to some initial integer

Example: Domino effect

One domino for each natural number, arranged in order.

- I will push domino 0 (the one at the front of the picture) towards the others.
- For every natural number m, if the m'th domino falls, then the $(m+1)$st domino will fall.

Conclude: All of the Dominoes will fall.

Proving by induction that a property holds for every natural number n

- Prove that the property holds for the natural number $n = 0$.
- Prove that if the property holds for $n = m$ (for any natural number m) then it holds for $n = m + 1$.

The validity of proof by mathematical induction is generally taken as an axiom. That is why it is referred to as the principle of mathematical induction rather than as a theorem.

A proof of a property by induction looks like this

Base Case: Show that the property holds for $n = 0$.

Inductive Step: Assume that the property holds for $n = m$. Show that it holds for $n = m + 1$.

Conclusion: You can now conclude that the property holds for every natural number n.

Example: Proof by induction

For every natural number n,

$$0 + 1 + \cdots + n = \frac{n(n+1)}{2}.$$

Base Case: Take $n = 0$. The left-hand-side and the right-hand-side are both 0 so they are equal.

Inductive Step: Assume that the property holds for $n = m$, so

$$0 + 1 + \cdots + m = \frac{m(m+1)}{2}.$$

Now consider $n = m + 1$. We must show that

$$0 + 1 + \cdots + m + (m+1) = \frac{(m+1)(m+2)}{2}.$$

Proof continued

Since

$$0 + 1 + \cdots + m = \frac{m(m+1)}{2}.$$

$$\begin{aligned} 0 + 1 + \cdots + m + (m+1) &= \frac{m(m+1)}{2} + m + 1 \\ &= \frac{m(m+1) + 2(m+1)}{2} \\ &= \frac{(m+1)(m+2)}{2} \end{aligned}$$

Other starting values

Suppose you want to prove a statement not for all natural numbers, but for all integers greater than or equal to some particular natural number b

Base Case: Show that the property holds for $n = b$.

Inductive Step: Assume that the property holds for $n = m$ for any $m \geq b$. Show that it holds for $n = m + 1$.

Conclusion: You can now conclude that the property holds for every integer $n \geq b$.

Example: Proof by induction

For all integers $n \geq 8$, n¢ can be obtained using 3¢ and 5¢ coins.

Base Case: For $n = 8$, 8¢ = 3¢ + 5¢.

Inductive Step: Suppose that m¢ can be obtained using 3¢ and 5¢ coins for any $m \geq 8$. We must show that $(m+1)$¢ can be obtained using 3¢ and 5¢ coins.

Consider cases

- There is a 5¢ coin among those used to make up the m¢.
  - Replace the 5¢ coin with two 3¢ coins. We obtain $(m+1)$¢.
- There is no 5¢ coin among those used to make up the m¢.
  - There are three 3¢ coins ($m \geq 8$).
    - Replace the three 3¢ coins with two 5¢ coins

Example: Proof by induction

For every integer $n \geq 3$, $4^n > 2^{n+2}$.

Base Case: Take $n = 3$. Then $4^n = 4^3 = 64$. Also, $2^{n+2} = 2^5 = 32$. So $4^n > 2^{n+2}$.

Inductive Step: For any $m \geq 3$, assume that the statement $4^m > 2^{m+2}$ is true. (This is called the "inductive hypothesis".) Now consider $n = m + 1$. We must show that $4^{m+1} > 2^{(m+1)+2} = 2^{m+3}$.

Here is the calculation. $4^{m+1} = 4 \times 4^m$. But by the inductive hypothesis, $4 \times 4^m > 4 \times 2^{m+2}$. Finally,

$$4 \times 2^{m+2} > 2 \times 2^{m+2} = 2^{m+3}.$$

Using induction to show that a program is correct

What does the following program do?

```
i = 0
M = 0
mylist = [1, 2, 6, 3, 4, 5]
while i < len(mylist):
    M = max(M, mylist[i])
    i = i + 1
print M
```


Using induction to show that a program is correct

```
i = 0
M = 0
mylist = [1, 2, 6, 3, 4, 5]
while i < len(mylist):
    M = max(M, mylist[i])
    i = i + 1
print M
```

Property: After the statement M = max(M , mylist[i]) gets executed, the value of M is max(mylist[0],...,mylist[i]).

Proof by induction

Property: After the statement M = max(M, mylist[i]) gets executed, the value of M is max(mylist[0],...,mylist[i]).

Base Case: Take i=0. Before the statement, M=0, so the statement assigns M to be the maximum of 0 and mylist[0], which is mylist[0].

Inductive Step: Assume that the statement is true for i=m for some m≥ 0. Now consider i=m+1. The statement assigns M to be the maximum of mylist[m+1] and max(mylist[0],...,mylist[m]), so after the statement, M is max(mylist[0],...,mylist[m+1]).

Strong induction

- Prove that the property holds for the natural number $n = 0$.
- Prove that **if** the property holds for $n = 0, 2, \ldots, m$ (and not just for m!) **then** it holds for $n = m + 1$.

Can also be used to prove a property for all integers greater than or equal to some particular natural number b

Example: Proof by strong induction

Every natural number $n \geq 2$, is a prime or a product of primes.

Base Case: Take $n = 2$. Then n is a prime number.

Inductive Step: Assume that the property holds for $n = m$ so *every* number i s.t. $2 \leq i \leq m$ is a prime or a produce of primes. Now consider $n = m + 1$.

Example: Number of multiplications

For any integer $n \geq 1$, if $x_1, x_2, \ldots, x_n$ are n numbers, then no matter how the parentheses are inserted into their product, the number of multiplications used to compute the product is $n - 1$.

Bad proofs: Arguing from example

An incorrect "proof" of the fact that the sum of any two even integers is even.

> *This is true because if $m = 14$ and $n = 6$, which are both even, then $m + n = 20$, which is also even.*

Bad proofs: Using the same letter to mean two different things

Consider the following "proof" fragment:

> *Suppose m and n are any odd integers. Then by definition of odd, $m = 2k + 1$ and $n = 2k + 1$ for some integer k.*

Bad proofs: Jumping to a conclusion

To jump to a conclusion means to allege the truth of something without giving an adequate reason.

> *Suppose m and n are any even integers. By definition of even, $m = 2r$ and $n = 2s$ for some integers r and s. Then $m + n = 2r + 2s$. So $m + n$ is even.*

Bad proofs: Circular reasoning

To engage in circular reasoning means to assume what is to be proved.

> *Suppose m and n are any odd integers. When any odd integers are multiplied, their product is odd. Hence mn is odd.*

Bad proofs: Confusion between what is known and what is still to be shown

Suppose m and n are any odd integers. We must show that mn is odd. This means that there exists an integer s such that

$$mn = 2s + 1.$$

Also by definition of odd, there exist integers a and b such that

$$m = 2a + 1 \text{ and } n = 2b + 1.$$

Then

$$mn = (2a + 1)(2b + 1) = 2s + 1.$$

So, since s is an integer, mn is odd by definition of odd.

Good proofs in practice[2]

State your game plan.

> *A good proof begins by explaining the general line of reasoning, for example, "We use case analysis" or "We argue by contradiction."*

[2] *Mathematics for Computer Science* by E. Lehman, F. T. Leighton, and A. R. Meyer.

Good proofs in Practice

Keep a linear flow.

> *Sometimes proofs are written like mathematical mosaics, with juicy titbits of independent reasoning sprinkled throughout. This is not good. The steps of an argument should follow one another in an intelligible order.*

A proof is an essay, not a calculation.

> *Many students initially write proofs the way they compute integrals. The result is a long sequence of expressions without explanation, making it very hard to follow. This is bad. A good proof usually looks like an essay with some equations thrown in. Use complete sentences.*

Structure your proof

- Theorem—A very important true statement.
- Proposition—A less important but still interesting statement.
- Lemma—A true statement used to prove other statements.
- Corollary—A simple consequence of a theorem or a proposition.

