COMP11120 Mathematical Techniques for Computer Science Part 1

Andrea Schalk
A.Schalk@manchester.ac.uk

Course webpage
studentnet.cs.manchester.ac.uk/ugt/COMP11120

September 20, 2015

Contents

Organizational issues

Structure of the unit

COMP11120 is a 20 credit course unit that is taught over two semesters. You will receive one mark at the end of the unit, and it is that mark which decides whether you pass or fail.

For internal purposes the first semester sometimes appears under the course code COMP11121, and the second as COMP11122, for example in Arcade.

The webpage for this course is at

`www.studentnet.cs.manchester.ac.uk/ugt/COMP11120`.

Electronic copies of these notes are available there. This unit does have a presence in Blackboard[1], and some material will be published there.

When and where

The unit is taught with two lectures and one examples class per week.

Semester 1

Lectures take place

- Mondays 12.00 in the Cordingley Theatre in the Humanities Building (Bridge Street) and
- Tuesdays 12.00 in 1.1 in the Kilburn Building.

For the examples classes you are split into four lab groups. Which group you are in is determined by the letter part of the name of your tutorial groups. Examples classes take place as follows:

- **Group W**: Fridays 14.00 in G102;
- **Group B+X**: Fridays 13.00 in G102;
- **Group Y**: Tuesdays 15.00 in G102.
- **Group Z**: Tuesdays 14.00 in G102;

Examples classes start in Week 2 and run through to Week 12.

[1] This is the University's E-learning environment.

Semester 2

Lectures take place[2]

- Wednesdays 9.00 in 1.1 and
- Thursdays at 10.00 in 1.1.

Examples classes take place

- **Group W**: Thursdays 15.00 in G102;
- **Group B+X**: Tuesdays at 14.00 in G102;
- **Group Y**: Thursdays at 14.00 in G102;
- **Group Z**: Tuesdays at 15.00 in G102.

Who

This course unit is taught by two members of staff:

- Renate Schmidt `Renate.Schmidt@manchester.ac.uk`,
- Andrea Schalk, `A.Schalk@manchester.ac.uk` (course leader).

Assessment

Overview

The assessment of this course unit has the following components:

75% Two 2 hour end of term examinations, one after Semester 1 and one after Semester 2, with

- the Semester 1 exam making up[3] 35% and
- the Semester 2 exam making up 40% of the final mark.

25% Coursework. This is split further into

- 5% Midterm test after Reading Week in Semester 1.
- 10% Assessed exercises for Semester 1. These are checked in the examples classes each week.
- 10% Assessed exercises for Semester 2. These are checked in the examples classes each week.

Examinations

There are written examinations after both, Semester 1 and Semester 2. Note that this course unit has been substantially revised for 2014/15, and will change again for 2015/16. As a consequence previous exam papers do not accurately reflect questions to appear in the exams from 2014/15 onwards. The exams will be based mostly on questions which are similar to the core questions from the examples classes, with a few marks (less than 20%) available for questions beyond that.

There will be three questions in the **Semester 1** exam, one each for the major topics this semester. This means there will be one question each on

[2]There may be changes to the timetable for Semester 2.

[3]Combined with the mid-term test that adds up to 40%, which means that Semester 1 counts the same as Semester 2.

- statements and proofs;
- logic;
- probability.

Material on complex numbers may be included in any question. You will be told about the Semester 2 exam at a later date.

Mid-term test

There is a test around the middle of Semester 1. It prepares you for the examination in January and gives you an idea how well you have understood the material taught so far. The questions will be similar to the exercises in the notes.

Assessed exercises

Note that in the first examples class in Week 1 we will not give marks. We expect you to try to solve the exercises from Chapter 0 given on the relevant sheet.

For each examples classes you are told to prepare a number of exercises (typically five exercises per wee;, see the sheets at the end of the notes). In some cases you have a choice regarding which exercises you want to do. This is to ensure that students who have seen similar material before still have something interesting to do. The exercises typically are about looking at a particular concept or technique. All the parts of some exercise are concerned with the same abstract property, so no matter which ones you do you will familiarize yourself with that property.

There are two kinds of exercises:

- **Core.** These belong to the core of the taught material, and there are typically three such exercises each week.
- **Extensional.** These questions extend the core material.

If you are stuck on one exercise for ten minutes without making progress then *move on to the next one*. Also note that the different parts to each exercise are *not necessarily in rising order of difficulty*.

The point of the examples classes are

- to give you feedback on whether you have understood the material and answered the exercises correctly with the right level of detail, and
- to help you with exercises you had difficulties solving, concentrating on the core exercises.

The rules for assessing exercises are quite simple. The **deadline** is the beginning of each examples class. If you are more than five minutes late you have to have a good reason, which you have to explain to the member of staff present. Otherwise your mark will be 0 for the week.

The **deliverables** are:

- A piece of paper with your solution to the exercise.
- More pieces of paper with all your rough work.[4]

[4]Nobody will look at these in detail and there's no need to be embarrassed about what it looks like.