Finish

> *At some point in a proof, you'll have established all the essential facts you need. Resist the temptation to quit and leave the reader to draw the "obvious" conclusion. Instead, tie everything together yourself and explain why the original claim follows.*

3.36pt

Foundations of Computer Science Comp109

University of Liverpool
Boris Konev
konev@liverpool.ac.uk
http://www.csc.liv.ac.uk/~konev/COMP109

Part 2. (Naive) Set Theory

Comp109 Foundations of Computer Science

Reading

- K. H. Rosen. *Discrete Mathematics and Its Applications* Chapter 2

Contents

- Notation for *sets*.
- Important sets.
- What is a *subset* of a set?
- When are two sets *equal*?
- *Operations* on sets.
- *Algebra* of sets.
- Bit strings.
- *Cardinality* of sets.
- Russell's paradox.

Notation

A *set* is a collection of objects, called the *elements* of the set. For example:

- $\{7, 5, 3\}$;
- $\{\text{Liverpool}, \text{Manchester}, \text{Leeds}\}$.

We have written down the elements of each set and contained them between the *braces* $\{\ \}$.

We write $a \in S$ to denote that the object a is an element of the set S:

$$7 \in \{7, 5, 3\}, \quad 4 \notin \{7, 5, 3\}.$$

Notes

- The order of elements does not matter
- Repeatitions do not count

Notation

For a large set, especially an infinite set, we cannot write down all the elements. We use a predicate P instead.

$$S = \{x \mid P(x)\}$$

denotes the set of objects x for which the predicate $P(x)$ is true.

Examples: Let $S = \{1, 3, 5, 7, \dots\}$. Then

$$S = \{x \mid x \text{ is an odd positive integer}\}$$

and

$$S = \{2n - 1 \mid n \text{ is a positive integer }\}.$$

More examples

Find simpler descriptions of the following sets by listing their elements:

- $A = \{x \mid x \text{ is an integer and } x^2 + 4x = 12\}$;
- $B = \{x \mid x \text{ a day of the week not containing "}u\text{" }\}$;
- $C = \{n^2 \mid n \text{ is an integer }\}$.

Important sets (notation)

The empty set has no elements. It is written as $\emptyset$ or as $\{\}$.

We have seen some other examples of sets in Part 1.

- $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ (the natural numbers)
- $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ (the integers)
- $\mathbb{Z}^+ = \{1, 2, 3, \dots\}$ (the positive integers)
- $\mathbb{Q} = \{x/y \mid x \in \mathbb{Z}, y \in \mathbb{Z}, y \neq 0\}$ (the rationals)
- $\mathbb{R}$: (real numbers)
  - $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$ the set of real numbers between a and b (inclusive)

Detour: Sets in python

Sets are the 'most elementary' data structures (though they don't always map well into the underlying hardware).

Some modern programming languages feature sets.

- For example, in Python one writes

```
empty = set()
m = {'a', 'b', 'c'}
n = {1, 2}
print 'a' in m
```


Computer representation of sets

Only finite sets can be represented

- Number of elements not fixed: List (?) Java&Python do differently
- All elements of A are drawn from some ordered sequence $S = s_1, \dots, s_n$: the characteristic vector of A is the sequence $(b_1, \dots, b_n)$ where

$$b_i = \begin{cases} 1 & \text{if} \quad s_i \in A \\ 0 & \text{if} \quad s_i \notin A \end{cases}$$

Sequences of zeros and ones of length n are called *bit strings* of length n. AKA *bit vectors* AKA *bit arrays*

Example

Let $S = \{1, 2, 3, 4, 5\}$, $A = \{1, 3, 5\}$ and $B = \{3, 4\}$.

- The characteristic vector of A is $(1, 0, 1, 0, 1)$.
- The characteristic vector of B is $(0, 0, 1, 1, 0)$.
- The set characterised by $(1, 1, 1, 0, 1)$ is $\{1, 2, 3, 5\}$.
- The set characterised by $(1, 1, 1, 1, 1)$ is $\{1, 2, 3, 4, 5\}$.
- The set characterised by $(0, 0, 0, 0, 0)$ is ...

Subsets

Definition A set B is called a *subset* of a set A if every element of B is an element of A. This is denoted by $B \subseteq A$.

Examples:

$$\{3, 4, 5\} \subseteq \{1, 5, 4, 2, 1, 3\}, \ \{3, 3, 5\} \subseteq \{3, 5\}, \ \{5, 3\} \subseteq \{3, 5\}.$$

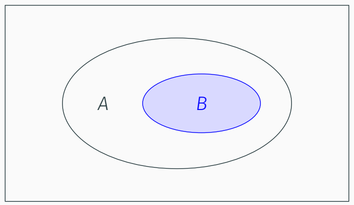


Figure 1: Venn diagram of $B \subseteq A$.

Detour: Subsets in Python

```
def isSubset(A, B):
    for x in A:
        if x not in B:
            return False
    return True
```

Testing the method:

```
print isSubset(n,m)
```

But then there is a built-in operation:

```
print n<m
```

Subsets and bit vectors

Let $S = \{1, 2, 3, 4, 5\}$, $A = \{1, 3, 5\}$ and $B = \{3, 4\}$.

- Is $A \subseteq B$?

- Is the set C, represented by $(1, 0, 0, 0, 1)$, a subset of the set D, represented by $(1, 1, 0, 0, 1)$?

Equality

Definition A set A is called *equal* to a set B if $A \subseteq B$ and $B \subseteq A$. This is denoted by $A = B$.

Examples:

$$\{1\} = \{1, 1, 1\},$$

$$\{1, 2\} = \{2, 1\},$$

$$\{5, 4, 4, 3, 5\} = \{3, 4, 5\}.$$

The union of two sets

Definition The union of two sets A and B is the set

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}.$$

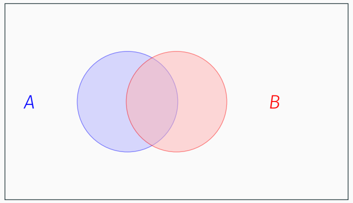


Figure 2: Venn diagram of $A \cup B$.

Example

Suppose

$$A = \{4, 7, 8\}$$

and

$$B = \{4, 9, 10\}.$$

Then

$$A \cup B = \{4, 7, 8, 9, 10\}.$$

Detour: Set union in Python

```
def union(A, B):
    result = set()
    for x in A:
        result.add(x)
    for x in B:
        result.add(x)
    return result
```

Testing the method:

```
print union(m, n)
```

But then there is a built-in operation:

```
print m.union(n)
```


Union of sets represented by bit vectors

Let $S = \{1, 2, 3, 4, 5\}$, $A = \{1, 3, 5\}$ and $B = \{3, 4\}$.

- Compute $A \cup B$.

- Compute the union of the set C, represented by $(1, 0, 0, 0, 1)$, and the set D, represented by $(1, 1, 0, 0, 1)$.

The intersection of two sets

Definition The intersection of two sets A and B is the set

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}.$$

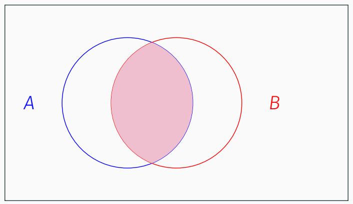


Figure 3: Venn diagram of $A \cap B$.

Example

Suppose

$$A = \{4, 7, 8\}$$

and

$$B = \{4, 9, 10\}.$$

Then

$$A \cap B = \{4\}$$

Detour: Set intersection in Python

```
def intersection(A, B):
    result = set()
    for x in A:
        if x in B:
            result.add(x)
    return result
```

Testing the method:

```
print intersection(m, n)
print intersection(n, {1})
```

But then there is a built-in operation:

```
print n.intersection({1})
```

Intersection of sets represented by bit vectors

Let $S = \{1, 2, 3, 4, 5\}$, $A = \{1, 3, 5\}$ and $B = \{3, 4\}$.