- You being able to explain your solution *promptly* when asked.

If you cannot explain your work when the teaching assistant[5] asks you to do so, or if you do not have your rough work with you, you will lose some or all of the marks for the given week.

Each exercise is **marked out of 2**, with the following mark allocation:

- 0 marks: You cannot prove that you made a serious attempt to solve the exercise before the start of your examples class.
- 1 mark: You can demonstrate that you made a serious attempt to solve the exercise before the start of the examples class but you are a long way away from the correct solution, or you have not given enough detail. At a *minimum* you should
  - copy the relevant definition(s) from the notes (the Glossary near the end will help you find those),
  - identify similar examples in the notes and
  - be able to describe to the TA where you got stuck.

  If with a little help from the TA you can complete your answer you may be able to move from 1 to 2 marks for that exercise.
- 2 marks: As the previous point, but your solution is largely correct.

You can only get the second mark for an **extensional exercise** if you have solved and completely understood and the **core exercises**. The TAs will ask you whether you have done so. If you say you have, but then cannot answer some questions about the core exercises, the TA will concentrate on those, and not look at your extensional exercises at all. Your maximum mark will then be 6 for that week. If you have questions ready for the core exercises the TA will help with those, and also give you marks for preparing extensional exercises if you have done so. Your maximum mark for that week is 8. If you have completely understood the core exercises and also solved all the extensional ones then your maximum mark for that week is 10.

The TAs have about five minutes per students in each examples class. To get the most out of this it is *vital* that you have your work ready, that you have a list of questions based on what you did not understand, and that you can promptly answer questions asked by the TA. If you are not ready when a TA comes to mark you you will go to the end of the queue, and there may then be less time to answer any questions you have.

It is your job to ask questions, and the TA's job is to answer those as far as possible within the examples classes. It is *not* the job of the TAs to work out what your questions should be based on the solution you show them!

These exercises are assessed work. If you copy somebody else's work you are committing a **plagiarism** offence. If you let somebody copy your work you are also committing plagiarism. This will result in a mark of 0 for the week, and if you commit repeated or particularly serious offences you may have to face School or Faculty panels.

A few days after each examples classes you can check the mark you received in Arcade. Sometimes transcription errors occur. If you find that you have no mark, or the wrong mark, for a particular examples

[5]Teaching assistants used to be called demonstrators in the School. You will still find people referring to demonstrators—it's the same thing. We also refer to teaching assistants as TAs.

class, then please immediately contact the course leader, Andrea Schalk, at `A.Schalk@manchester.ac.uk` or talk to the academic in your next examples class.

Extensions. In the interest of providing further feedback, solutions are published after the last examples class each week. As a consequence it is *not possible* to give extensions for the work to be prepared for each week.

However, we count the **best eight out of the ten possible marks** you obtain each semester. This is to account for the fact that students may have to miss up to two examples classes. If you have good reasons to miss more than two examples classes then contact the course leader, Andrea Schalk.

Also note that you **must attend at least seven out of ten** examples classes for your coursework mark to count.

If you cannot attend your usual examples class in a week, but are free for another (check the information on page 3 to see when and where these take place) then you should attend that examples class. Please make yourself known at the beginning to the member of staff present, or to a teaching assistant. It will then be possible to mark your work in that examples class. However, there **must be** a good reason to attend another examples class, and you may not do so routinely.

The discipline of mathematics

Mathematics is a foundational subject for many sciences and engineering disciplines, and this holds true for computer science as well.

This course unit

In order to master the subject sufficiently to be able to apply it it is not sufficient to treat mathematics as a series of recipes for various calculations.

Instead mathematics formally defines a number of abstract concepts, and then derives their properties (and so justifies calculations of various kinds). As a discipline mathematics is unique in that its theories may be proved or disproved conclusively. It is not necessary to acquire data to study whether the predictions of a theory match observations from experiments—instead one proves statements based on universally accepted derivation rules.

This course unit provides an introduction to those areas of (abstract) mathematics which are of particular importance in computer science. Its aim is to do so rigorously by introducing various concepts formally, and then to show what properties these have. For this purpose it is also necessary to talk about the notion of proof in order to make it clear how valid derivations are formed. Throughout the notes examples from applications within computer science are given wherever possible.

This is the only course unit teaching mathematics in the curriculum for computer science students in Manchester. It therefore has to lay the foundations for everything that is to follow, and in some cases you will not see the material introduced here until the second, or even the third year.

How to learn the material

Abstract mathematics requires a fairly lengthy learning process. Typically material has to be studied more than once to be fully understood. The point of weekly assessed exercises is to encourage students to try their best to understand various abstract concepts and their uses. In the examples

classes you get feedback regarding your understanding, and the revision period should provide another opportunity to improve your understanding before the exams.

Mathematics isn't a spectator sport—the only way to learn it is to do it yourself. Watching others carry out calculations or proofs may be instructive at first, but in the end the only way of properly understanding it is to make your own brain work through the problems given.