- Compute $A \cap B$.
- Compute the intersection of the set C, represented by $(1, 0, 0, 0, 1)$, and the set D, represented by $(1, 1, 0, 0, 1)$.

The relative complement

Definition The relative complement of a set B relative to a set A is the set

$$A - B = \{x \mid x \in A \text{ and } x \notin B\}.$$

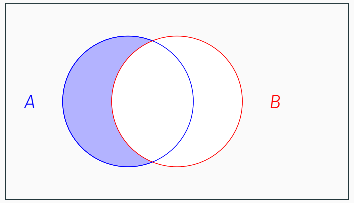


Figure 4: Venn diagram of $A - B$.

Example

Suppose

$$A = \{4, 7, 8\}$$

and

$$B = \{4, 9, 10\}.$$

Then

$$A - B = \{7, 8\}$$

Detour: Set complement in Python

```
def complement(A, B):
    result = set()
    for x in A:
        if x not in B:
            result.add(x)
    return result
```

Testing the method:

```
print complement(m, {'a'})
```

But then there is a built-in operation:

```
print m-{'a'}
```


Relative complement and bit vectors

Let $S = \{1, 2, 3, 4, 5\}$, $A = \{1, 3, 5\}$ and $B = \{3, 4\}$.

- Compute $A - B$.
- Compute the relative complement of the set C, represented by $(1, 0, 0, 0, 1)$, related to the set D, represented by $(1, 1, 0, 0, 1)$.

The complement

When we are dealing with subsets of some large set U, then we call U the *universal set* for the problem in question.

Definition The complement of a set A is the set

$$\sim A = \{x \mid x \notin A\} = U - A.$$

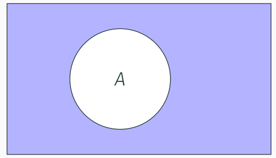


Figure 5: Venn diagram of $\sim A$. (The rectangle is U)

Complement and bit vectors

Let $S = \{1, 2, 3, 4, 5\}$, $A = \{1, 3, 5\}$ and $B = \{3, 4\}$.

- Compute $\sim A$.
- Compute $\sim B$.
- Compute the complement of the set C, represented by $(1, 0, 0, 0, 1)$.

The symmetric difference

Definition The symmetric difference of two sets A and B is the set

$$A \Delta B = \{x \mid (x \in A \text{ and } x \notin B) \text{ or } (x \notin A \text{ and } x \in B)\}.$$

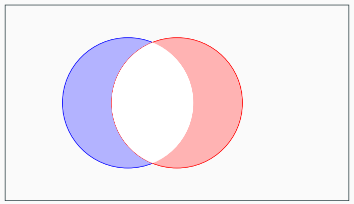


Figure 6: Venn diagram of $A \Delta B$.

Example

Suppose

$$A = \{4, 7, 8\}$$

and

$$B = \{4, 9, 10\}.$$

Then

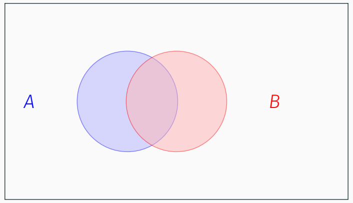
$$A \Delta B = \{7, 8, 9, 10\}$$

The algebra of sets

Suppose that A, B and U are sets with $A \subseteq U$ and $B \subseteq U$.

Commutative laws:

$$A \cup B = B \cup A, \quad A \cap B = B \cap A;$$



Proving the commutative law $A \cup B = B \cup A$

Definition: $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$ $B \cup A = \{x \mid x \in B \text{ or } x \in A\}$.

These are the same set. To see this, check all possible cases.

Case 1: Suppose $x \in A$ and $x \in B$. Since $x \in A$, the definitions above show that x is in both $A \cup B$ and $B \cup A$.

Case 2: Suppose $x \in A$ and $x \notin B$. Since $x \in A$, the definitions above show that x is in both $A \cup B$ and $B \cup A$.

Case 3: Suppose $x \notin A$ and $x \in B$. Since $x \in B$, the definitions above show that x is in both $A \cup B$ and $B \cup A$.

Case 4: Suppose $x \notin A$ and $x \notin B$. The definitions above show that x is not in $A \cup B$ and x is not in $B \cup A$.

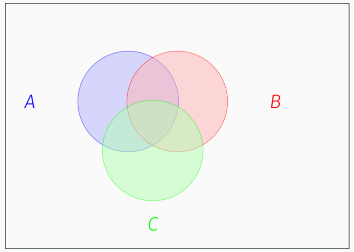
So, for all possible x, either x is in both $A \cup B$ and $B \cup A$, or it is in neither. We conclude that the sets $A \cup B$ and $B \cup A$ are the same.

The algebra of sets

Suppose that A, B, C, U are sets with $A \subseteq U$, $B \subseteq U$, and $C \subseteq U$.

Associative laws:

$$A \cup (B \cup C) = (A \cup B) \cup C, \quad A \cap (B \cap C) = (A \cap B) \cap C;$$



Proving the associative law $A \cup (B \cup C) = (A \cup B) \cup C$

This is almost as easy as proving the commutative law, but now there are 8 cases to check, depending on whether $x \in A$, whether $x \in B$ and whether $x \in C$.

Definition: $X \cup Y = \{x \mid x \in X \text{ or } x \in Y\}$

Here is one case: Suppose $x \in A$, $x \notin B$ and $x \notin C$. Since $x \in A$, we can use the definition with $X = A$ and $Y = B \cup C$ to show that $x \in A \cup (B \cup C)$.

Since $x \in A$, we can use the definition with $X = A$ and $Y = B$ to show that $x \in A \cup B$. Then we can use the definition with $X = A \cup B$ and $Y = C$ to show that $x \in (A \cup B) \cup C$.

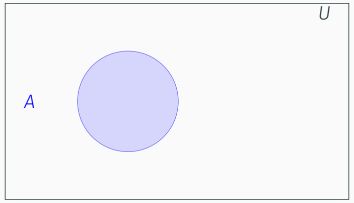
Writing out all eight cases is tedious, but it is not difficult.

The algebra of sets

Suppose that A and U are sets with $A \subseteq U$.

Identity laws:

$$A \cup \emptyset = A, \; A \cup U = U, \; A \cap U = A, \; A \cap \emptyset = \emptyset;$$

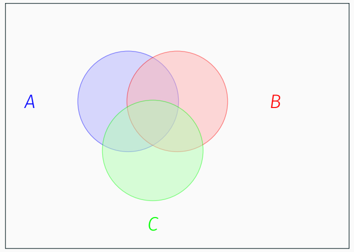


The algebra of sets

Suppose that A, B, C, U are sets with $A \subseteq U$, $B \subseteq U$, and $C \subseteq U$.

Distributive laws:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C), \; A \cup (B \cap C) = (A \cup B) \cap (A \cup C);$$

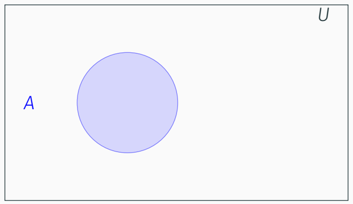


The algebra of sets

Suppose that A and U are sets with $A \subseteq U$. Let $\sim A = U - A$. Then

Complement laws:

$$A \cup \sim A = U, \; \sim U = \emptyset, \; \sim(\sim A) = A, A \cap \sim A = \emptyset, \; \sim \emptyset = U;$$

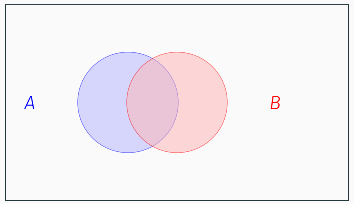


The algebra of sets

Suppose that A, B and U are sets with $A \subseteq U$, and $B \subseteq U$. Recall that $\sim X = U - X$ and $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$ and $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$. Then

De Morgan's laws:

$$\sim(A \cup B) = \sim A \cap \sim B, \; \sim(A \cap B) = \sim A \cup \sim B.$$



A proof of De Morgan's law $\sim(A \cap B) = \sim A \cup \sim B$

Case 1: Suppose $x \in A$ and $x \in B$. From the definition of $\cap$, $x \in A \cap B$. So from the definition of $\sim$, $x \notin \sim(A \cap B)$. From the definition of $\sim$, $x \notin \sim A$ and also $x \notin \sim B$. So from the definition of $\cup$, $x \notin \sim A \cup \sim B$.

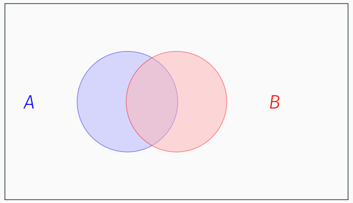
Case 2: Suppose $x \in A$ and $x \notin B$. From the definition of $\cap$, $x \notin A \cap B$. So from the definition of $\sim$, $x \in \sim(A \cap B)$. From the definition of $\sim$, $x \notin \sim A$ but $x \in \sim B$. So from the definition of $\cup$, $x \in \sim A \cup \sim B$.

Case 3: Suppose $x \notin A$ and $x \in B$. From the definition of $\cap$, $x \notin A \cap B$. So from the definition of $\sim$, $x \in \sim(A \cap B)$. From the definition of $\sim$, $x \in \sim A$ but $x \notin \sim B$. So from the definition of $\cup$, $x \in \sim A \cup \sim B$.

Case 4: Suppose $x \notin A$ and $x \notin B$. From the definition of $\cap$, $x \notin A \cap B$. So from the definition of $\sim$, $x \in \sim(A \cap B)$. From the definition of $\sim$, $x \in \sim A$ and $x \in \sim B$. So from the definition of $\cup$, $x \in \sim A \cup \sim B$.

Using the algebra of sets

Prove that $A \Delta B = (A \cup B) \cap \sim (A \cap B)$. (See the next slide.)



$$
\begin{aligned}
(A \cup B) \cap \sim (A \cap B) &= (A \cup B) \cap (\sim A \cup \sim B) \text{ De Morgan} \\
&= ((A \cup B) \cap \sim A) \cup ((A \cup B) \cap \sim B) \text{ distributive} \\
&= (\sim A \cap (A \cup B)) \cup (\sim B \cap (A \cup B)) \text{ commutative} \\
&= ((\sim A \cap A) \cup (\sim A \cap B)) \cup ((\sim B \cap A) \cup (\sim B \cap B)) \text{ distributive} \\
&= ((A \cap \sim A) \cup (B \cap \sim A)) \cup ((A \cap \sim B) \cup (B \cap \sim B)) \text{ commutative} \\
&= (\emptyset \cup (B \cap \sim A)) \cup ((A \cap \sim B) \cup \emptyset) \text{ complement} \\
&= (A \cap \sim B) \cup (B \cap \sim A) \text{ commutative and identity} \\
&= A \Delta B \text{ definition}
\end{aligned}
$$

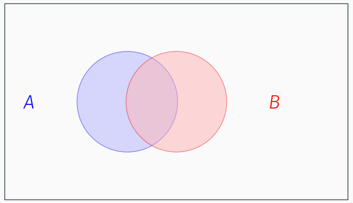
Cardinality of sets

Definition The cardinality of a *finite* set S is the number of elements in S, and is denoted by $|S|$.

Computing the cardinality of a union of two sets

If A and B are sets then

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

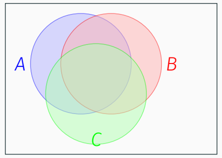


Example

Suppose there are 100 third-year students. 40 of them take the module "Sequential Algorithms" and 80 of them take the module "Multi-Agent Systems". 25 of them took both modules. How many students took neither modules?

Computing the cardinality of a union of three sets

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

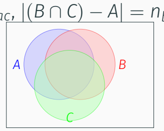


These are special cases of the principle of inclusion and exclusion which we will study later.

Proof (optional)

We need lots of notation.

- $|A - (B \cup C)| = n_a$, $|B - (A \cup C)| = n_b$, $|C - (A \cup B)| = n_c$,
- $|(A \cap B) - C| = n_{ab}$, $|(A \cap C) - B| = n_{ac}$, $|(B \cap C) - A| = n_{bc}$,
- $|A \cap B \cap C| = n_{abc}$.



Then

$$
\begin{aligned}
|A \cup B \cup C| &= n_a + n_b + n_c + n_{ab} + n_{ac} + n_{bc} + n_{abc} \\
&= (n_a + n_{ab} + n_{ac} + n_{abc}) + (n_b + n_{ab} + n_{bc} + n_{abc}) \\
&+ (n_c + n_{ac} + n_{bc} + n_{abc}) - (n_{ab} + n_{abc}) \\
&- (n_{ac} + n_{abc}) - (n_{bc} + n_{abc}) + n_{abc}
\end{aligned}
$$

Reflection

The following statements hold:

- $\emptyset \in \{\emptyset\}$ but $\emptyset \notin \emptyset$;
- $\emptyset \subseteq \{5\}$;
- $\{2\} \not\subseteq \{\{2\}\}$ but $\{2\} \in \{\{2\}\}$;
- $\{3, \{3\}\} \neq \{3\}$.

Why is this set theory "naive"

It suffers from paradoxes.

Why is this set theory "naive"

It suffers from paradoxes.

A leading example:

> *A barber is the man who shaves all those, and only those, men who do not shave themselves.*

- Who shaves the barber?

Russell's Paradox

Russell's paradox shows that the 'object' $\{x \mid P(x)\}$ is not always meaningful.

Set $A = \{B \mid B \notin B\}$

Problem: do we have $A \in A$?

Abbreviate, for any set C, by $P(C)$ the statement $C \notin C$. Then $A = \{B \mid P(B)\}$.

- If $A \in A$, then (from the definition of P), not $P(A)$. Therefore $A \notin A$.
- If $A \notin A$, then (from the definition of P), $P(A)$. Therefore $A \in A$.

Foundations of Computer Science Comp109

University of Liverpool
Boris Konev
konev@liverpool.ac.uk
http://www.csc.liv.ac.uk/~konev/COMP109

Part 4. Function

Comp109 Foundations of Computer Science

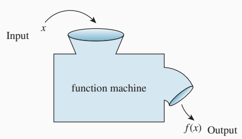
Reading

- Discrete Mathematics and Its Applications K. Rosen, Section 2.3.
- Discrete Mathematics with Applications S. Epp, Chapter 7.

Contents

- Functions: definitions and examples
- Domain, codomain, and range
- Injective, surjective, and bijective functions
- Invertible functions
- Compositions of functions
- Functions and cardinality
- Pigeon hole principle
- Cardinality of infinite sets

Functions



Examples:

- $y = x^2$
- $y = \sin(x)$
- first letter of your name

Functions/methods on programming

Java

```
public int f(int x) {
    return x+5;
}
```

C/C++

```
int f(int x) {
    return x+5;
}
```

Python

```
def f(int x):
    return x+5
```


Definition

A **function** from a set A to a set B is an assignment of exactly one element of B to each element of A.

We write $f(a) = b$ if b is the unique element of B assigned by the function f to the element of a.

If f is a function from A to B we write $f : A \to B$.

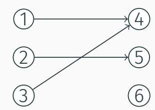


Figure 1: A function $f : \{1, 2, 3\} \to \{4, 5, 6\}$

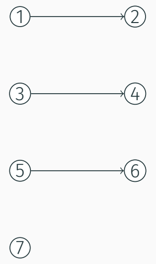


Figure 2: No function

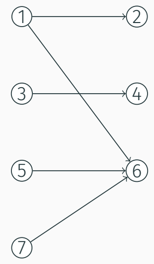


Figure 3: No function

Domain, codomain, and range

Suppose $f : A \to B$.