You should think of this kind of learning as happening on a spiral: Every time you revisit a topic you will find yourself having achieved a higher level of understanding, provided you really try to get your head around it. This kind of material is not suitable for postponing putting all the work until revision time. It is also very difficult to catch up once you have fallen behind. For this reason doing set work every week is rewarded by giving you marks, determining 20% of the mark for each semester.

Notes. You should think of the lecture notes as providing an account of the *examinable material*. The exercises are part of that material—usually not the specific results obtained in them, but the techniques. You are required to *read the notes independently*.

Exercises. You will have worked on some of the exercises as part of the assessed work, the others are available for revision purposes. You will receive solutions to most[6] of the exercises in the course of term. There are also *optional exercises* which are not part of the examinable material, and to which no solutions will be handed out.[7] Note that the exercises appear in the text where they are more appropriate. When working on an exercise for assessed work you will almost certainly have to (re-)read the material preceding the exercise.

Lectures. The point of the lectures is to help you understand the notes. This is done by explaining the more abstract or complex concepts, and showing you examples. Not everything that is examinable will be mentioned in some lecture—what is written in the notes counts here.

Examples classes. The point of the examples classes is to give you feedback on exercises you have tried yourself, and to help you with questions you could not do, but also to help you with concepts or ideas you are finding difficult.

In that way all the contact hours are dedicated to help you learn the material, but ultimately what you put in yourself will be the biggest decider in what you get out.

Additional help. If you find you are struggling with the material there are a number of actions you may take. You may want to consult an alternative text in case the explanations as given in the notes don't suit your way of understanding. You should ask lots of questions in the examples class. You should also talk to your friends on the programme, for example the other members of your tutorial group. It may make sense for some of you to form a study group and to tackle the material together. The weekly PASS sessions are an excellent way of getting help from students who are in their second or third year.

[6] At a niminum this will include solutions to all assessed exercises.

[7] In case you are wondering: The optional exercises are intended or students who wish to obtain a more complete understanding of various concepts. Some of them are intended to get those interested started on thinking some way beyond the material presented here.

Expected effort

The School expects students to work around 40 hours a week (although some students put in more). This means that for every ten credit course unit you should expect to work roughly seven hours a week (including contact hours).

In COMP11120 you have three contact hours a week, which means you are expected to work **four hours a week** working through the notes and solving the exercises (in particular the assessed ones).

Academic malpractice

Academic malpractice is a general term that covers a number of ways in which students may break regulations by submitting work as their own which in part is based on unattributed contributions by others.

Note that the rules for collaboration are **different for COMP11120** from almost all other units you may take in the School of Computer Science. Please do not apply them to other units.

We are happy for you to

- try to understand the exercises in a group, including working out how to solve a particular exercise and
- compare your solution to somebody else's.

You **must not**

- copy somebody else's solution (even if you make some changes along the way) or
- give your solution to somebody (either to copy, or adapt, or to take a picture of it) or
- ask to look at somebody else's solution for an exercise which you haven't already solved.

If you are working as a group to understand a particular exercise, then each of you must **write down your own version** of your solution. If you aim to give a counterexample to a statement you should find different counterexamples based on the understanding you gained jointly. If you aim to give a proof then each of you should independently formulate your understanding, and you should not agree on using specific variable names, for example. Students who submit identical (or very similar) answers will be penalized.

If you and another student have solved an exercise independently then we are happy for you to compare your solutions, and if you think the solutions are different you should try to work out between you whether they are both correct, and why. If one of you has an incorrect solution, and the other solution is deemed correct, then the first student may change their solution, but this **must not be done** by merely copying the correct version. Instead the student should start from their incorrect solution and adjust that. Similarly, you **must not let another student copy** your solution, even if you feel sorry for them. They will get at least one mark for their incorrect attempt, and if they've understood what was wrong they should be able to fix their solution for two marks.

Where one student has solved an exercise and another hasn't it is fine for the first to explain how to tackle the questions, to give tips for how to

find the solution, and to explain the concepts involved. In a situation like this it is not okay to show your solution to the other student.

Remember also that you have to be able to explain your solution to get the marks, and that identical write-ups will be treated as copying.

These notes

You will receive a number of handouts over the course of the academic year, referred to here as 'these notes'. They will cover the following topics:

- complex numbers,
- statements and proofs,
- logic,
- recursion and induction,
- relations,
- linear algebra (vectors and matrices).

These notes have been tailored specifically to the curriculum in computer science here at the University of Manchester. They can be read on a number of levels, and are meant to accompany you through your time here, allowing you to go back and reread material as it becomes relevant to your studies in one of the more advanced course units.

The notes were new for the academic year 2014/15, and have been revised again for 2015/16. Nonetheless there is doubtlessly room for improvement. I would like to invite you to email us if you have suggestions, for example

- if you think you have found a mistake,
- if you think that there is a passage that is misleading,
- issues that require more detailed explanations,
- issues that require more worked examples,
- exercises whose instructions are unclear,
- examples and exercises that we should think about adding,
- online resources you have found useful.

We will certainly incorporate such suggestions for future years, but if there is a particularly important issue, or one where we get several requests, We may hand out supplementary notes as term progresses. We will use my interactions with you in examples classes and lectures to guide me, but we would very much appreciate receiving emails on this topic.

We would like to thank Graham Gough, Jonas Lorenz, Joe Razavi, Francis Southern, Yegor Guskov, Francisco Lobo, Sarah Nogueira, Chris Tedd, Luca Minciullo, David Pauksztello and Sami Alabed for helping us improve these notes. Some of the ideas for operations on lists come from notes originally written for COMP112 by David Lester, and the idea of including recursively defined functions for natural numbers is based on notes by Graham Gough for a previous version of this course.

Additional literature

There is no book that covers all the material in these notes. They have been specifically written for computer science students going through the curriculum here in Manchester and they cover some material that is not present in most text books. Nonetheless it can be beneficial to look at an alternative presentation of material, and we suggest some books you may want to try for this purpose. These are all present in the departmental as well as the university library for you to borrow. There are certainly other books available that cover much of the material, and you will also find the internet a useful resource on specific topics.

Here is an account of what is covered in which of the suggested books.

- **Basics**. If you find the material in this chapter difficult you should use resources on the web to help you, and maybe also get some of the books below and read about the relevant part.
- **Complex numbers**. This is not covered in most text books aimed at computer scientists, but you will find relevant material in Jordan and Smith. All the operations described in the notes also appear in the Wikipedia article on complex numbers at `https://en.wikipedia.org/wiki/Complex_number`, and there are a number of webpages available that aim to explain the general ideas.
- **Statements and proofs**. The definitions and many of their properties are covered in Epp, some of them also in Truss. All the definitions have entries on Wikipedia, and in many other online sources. (for example, the notion of a commutative operation is explained at `https://en.wikipedia.org/wiki/Commutative_property` such sources provide an alternative point of view that you may find helpful.
- **Logic**. We will not be following any particular text books, but Truss and Epp contain chapters that cover formal logical systems, in particular, the basics of Boolean semantics and properties of propositional formulas. Again you can use a search engine to find additional material on each topic on the web.
- **Probability** This subject is addressed to some extent in Epp, and also in Jordan and Smith, but less generally than in these notes.
- **Recursion and induction**. Recursion and induction are both covered in Epp as well as in Truss, but not at the same level of generality. There are many resources available on the web, for example the Wikipedia article on structural induction here `https://en.wikipedia.org/wiki/Structural_induction`.
- **Relations**. Relations are covered in both, Epp and Truss, and there are many online resources.
- **Linear algebra.**

Writing about mathematics is difficult, and in particular this holds for giving rigorous arguments. The following book gives a lot of good advice on that subject:
Kevin Houston, **How to Think Like a Mathematician**, *Cambridge University Press*, 2009, ISBN: 052171978X.

Here are some text books on mathematics for computer scientists:
Susanna Epp. **Discrete Mathematics with Applications**, *Brooks/Cole* 2011, ISBN: 0495826162.

This book covers much of our material as well as significant parts of COMP11212.

J.K. Truss. **Discrete mathematics for Computer Scientists**, *Addison-Wesley*, 1999. ISBN: 0201360616

This book covers roughly the same material as the first, but with fewer applications, and in slightly less detail.

D.W. Jordan and P. Smith. **Mathematical techniques: an introduction for the engineering, physical, and mathematical sciences**, *Oxford University Press* 2008, ISBN: 9780199282012.

Much of this book is concerned with continuous mathematics, so if you need a refresher it may well be useful for that purpose beyond what is said above.

D. C. Montgomery and George C. Runger. **Applied statistics and probability for engineers (5th edition)**, *Wiley* 2010, ISBN: 978047050578.

This is a very applied text which may help you with connecting the concepts taught here with applications.

E. Angel, **Interactive computer graphics: a top-down approach using OpenGL**, *Pearson* 2008, ISBN: 9780321549433.

This book connects the material on matrices and vectors with the intended application in computer graphics.

Chapter 0

Basics

This chapter explains some concepts most of which you should have encountered before coming to university. You may not have been given formal definitions. Whenever you find concepts used in the notes that have familiar names, but whose usage puzzles you, you should check whether there is something in this chapter which helps. There will be no lectures about the material in this chapter, but the examples classes in Week 1 are there to make sure you understand the ideas and notation used here. These will appear in other course units you take since they provide a universal language that is used outside of mathematics as well.

Note that we here assume that certain collections of numbers, with various operations, have already been defined. You will see formal definitions of most of these (real numbers being the exception) in Chapter 5 which we will study in Semester 2. The purpose of assuming they are present at the start is to allow us to use them as examples.

0.1 Numbers

Naively speaking, numbers are entities we often use when we wish to calculate something. Mathematically speaking, there is typically rather more going on: Numbers are sets with operations, and these operations have particular properties. Many of these properties are named and studied in Chapter 2.