- A is called the *domain* of f. B is called the *codomain* of f.
- The *range* $f(A)$ of f is
$$f(A) = \{f(x) \mid x \in A\}.$$

Codomain vs range

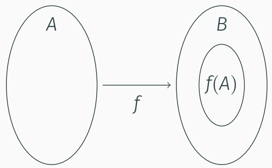
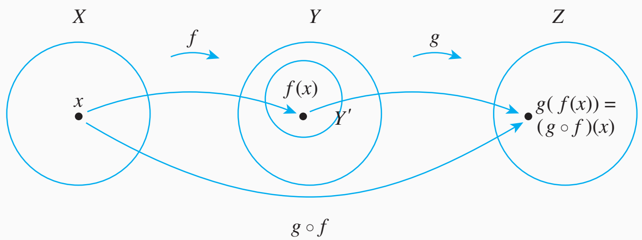


Figure 4: the range of f

Composition of functions

If $f : X \to Y$ and $g : Y \to Z$ are functions, then their **composition** $g \circ f$ is a function from X to Z given by
$$(g \circ f)(x) = g(f(x)).$$



Example

Consider the function $f : \mathbb{R} \to \mathbb{R}$ given by $f(x) = x^2$ and the function $g : \mathbb{R} \to \mathbb{R}$ given by $g(x) = 4x + 3$. Calculate $g \circ f$, $f \circ g$, $f \circ f$ and $g \circ g$.

Injective (one-to-one) functions

Definition Let $f : A \to B$ be a function. We call f an *injective* (or *one-to-one*) function if
$$f(a_1) = f(a_2) \Rightarrow a_1 = a_2 \text{ for all } a_1, a_2 \in A.$$

This is logically equivalent to $a_1 \neq a_2 \Rightarrow f(a_1) \neq f(a_2)$ and so injective functions never repeat values. In other words, different inputs give different outputs.

Examples

$f : \mathbb{Z} \to \mathbb{Z}$ given by $f(x) = x^2$ is not injective.

$h : \mathbb{Z} \to \mathbb{Z}$ given by $h(x) = 2x$ is injective.

Surjective (or onto) functions

Definition $f : A \to B$ is *surjective* (or onto) if the range of f coincides with the codomain of f. This means that for every $b \in B$ there exists $a \in A$ with $b = f(a)$.

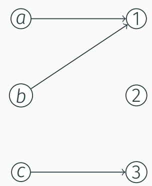
Examples

$f : \mathbb{Z} \to \mathbb{Z}$ given by $f(x) = x^2$ is not surjective.

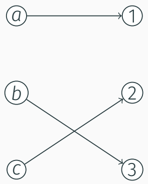
$h : \mathbb{Z} \to \mathbb{Z}$ given by $h(x) = 2x$ is not surjective.

$h' : \mathbb{Q} \to \mathbb{Q}$ given by $h'(x) = 2x$ is surjective.

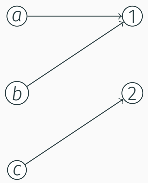
Classify $f : \{a, b, c\} \to \{1, 2, 3\}$ given by



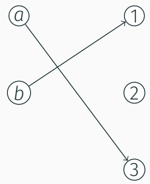
Classify $g : \{a, b, c\} \to \{1, 2, 3\}$ given by



Classify $h : \{a, b, c\} \to \{1, 2\}$ given by



Classify $h' : \{a, b, c\} \to \{1, 2, 3\}$ given by



Bijections

We call f *bijective* if f is both injective and surjective.

Examples

$f : \mathbb{Q} \to \mathbb{Q}$ given by $f(x) = 2x$ is bijective.

Inverse functions

If f is a bijection from a set X to a set Y, then there is a function f^{-1} from Y to X that "undoes" the action of f; that is, it sends each element of Y back to the element of X that it came from. This function is called the inverse function for f.

Then $f(a) = b$ if, and only if, $f^{-1}(b) = a$.

Example

$k : \mathbb{R} \to \mathbb{R}$ given by $k(x) = 4x + 3$ is invertible and

$$k^{-1}(y) = \frac{1}{4}(y - 3).$$

Example

Let $A = \{x \mid x \in \mathbb{R}, x \neq 1\}$ and $f : A \to A$ be given by

$$f(x) = \frac{x}{x - 1}.$$

Show that f is bijective and determine the inverse function.

Bijections and representations

Let $S = \{1, 2, \dots, n\}$ and let B^n be the set of bit strings of length n. The function

$$f : Pow(S) \to B^n$$

which assigns each subset A of S to its characteristic vector is a bijection.

Cardinality of finite sets and functions

Recall: *The cardinality of a finite set S is the number of elements in S*

A bijection $f : S \to \{1, \dots, n\}$.

For finite sets A and B

- $|A| \geq |B|$ iff there is a surjective function from A to B.
- $|A| \leq |B|$ iff there is a injective function from A to B.
- $|A| = |B|$ iff there is a bijection from A to B.

The pigeonhole principle

Let $f : A \to B$ be a function where A and B are finite sets.

The *pigeonhole principle* states that if $|A| > |B|$ then at least one value of f occurs more than once.

In other words, we have $f(a) = f(b)$ for some distinct elements a, b of A.

Pigeons and pigeonholes

If (N+1) pigeons occupy N holes, then some hole must have at least 2 pigeons.

Example

Problem. There are 15 people on a bus. Show that at least two of them have a birthday in the same month of the year.

Example

Problem. How many different surnames must appear in a telephone directory to guarantee that at least two of the surnames begin with the same letter of the alphabet and end with the same letter of the alphabet?

Example

Problem. Five numbers are selected from the numbers $1, 2, 3, 4, 5, 6, 7$ and 8. Show that there will always be two of the numbers that sum to 9.

Extended pigeonhole principle

Consider a function $f : A \to B$ where A and B are finite sets and $|A| > k|B|$ for some natural number k. Then, there is a value of f which occurs at least $k + 1$ times.

Example

Problem. How many different surnames must appear in a telephone directory to guarantee that at least five of the surnames begin with the same letter of the alphabet and end with the same letter of the alphabet?

Example

Problem. Show that in any group of six people there are either three who all know each other or three complete strangers.

Bijections and cardinality

Recall that the cardinality of a finite set is the number of elements in the set.

Sets A and B have the same cardinality iff there is a bijection from A to B.

Example: The cardinality of the power set.

Definition The power set $Pow(A)$ of a set A is the set of all subsets of A. In other words,

$$Pow(A) = \{C \mid C \subseteq A\}.$$

For all $n \in \mathbb{Z}^+$ and all sets A: if $|A| = n$, then $|Pow(A)| = 2^n$.

Power set and bit vectors

Recall that if all elements of a set A are drawn from some ordered sequence $S = s_1, \dots, s_n$: the characteristic vector of A is the sequence $(b_1, \dots, b_n)$ where

$$b_i = \begin{cases} 1 & \text{if} \quad s_i \in A \\ 0 & \text{if} \quad s_i \notin A \end{cases}$$

We use the correspondence between bit vectors and subsets: $|Pow(A)|$ is the number of bit vectors of length n.

The number of n-bit vectors is 2^n

We prove the statement by induction.

Base Case: Take $n = 1$. There are two bit vectors of length 1: (0) and (1).

The number of n-bit vectors is 2^n

Inductive Step: Assume that the property holds for $n = m$, so the number of m-bit vectors is 2^m. Now consider the set B of all $(m+1)$-bit vectors. We must show that $|B| = 2^{m+1}$.

Every $(b_1, b_2, \dots, b_{m+1}) \in B$ starts with an m-bit vector $(b_1, b_2, \dots, b_m)$ followed by b_{m+1}, which can be either 0 or 1.

Thus

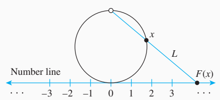
$$|B| = 2^m + 2^m = 2^{m+1}.$$

Infinite sets

Sets A and B have the same cardinality iff there is a bijection from A to B.