0.1.1 Natural numbers

The **natural numbers** are often also referred to as *counting numbers*, and the collection of all of them is typically written as $\mathbb{N}$.

The simplest way of formally describing them is to say that

- there is a natural number 0 and
- given a natural number n there is another natural number Sn, the **successor of** n, more usually written as $n + 1$.

Every natural number can be generated in this way, although to reach 123456, for example, one has to apply the successor operation quite a few times! This also means that given a natural number n, we know that one of the following is the case:

- either $n = 0$ or

- there exists a natural number m with $n = Sm$ (or, if you prefer, $n = m + 1$).

This might seem like a trivial observation, but it is the basis of using the concept of *recursion* to define properties or functions for the natural numbers, and also for being able to prove properties by *induction*.

This is described in detail in Section 5.1 of these notes. Here we look at the informal notions you have met at school.

With the natural numbers come some operations we use:

- Given natural numbers m and n we can add these to get

$$n + m,$$

  and 0 is a unit, or neutral element, for that operation.

- Given natural numbers m and n we can multiply these to get

$$m \cdot n,$$

  and $1 = S0$ is a unit for that operation. Note that it is customary to write mn for $m \cdot n$ where this cannot lead to confusion.

In the exercises in Chapters 1–3 you are allowed to use the following about natural numbers:

Fact 1. *Given k, m, and n in $\mathbb{N}$ we have*

$$\begin{aligned} m + n &= n + m && \textit{commutativity of } + \\ (k + m) + n &= k + (m + n) && \textit{associativity of } + \\ m + 0 &= m = 0 + m && 0 \textit{ unit for } + . \end{aligned}$$

For the same variables we also have

$$\begin{aligned} m \cdot n &= nm && \textit{commutativity of } \cdot \\ (k \cdot m) \cdot n &= k \cdot (m \cdot n) && \textit{associativity of } \cdot \\ m \cdot 1 &= m = 1 \cdot m && 1 \textit{ unit for } \cdot . \end{aligned}$$

For the same variables we also have the property

$$k \cdot (m + n) = k \cdot m + k \cdot n \qquad \cdot \textit{ distributes over } + .$$

A mathematician might say that the natural numbers form a commutative monoid with unit 0 when looking at the addition operation, and a commutative monoid with unit 1 when looking at multiplication. In Section 2.3.2 we look formally at the properties given by these equalities.

There is one additional property we require.

Fact 2. *Given m in $\mathbb{N}$ and $n \in \mathbb{N} \setminus \{0\}$ there exist unique numbers k and l in $\mathbb{N}$ such that*

- $0 \leq l < m$ *and*
- $n = km + l$.

We use this fact to define a division operation on natural numbers, known as **integer division**[1]. One defines[2]

$$n \operatorname{div} m$$

to be the unique number $k \in \mathbb{N}$ in Fact 2. This is the number of times m divides n (leaving a remainder). One defines the **remainder for integer division** by setting

$$m \bmod n$$

to be the unique l from Fact 2. This is the remainder n leaves when divided by m.

Note that it is not necessarily the case that

$$m \cdot (n \operatorname{div} m) = n,$$

for example

$$2 \cdot (3 \operatorname{div} 2) = 2 \cdot 1 = 2 \neq 3.$$

This is different to the way a division operation is usually defined (see the discussion in the section on rational numbers below), and that is the reason that this kind of division has a different name, and a different symbol.

One may define notions of divisibility and evenness for natural numbers. We give formal definitions for integers in the following section, and these definitions also apply to natural numbers.

We might also want to think about which equations we can solve in the natural numbers. Assume that m and n are elements of $\mathbb{N}$.

For example, we can solve

$$m + x = n,$$

within $\mathbb{N}$, provided that[3] m is less than or equal to n, which we write as $m \leq n$.

We can also solve

$$mx = n,$$

within $\mathbb{N}$ provided that $n \bmod m = 0$. Because of the side conditions required we see that a lot of equations we can write down using the available operations do not have a solution.

We can use the natural numbers to count something, for example the number of instructions in a computer program, or the number of times a program will carry out the body of a loop. This is important to do when we are trying to estimate how long it may take a program to run on a large-size problem.

There are a lot of natural numbers, namely infinitely many. But by mathematical standards the natural numbers are the smallest infinite set, and there are substantially larger ones. Sets of this size are set to be *countably infinite*. This is formally defined in Section 4.0.1.

Computer languages do typically *not* implement the natural numbers—instead, a programming language will have support for all natural numbers up to a particular maximum. Nothing truly infinite can be implemented in any real-world computer (but there are theoretical computation devices which have infinite storage). Quite often programming languages have a built-in type for integers instead of natural numbers, as is the case with Java.

[1] Sometimes also called *Euclidean division*.

[2] See the following section to see that this idea can be extended to the integers.

[3] The solution to such an equation would have to satisfy $x = n - m$ and this is not always defined.

0.1.2 Integers

A simple way of explaining the **integers** is that one wants to expand the natural numbers in order to make it possible for every number to have an *inverse* with respect to addition, that is, for every number n there is a number m, usually written as $-n$, with the property that

$$n + m = 0 = m + n.$$

.

Defining the integers formally in a way that supports the above idea is quite tricky. Such a description is given in Chapter 5 on page 170. It's fairly easy to describe the elements of this set, called[4] $\mathbb{Z}$, once one has the natural numbers since one can say

$$\mathbb{Z} = \mathbb{N} \cup \{-n \mid n \in \mathbb{N} \setminus \{0\}\},$$

but this does not tell us anything about how to calculate with these numbers. So this does not, mathematically speaking, define the integers with all the operations we customarily use for them.

The **absolute,** $|m|$, of an integer m is defined to be[5]

- m if m is greater than or equal to 0 and
- $-m$ if m is less than 0.

We (very rarely) use $\mathbb{Z}^+$ to refer to those integers[6] which are greater than or equal to 0.

Fact 3. *The equalities from Fact 1 also hold if the variables are elements of* $\mathbb{Z}$*. Additionally,*

for every $n \in \mathbb{Z}$ *there exists a unique* $m \in \mathbb{Z}$ *with* $m + n = 0 = n + m$*.*

We say that this number m *is the* ***additive inverse for*** n ***with respect to addition****. The number* $-n$ *is defined to be the additive inverse of* n*.*

A mathematician would say that $\mathbb{Z}$ forms a commutative ring with multiplicative unit 1.

Many people use *subtraction* as an operation. However, it is much preferable to think of this not as an operation, but as

$$n - m$$

being a shortcut for adding the additive inverse of m, $-m$, to n—in other words, this is merely a shortcut for

$$n + (-m).$$

Please do not talk about subtraction on this course unit, but about adding additive inverses. The reason this is good discipline is that there are many

[4]The notation $\mathbb{Z}$ for the set of integers is very common within mathematics, the letter coming from the German word 'Zahlen', or numbers. You may know this set under a different name, but that should not worry you.

[5]See Example 0.12 for a definition of this as a function, although that definition is for real numbers.

[6]This set of numbers is, of course, equivalent to $\mathbb{N}$

situations where not all inverses exist,[7] and so you should pause to think whether the operation you wish to carry out is legal.

Fact 2 changes a bit when we use it for integers.

Fact 4. *Given m in $\mathbb{Z}$ and $n \in \mathbb{Z} \setminus \{0\}$ there exist unique numbers k and l in $\mathbb{Z}$ such that*

- $0 \leq l < |m|$ *and*
- $n = km + l$.

Hence we may extend the definitions of the operations of mod and div, that come from **integer division**, to the integers. In other words, for integers m and n,

- $n \operatorname{div} m$ is the unique k, and
- $n \bmod m$ is the unique l,

from the above fact.

You are expected to use the following definitions when answering exercises about natural numbers.

The notions ofevenness and oddness transfer with the same definitions as for natural numbers.

Definition 1. Given integers m and n we say that n **is divisible by** m or that m **divides** n if and only if there exists an integer k such that

$$m \cdot k = n.$$

Note that m divides n if and only if $n \bmod m = 0$.

Exercise 1. Use Fact 2 to argue that the previous sentence is correct.

Definition 2. An integer n is **even** if and only if n is divisible by 2. Such a number is **odd** if and only if it is not divisible by 2.

Exercise 2. Use Fact 2 to argue that a natural number n is even if and only if $n \bmod 2 = 0$, and odd if and only if $n \bmod 2 = 1$.

How do the even numbesrs relate to those numbers which are a multiple of 2? Can you make your answer formal? Do your answers change if n is an integer?

The fact that every number has an additive inverse means that for m and n in $\mathbb{Z}$ we can solve all equations of the form

$$m + x = n$$

within $\mathbb{Z}$ without reservations. Indeed, every equation in one variable which involves addition and inverses has a unique solution. On the other hand, equations of the form

$$mx = n$$

[7] For example, for the rational and real numbers, the number 0 has no multiplicative inverse, see Facts 5 and 6.

are still not all[8] solvable within $\mathbb{Z}$.

If we accept that there are infinitely many natural numbers then it is clear that there are also infinitely many integers. Because the natural numbers are embedded inside the integers one might assume that there are more of the latter, But actually, this is not a sensible notion of size for sets. Mathematically speaking, $\mathbb{N}$ and $\mathbb{Z}$ have the same size, see Section 4.0.1 for details of what that statement means.

Many programming languages support a data type for the integers. However, only finitely many of them are represented. In Java, for example, integers are given by the primitive type `int`, and range from -2^{31} to $2^{31}-1$. In the programming language C the language specification does not state what the smallest and greatest possible integers are—different compilers have different implementations here. You have to work out what is safe to use in your system.