Examples:

- $\mathbb{Z}$ and even integers
  - consider $f(n) = 2n$
- $\{x \in \mathbb{R} \mid 0 < x < 1\}$ and $\mathbb{R}^+$
  - consider $g(x) = \frac{1}{x} - 1$
- $\{x \in \mathbb{R} \mid 0 < x < 1\}$ and $\mathbb{R}$



Countable sets

A set that is either finite or has the same cardinality as $\mathbb{N}$ is called **countable**.

- $\mathbb{Z}$

$$\cdots \quad -4 \quad -3 \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad \cdots$$

Countable Sets: $\mathbb{Q}$

$$\begin{array}{ccccccc}
\frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \cdots \\
\frac{2}{1} & \frac{2}{2} & \frac{2}{3} & \frac{2}{4} & \frac{2}{5} & \frac{2}{6} & \cdots \\
\frac{3}{1} & \frac{3}{2} & \frac{3}{3} & \frac{3}{4} & \frac{3}{5} & \frac{3}{6} & \cdots \\
\frac{4}{1} & \frac{4}{2} & \frac{4}{3} & \frac{4}{4} & \frac{4}{5} & \frac{4}{6} & \cdots \\
\frac{5}{1} & \frac{5}{2} & \frac{5}{3} & \frac{5}{4} & \frac{5}{5} & \frac{5}{6} & \cdots \\
\frac{6}{1} & \frac{6}{2} & \frac{6}{3} & \frac{6}{4} & \frac{6}{5} & \frac{6}{6} & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \ddots
\end{array}$$

Uncountable sets

- A set that is not countable is called **uncountable**.
  - $S = \{x \in \mathbb{R} \mid 0 < x < 1\}$ is uncountable

Cantor's diagonal argument

Suppose S is countable. Then the decimal representations of these numbers can be written as a list

$$\begin{aligned}
a_1 &= 0.a_{11}\ a_{12}\ a_{13} \dots\ a_{1n} \dots \\
a_2 &= 0.a_{21}\ a_{22}\ a_{23} \dots\ a_{2n} \dots \\
a_3 &= 0.a_{31}\ a_{32}\ a_{33} \dots\ a_{3n} \dots \\
&\ \vdots \\
a_n &= 0.a_{n1}\ a_{n2}\ a_{n3} \dots\ a_{nn} \dots \\
&\ \vdots
\end{aligned}$$

Let $d = 0.d_1\ d_2\ d_3 \dots d_n \dots$ where

$$d_i = \begin{cases} 1, \text{ if } a_{ii} \neq 1 \\ 2, \text{ if } a_{ii} = 1 \end{cases}$$

Then d is not in the sequence a_1, a_2, a_3...

Foundations of Computer Science Comp109

University of Liverpool
Boris Konev
konev@liverpool.ac.uk
http://www.csc.liv.ac.uk/~konev/COMP109

Part 3. Relations

Comp109 Foundations of Computer Science

Reading

Discrete Mathematics and Its Applications K. Rosen, Chapter 9.

Contents

- The Cartesian product
- Definition and examples
- Representation of binary relations by directed graphs
- Representation of binary relations by matrices
- Properties of binary relations
- Transitive closure
- Equivalence relations and partitions
- Partial orders and total orders.
- Unary relations

Motivation

- Intuitively, there is a "relation" between two things if there is some connection between them.
  E.g.
  - 'friend of'
  - $a < b$
  - m divides n
- Relations are used in crucial ways in many branches of mathematics
  - Equivalence
  - Ordering
- Computer Science

Databases and relations

A database table ≈ relation

TABLE 1 Students.

Student_name	*ID_number*	*Major*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.45
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

Ordered pairs

Definition The Cartesian product $A \times B$ of sets A and B is the *set* consisting of all pairs (a, b) with $a \in A$ and $b \in B$, i.e.,

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}.$$

Note that $(a, b) = (c, d)$ if and only if $a = c$ and $b = d$.

Note

- $\{1, 2\} = \{2, 1\}$ but $(1, 2) \neq (2, 1)$.

Example

- Let $A = \{1, 2\}$ and $B = \{a, b, c\}$. Then

$$A \times B = \{(1, a), (2, a), (1, b), (2, b), (1, c), (2, c)\}.$$

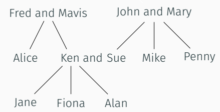
- $B \times A =$

Relations

Definition A binary relation between two sets A and B is a subset R of the Cartesian product $A \times B$.

If $A = B$, then R is called a binary relation on A.

Example: Family tree



Write down

- $R = \{(x, y) \mid \ x \text{ is a grandfather of } y\ \}$;
- $S = \{(x, y) \mid \ x \text{ is a sister of } y\ \}$.

Example 2

Write down the ordered pairs belonging to the following binary relations between $A = \{1, 3, 5, 7\}$ and $B = \{2, 4, 6\}$:

- $U = \{(x, y) \in A \times B \mid x + y = 9\}$;
- $V = \{(x, y) \in A \times B \mid x < y\}$.

Example 3

Let $A = \{1, 2, 3, 4, 5, 6\}$. Write down the ordered pairs belonging to

$$R = \{(x, y) \in A \times A \mid \ x \text{ is a divisor of } y\ \}.$$

Representation of binary relations: directed graphs

- Let A and B be two finite sets and R a binary relation between these two sets (i.e., $R \subseteq A \times B$).
- We represent the elements of these two sets as vertices of a graph.
- For each $(a, b) \in R$, we draw an arrow linking the related elements.
- This is called the directed graph (or digraph) of R.

Example

Consider the relation V between $A = \{1, 3, 5, 7\}$ and $B = \{2, 4, 6\}$ such that $V = \{(x, y) \in A \times B \mid x < y\}$.

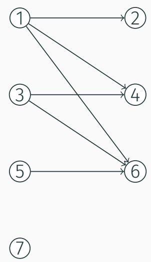


Figure 1: digraph of V

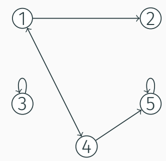
Digraphs of binary relations on a single set

A binary relation between a set A and itself is called “a binary relation on A”.

To represent such a relation, we use a directed graph in which a single set of vertices represents the elements of A and arrows link the related elements.

Consider the relation $V \subseteq A \times A$ where $A = \{1, 2, 3, 4, 5\}$ and

$$V = \{(1, 2), (3, 3), (5, 5), (1, 4), (4, 1), (4, 5)\}.$$



Functions as relations

- Recall that a function f from a set A to a set B assigns exactly one element of B to each element of A.
  - Gives rise to the relation $R_f = \{(a, b) \in A \times B \mid b = f(a)\}$
- If a relation $S \subseteq A \times B$ is such that for every $a \in A$ there exists at most one $b \in B$ with $(a, b) \in S$, relation S is functional.
- (Sometimes in the literature, functions are introduced through functional relations.)

Inverse relation

Definition Given a relation $R \subseteq A \times B$, we define the *inverse relation* $R^{-1} \subseteq B \times A$ by

$$R^{-1} = \{(b, a) \mid (a, b) \in R\}.$$

Example: The inverse of the relation *is a parent of* on the set of people is the relation *is a child of*.

Composition of relations

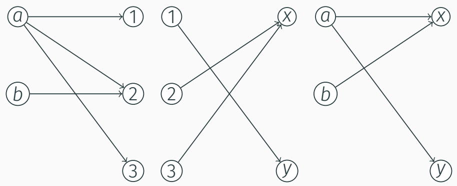
Definition Let $R \subseteq A \times B$ and $S \subseteq B \times C$. The (functional) composition of R and S, denoted by $S \circ R$, is the binary relation between A and C given by

$$S \circ R = \{(a, c) \mid \ \text{exists } b \in B \text{ such that } aRb \text{ and } bSc\}.$$

Example: If R is the relation *is a sister of* and S is the relation *is a parent of*, then

- $S \circ R$ is the relation *is an aunt of*;
- $S \circ S$ is the relation *is a grandparent of*.