0.1.3 Rational numbers

One can view the **rational numbers**, usually written[9] as $\mathbb{Q}$, as the numbers required to have a multiplicative inverse for every number other than 0. But again, giving a formal definition of these numbers is not straightforward if one wants to ensure that all the previous operations are available.

One way of talking about the rational numbers is to introduce the notion of a *fraction*, written as

$$m/n,$$

where m and n are integers.

But we cannot define the rational numbers to be the collection of all fractions since several fractions may describe the same rational number: We expect $2/4$ to describe the same number as $1/2$

Formally we have to define a notion of equality (or equivalence) on fractions, whereby

$$m/n = m'/n' \qquad \text{if and only if} \qquad mn' = m'n.$$

There is a formal definition of the rational numbers, and their addition and multiplication, on page 173.

We have quite a bit of structure on $\mathbb{Q}$. All the facts for integers still hold, but we get a new property.

Fact 5. *The statements from Fact 3 remain true if all variables are taken to be elements of* $\mathbb{Q}$*. In addition,*

$$\textit{for all } q \in \mathbb{Q} \setminus \{0\} \textit{ there exists } q' \in \mathbb{Q} \textit{ such that} q \cdot q' = 1 = q' \cdot 1.$$

We say that q' *is the* ***multiplicative inverse for*** q*. Every element* $q \neq 0$ *has a multiplicative inverse and the standard notation for this element is* q^{-1}*.*

A mathematician would say that $\mathbb{Q}$ with addition and multiplication is a field.

[8] The solution would have to satisfy $x = m/n$, and this is not defined for all m and n.

[9] The name comes from the Italian 'quoziente', quotient. We look at why this is in Semester 2, see Section 6.3.

Many people speak of *division* as an operation on rational (and real) numbers, but again, this is merely a shortcut: Writing

$$q'/q$$

is an instruction to multiply q' with the multiplicative inverse of q, that is, it is a shortcut for

$$q' \cdot q^{-1}.$$

The number 0 does not have a multiplicative inverse, and that is why division by 0 is not allowed. In this course unit, please try not refer to division as an operation, and when you multiply with inverses, always check to ensure these exist.

Exercise 3. What properties would a multiple inverse for 0 have to satisfy? Argue that one such cannot exist.

The notion of the absolute can be extended to cover the rationals, using the same definition.

Given q and q' in $\mathbb{Q}$ we can now solve all equations of the form

$$q + x = q' \qquad \text{and} \qquad qx = q' \qquad (\text{if } q \neq 0)$$

within $\mathbb{Q}$, provided that,[10] for the second equation, $q \neq 0$. And indeed, every equation with one unknown involving addition, multiplication and inverses for these operations is solvable provided that it is not equivalent to one of the form $qx = q'$ where $q = 0$, $q' \neq 0$.[11].

The rational numbers are sufficient for a number of practical purposes; for example, to measure the length, area, and volume of something to any given precision, and also to do calculations with such quantities.

There are infinitely many rational numbers, but mathematically speaking, $\mathbb{Q}$ has the same size as $\mathbb{N}$. See Section 4.0.1 for how to compare the size of sets.

Most mainstream programming languages do not have a datatype for the rationals (or for fractions), but those aimed a algebraic computations (such as Mathematica and Matlab) do.

0.1.4 Real numbers

The rational numbers allow us to measure anything up to arbitrary precision, we may add and subtract them, and there are additive and multiplicative inverses (the latter with the exception of 0), which allows us to solve many equations. Why do we need a larger set of numbers?

There are several approaches to this question. Here we give two. If we look at the rational numbers drawn on a line then there are a lot of gaps.

Mathematically speaking we may define a *sequence* (of rational numbers), that is a list of numbers

$$x_n \in \mathbb{Q}, \qquad \text{one for each } n \in \mathbb{N}.$$

[10] Note how the restrictions we have to make on equations to ensure they are solvable connect with where the operations involved are defined (or not) for the various sets of numbers discussed here.

[11] Should I say something somewhere about equations with more than one unknown? Do I need to introduce the notion of a system of linear equations?

Sometimes a sequence can be said to *converge to a number*, that is, the sequence gets arbitrarily close to the given number and never moves away from it.[12] If such a number exists it is called the *limit of the sequence*. For example, the limit of

$$1,\ 1/2,\ 1/4,\ 1/8,\ \ldots \qquad \text{that is} \qquad 1/2^n \text{ for } n \in \mathbb{N}$$

is 0.

Let us consider the sequence defined as follows:

$$x_0 = 1$$
$$x_{n+1} = \frac{x_n^2 + 2}{2x_n}$$

We may calculate the first few members of the sequence to get

$$1,\ 3/2,\ 17/12,\ 577/408,\ \ldots$$

and, if expressed in decimal notation,

$$1,\ 1.5,\ 1.41\overline{6},\ 1.4142568627451,\ \ldots$$

One may show that

$$x_n^2$$

gets closer and closer to 2, so we may think of the above as approximating a number r with the property that $r^2 = 2$.