Digraph representation of compositions



R $\quad$ S $\quad$ $S \circ R$

Computer friendly representation of binary relations: matrices

- Another way of representing a binary relation between finite sets uses an array.
- Let $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_m\}$ and $R \subseteq A \times B$.
- We represent R by an array M of n rows and m columns. Such an array is called a *n by m matrix.*
- The entry in row i and column j of this matrix is given by $M(i, j)$ where

$$M(i,j) = \begin{cases} T & \text{if} \quad (a_i, b_j) \in R \\ F & \text{if} \quad (a_i, b_j) \notin R \end{cases}$$

Example 1

Let $A = \{1, 3, 5, 7\}$, $B = \{2, 4, 6\}$, and

$$U = \{(x, y) \in A \times B \mid x + y = 9\}$$

Assume an enumeration $a_1 = 1$, $a_2 = 3$, $a_3 = 5$, $a_4 = 7$ and $b_1 = 2$, $b_2 = 4$, $b_3 = 6$. Then M represents U, where

$$M = \begin{bmatrix} F & F & F \\ F & F & T \\ F & T & F \\ T & F & F \end{bmatrix}$$

Example 2

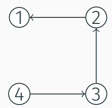
Let $A = \{a, b, c, d\}$ and suppose that $R \subseteq A \times A$ has the following matrix representation:

$$M = \begin{bmatrix} F & T & T & F \\ F & F & T & T \\ F & T & F & F \\ T & T & F & T \end{bmatrix}$$

List the ordered pairs belonging to R.

Example

The binary relation R on $A = \{1, 2, 3, 4\}$ has the following digraph representation.



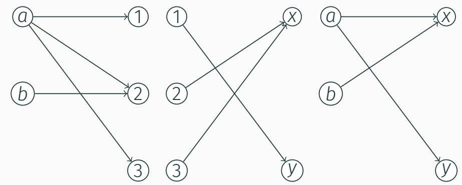
- The ordered pairs $R =$
- The matrix

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

- In words:

Matrices and composition

Now let's go back and see how this works for matrices representing relations



$$R : \begin{bmatrix} T & T & T \\ F & T & F \end{bmatrix} \qquad S : \begin{bmatrix} F & T \\ T & F \\ T & F \end{bmatrix} \qquad S \circ R : \begin{bmatrix} T & T \\ T & F \end{bmatrix}$$

The formal description

Given two matrices with entries "T" and "F" representing the relations we can form the matrix representing the composition. This is called the *logical (Boolean) matrix product*.

Let $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_m\}$ and $C = \{c_1, \ldots, c_p\}$.

The logical matrix M representing R is given by:

$$M(i,j) = \begin{cases} T & \text{if} \quad (a_i, b_j) \in R \\ F & \text{if} \quad (a_i, b_j) \notin R \end{cases}$$

The logical matrix N representing S is given by

$$N(i,j) = \begin{cases} T & \text{if} \quad (b_i, c_j) \in S \\ F & \text{if} \quad (b_i, c_j) \notin S \end{cases}$$

Matrix representation of compositions

Then the entries $P(i, j)$ of the logical matrix P representing $S \circ R$ are given by

- $P(i, j) = T$ if there exists l with $1 \leq l \leq m$ such that $M(i, l) = T$ and $N(l, j) = T$.
- $P(i, j) = F$, otherwise.

We write $P = MN$.

The example from before

Let R be the relation between $A = \{a, b\}$ and $B = \{1, 2, 3\}$ represented by the matrix

$$M = \begin{bmatrix} T & T & T \\ F & T & F \end{bmatrix}$$

Similarly, let S be the relation between B and $C = \{x, y\}$ represented by the matrix

$$N = \begin{bmatrix} F & T \\ T & F \\ T & F \end{bmatrix}$$

Example

Then the matrix $P = MN$ representing $S \circ R$ is

$$P = \begin{bmatrix} T & T \\ T & F \end{bmatrix}$$

Infix notation for binary relations

If R is a binary relation then we write xRy whenever $(x, y) \in R$. The predicate xRy is read as x is R-related to y.

Properties of binary relations (1)

A binary relation R on a set A is

- *reflexive* when xRx for all $x \in A$.

$$\forall x\, A(x) \implies xRx$$

- *symmetric* when xRy implies yRx for all $x, y \in A$;

$$\forall x, y\; xRy \implies yRx$$

Properties of binary relations (2)

A binary relation R on a set A is

- *antisymmetric* when xRy and yRx imply $x = y$ for all $x, y \in A$;

$$\forall x, y\; xRy \text{ and } yRx \implies y = x$$

- *transitive* when xRy and yRz imply xRz for all $x, y, z \in A$.

$$\forall x, y, z\; xRy \text{ and } yRz \implies xRz$$

Example

- *reflexive* xRx
- *symmetric* $xRy \implies yRx$
- *antisymmetric* $xRy, yRx \implies x = y$
- *transitive* $xRy, yRz \implies xRz$

Let $A = \{1, 2, 3\}$.

$R_1 = \{(1, 1), (2, 2), (3, 3), (2, 3), (3, 2)\}$

$R_2 = \{(2, 2), (2, 3), (3, 2), (3, 3)\}$

$R_3 = \{(1, 1), (2, 2), (3, 3), (1, 3)\}$

$R_4 = \{(1, 3), (3, 2), (2, 3)\}$

Digraf representation

In the directed graph representation, R is

- *reflexive* if there is always an arrow from every vertex to itself;
- *symmetric* if whenever there is an arrow from x to y there is also an arrow from y to x;
- *antisymmetric* if whenever there is an arrow from x to y and $x \neq y$, then there is no arrow from y to x;
- *transitive* if whenever there is an arrow from x to y and from y to z there is also an arrow from x to z.

Example

Which of the following define a relation that is reflexive, symmetric, antisymmetric or transitive?

- x divides y on the set $\mathbb{Z}^+$ of positive integers;
- $x \neq y$ on the set $\mathbb{Z}$ of integers;
- x *has the same age as* y on the set of people.

Transitive closure

Given a binary relation R on a set A, the *transitive closure* R^* of R is the (uniquely determined) relation on A with the following properties:

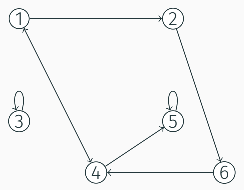
- R^* is transitive;
- $R \subseteq R^*$;
- If S is a transitive relation on A and $R \subseteq S$, then $R^* \subseteq S$.

Example

Let $A = \{1, 2, 3\}$. Find the transitive closure of

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

Finding the transitive closure is easier with the digraph representation



Reachability relation

Transitivity and composition

A relation S is transitive if and only if $S \circ S \subseteq S$.

This is because

$$S \circ S = \{(a, c) \mid \text{ exists } b \text{ such that } aSb \text{ and } bSc\}.$$

Let S be a relation. Set $S^1 = S$, $S^2 = S \circ S$, $S^3 = S \circ S \circ S$, and so on.

Theorem Denote by S^* the transitive closure of S. Then xS^*y if and only if there exists $n > 0$ such that xS^ny.

Transitive closure in matrix form

The relation R on the set $A = \{1, 2, 3, 4, 5\}$ is represented by the matrix

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

Determine the matrix $R \circ R$ and hence explain why R is not transitive.

Computation

$$\begin{bmatrix} T & F & F & T & F \\ F & T & F & F & T \\ F & F & T & F & F \\ T & F & T & F & F \\ F & T & F & T & F \end{bmatrix}\begin{bmatrix} T & F & F & T & F \\ F & T & F & F & T \\ F & F & T & F & F \\ T & F & T & F & F \\ F & T & F & T & F \end{bmatrix} = \begin{bmatrix} T & F & T & T & F \\ F & T & F & T & T \\ F & F & T & F & F \\ T & F & T & T & F \\ T & T & T & F & T \end{bmatrix}$$

$$R \circ R = \{(a, c) \mid \text{ exists } b \in A \text{ such that } aRb \text{ and } bRc\}.$$

Note (in red) that there are pairs (a, c) that are in $R \circ R$ but not in R. Hence, R is not transitive.

Detour: Warshall's algorithm

```
def warshall(a):
    assert (len(row) == len(a) for row in a)
    n = len(a)
    for k in range(n):
        for i in range(n):
            for j in range(n):
                a[i][j] = a[i][j] or
                    (a[i][k] and a[k][j])
    return a

print warshall([[1,0,0,1,0],
                [0,1,0,0,1],
                [0,0,1,0,0],
                [1,0,1,0,0],
                [0,1,0,1,0]])
```


Important relations: Equivalence relations

Definition A binary relation R on a set A is called an *equivalence relation* if it is reflexive, transitive, and symmetric.

Examples:

- the relation R on the non-zero integers given by xRy if $xy > 0$;
- the relation *has the same age* on the set of people.

Definition The *equivalence class* E_x of any $x \in A$ is defined by

$$E_x = \{y \mid yRx\}.$$

Example

Define a relation R on the set $\mathbb{R}$ of real numbers by setting xRy if and only if $x - y$ is an integer. Prove that R is an equivalence relation. Moreover,

- $E_0 = \mathbb{Z}$ is the equivalence class of 0;
- $E_{\frac{1}{2}} = \{\dots, -2\frac{1}{2} - 1\frac{1}{2}, -\frac{1}{2}, \frac{1}{2}, 1\frac{1}{2}, 2\frac{1}{2}, \dots\}$ is the equivalence class of $\frac{1}{2}$.

Functions and equivalence relations

Let $f : A \to B$ be a function. Define a relation R on A by

$$a_1 R a_2 \Leftrightarrow f(a_1) = f(a_2).$$

Then R is an equivalence relation on A. The equivalence class E_a of $a \in A$ is given by

$$E_a = \{a' \in A \mid f(a') = f(a)\}.$$

Example: A is a set of cars, B is the set of real numbers, and f assigns to any car in A its length. Then $a_1 R a_2$ if and only if a_1 and a_2 are of the same length.

Partition of a set

A *partition* of a set A is a collection of non-empty subsets $A_1, \dots, A_n$ of A satisfying:

- $A = A_1 \cup A_2 \cup \dots \cup A_n$;
- $A_i \cap A_j = \emptyset$ for $i \neq j$.

The A_i are called the blocks of the partition.

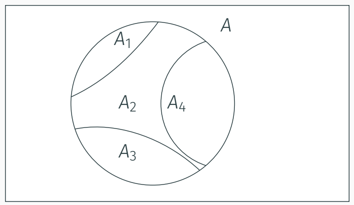


Figure 3: Partition of A

Connecting partitions and equivalence relations

Theorem Let R be an equivalence relation on a non-empty set A. Then the equivalence classes $\{E_x \mid x \in A\}$ form a partition of A.

Proof (Optional)

The proof is in four parts:

(1) We show that the equivalence classes $E_x = \{y \mid yRx\}$, $x \in A$, are non-empty subsets of A: by definition, each E_x is a subset of A. Since R is reflexive, xRx. Therefore $x \in E_x$ and so E_x is non-empty.

(2) We show that A is the union of the equivalence classes $E_x, x \in A$: We know that $E_x \subseteq A$, for all E_x, $x \in A$. Therefore the union of the equivalence classes is a subset of A. Conversely, suppose $x \in A$. Then $x \in E_x$. So, A is a subset of the union of the equivalence classes.

(Optional) Proof (continued)

The purpose of the last two parts is to show that distinct equivalence classes are disjoint, satisfying (ii) in the definition of partition.

(3) We show that if xRy then $E_x = E_y$: Suppose that xRy and let $z \in E_x$. Then, zRx and xRy. Since R is a transitive relation, zRy. Therefore, $z \in E_y$. We have shown that $E_x \subseteq E_y$. An analogous argument shows that $E_y \subseteq E_x$. So, $E_x = E_y$.

(4) We show that any two distinct equivalence classes are disjoint: To this end we show that if two equivalence classes are not disjoint then they are identical. Suppose $E_x \cap E_y \neq \emptyset$. Take a $z \in E_x \cap E_y$. Then, zRx and zRy. Since R is symmetric, xRz and zRy. But then, by transitivity of R, xRy. Therefore, by (3), $E_x = E_y$.

Connecting partitions and equivalence relations

Theorem Suppose that $A_1, \ldots, A_n$ is a partition of A. Define a relation R on A by setting: xRy if and only if there exists i such that $1 \leq i \leq n$ and $x, y \in A_i$. Then R is an equivalence relation.

Proof (Optional)

- Reflexivity: if $x \in A$, then $x \in A_i$ for some i. Therefore xRx.
- Transitivity: if xRy and yRz, then there exists A_i and A_j such that $x, y \in A_i$ and $y, z \in A_j$. $y \in A_i \cap A_j$ implies $i = j$. Therefore $x, z \in A_i$ which implies xRz.
- Symmetry: if xRy, then there exists A_i such that $x, y \in A_i$. Therefore yRx.

Important relations: Partial orders

Definition A binary relation R on a set A which is reflexive, transitive and antisymmetric is called a partial order.

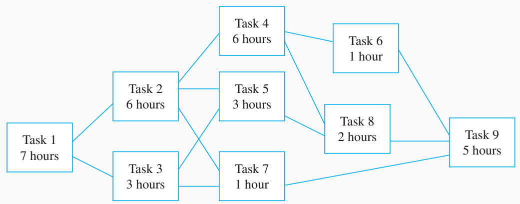
Partial orders are important in situations where we wish to characterise precedence.

Examples:

- the relation $\leq$ on the the set $\mathbb{R}$ of real numbers;
- the relation $\subseteq$ on $Pow(A)$;
- "*is a divisor of*" on the set $\mathbb{Z}^+$ of positive integers.

Example: Job scheduling

Task	Immediately Preceding Tasks
1	
2	1
3	1
4	2
5	2, 3
6	4
7	2, 3
8	4, 5
9	6, 7, 8



Predecessors in partial orders

If R is a partial order on a set A and xRy, $x \neq y$ we call x a predecessor of y.

If x is a predecessor of y and there is no $z \notin \{x, y\}$ for which xRz and zRy, we call x an immediate predecessor of y.

Important relations: Total orders

Definition A binary relation R on a set A is a total order if it is a partial order such that for any $x, y \in A$, xRy or yRx.

The Hasse diagram of a total order is a chain.

Examples

- the relation $\leq$ on the set $\mathbb{R}$ of real numbers;
- the usual lexicographical ordering on the words in a dictionary;
- the relation "is a divisor of" is *not* a total order.

n-ary relations

The Cartesian product $A_1 \times A_2 \times \cdots \times A_n$ of sets $A_1, A_2, \ldots, A_n$ is defined by

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, \ldots, a_n) \mid a_1 \in A_1, \ldots, a_n \in A_n\}.$$

Here $(a_1, \ldots, a_n) = (b_1, \ldots, b_n)$ if and only if $a_i = b_i$ for all $1 \leq i \leq n$.

An n-ary relation is a subset of $A_1 \times \ldots\ldots A_n$

Databases and relations

A database table $\approx$ relation

TABLE 1 Students.

Student_name	*ID_number*	*Major*	*GPA*
Ackermann	231455	Computer Science	3.88
Adams	888323	Physics	3.45
Chou	102147	Computer Science	3.49
Goodfriend	453876	Mathematics	3.45
Rao	678543	Mathematics	3.90
Stevens	786576	Psychology	2.99

Students $= \{$

Unary relations

Unary relations are just subsets of a set.

Example: The unary relation `EvenPositiveIntegers` on the set $\mathbb{Z}^+$ of positive integers is

$$\{x \in \mathbb{Z}^+ \mid x \text{ is even}\}.$$