Optional Exercise 1. Show that there is no rational number q with the property that $q^2 = 2$. *Hint: Assume that you have $q = m/n$ for some natural numbers m and n and derive a contradiction.*

Hence there are numbers that are approximated by sequences of rational numbers which are not themselves rational. Or, if we draw the rational numbers as a line then it has a lot of gaps.

One can define the notion of a *Cauchy sequence*. One may think of this as a sequence that should have a limit (because the sequence contracts to a smaller and smaller part of the rational numbers), but where there is no suitable rational number for it to converge to. One can define the **real numbers** $\mathbb{R}$ as being all the limits for all the Cauchy sequences one can build from the rationals. This gives a 'complete' set of numbers in the sense that every Cauchy sequence built from elements of $\mathbb{R}$ has a limit in $\mathbb{R}$. We use $\mathbb{R}^+$ to refer to those real numbers which are greater than or equal to 0. The numbers in $\mathbb{R} \setminus \mathbb{Q}$ are known as the *irrational numbers*.

We do not give a formal definition of the real numbers in these notes—the above outline should convince you that this is reasonably complicated to do rigorously. We may think of the rational numbers as being included in $\mathbb{R}$. Also the real numbers again come with the operations of addition and multiplication, and inverses for these (but 0 still does not have a multiplicative inverse), and we again have the previous distributivity law for these operations. Just like the rational numbers, the reals with these operations form a *field*.

[12] This can be defined mathematically but would take up more space than we want to give it here.

An alternative approach to introducing numbers beyond the rationals is as follows. Within the rational numbers we are able to solve all 'sensible' equations in one variable involving addition, multiplication and their inverses with rational numbers. We may even add multiples of that variable with each other.[13] But we may not multiply the unknown with itself: Equations of the form

$$xx = q \qquad \text{or} \qquad x^2 = q$$

are not all solvable with $\mathbb{Q}$. By moving from $\mathbb{Q}$ to $\mathbb{R}$ we add a lot of solutions to such equations to our set of numbers. For example, all equations of the form

$$x^n = r$$

are solvable for $n \in \mathbb{N}$ and $r \in \mathbb{R}^+$. Indeed, we may replace $n \in \mathbb{N}$ by $q \in \mathbb{Q}$ and we still have solutions.[14]

The situation becomes quite complicated. First of all we define a new symbol: We write

$$\sum_{i=0}^{n} a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

for a sum of a finite number of elements.[15]

Given a **polynomial equation**, that is one of the form

$$\sum_{i=0}^{n} a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = 0$$

where $a_i \in \mathbb{Q}$ for $0 \leq i \leq n$ there may be up to n different solutions, or there may be none at all. Those real numbers that are solutions to such polynomial equations are known as *algebraic numbers*. Examples are $\sqrt{2}$, $\sqrt[5]{17}$ and $\sqrt[3]{3/2}$.

But not all elements of $\mathbb{R}$ can be written as solutions to such equations. Those that can not are the *transcendental numbers*; famous examples are e and π, and less well-known ones e^π and $2^{\sqrt{2}}$. We can therefore *not*[16] use the idea that $\mathbb{R}$ arises from $\mathbb{Q}$ by adding solutions to equations over $\mathbb{Q}$ to formally define $\mathbb{R}$.

Fact 6. *All statements from Fact 5 remain true if the variables are taken to be elements of* $\mathbb{R}$.

A mathematician would say that the real numbers, with additiona nd application, also from a field.

All the sets of numbers discussed so far are *ordered*, that is, given two numbers we may compare them. See Section 6.4 on how one generally talks about this idea. Here we are concerned with giving additional facts you may want to use in solving exercises.

[13] These equations are called *linear*.

[14] And we may even replace $q \in Q$ by $r' \in \mathbb{R}$ and use the idea of the *continuity of a function* to define the operation of forming x to the power of r', and we can still find solutions.

[15] This idea is formally introduced on page 5.1 in Chapter 5.

[16] There is a way of algebraically defining the real numbers, but that requires a lot of mathematical theory to be set up that is fairly advanced.

Fact 7. *Let r, r', s and s' be elements of $\mathbb{R}$. Then the following hold:*

If $r \leq r'$ *and* $s \leq s'$	*then*	$r + s \leq r' + s'$.
If $r \leq r'$ *and* $s \geq 0$	*then*	$r \cdot s \leq r' \cdot s$.
If $r \leq r'$ *and* $s \leq 0$	*then*	$r \cdot s \geq r' \cdot s$.
If $r \leq s$	*then*	$-r \geq -s$.
If $r \leq s$	*then*	$r^{-1} \geq s^{-1}$

Real numbers are often referred to using *decimal expansions*. Such an expansion is given by an integer together with a sequence of digits (one digit for each natural number). For the integer 0 for example one gets numbers typically written

$$0.d_1d_2d_3\ldots,$$

where for $i \in \mathbb{N}$ we know that $d_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. It is common not to write trailing 0s (that is 0s where there is no different digit occurring to the right), so we write 3.14 instead of $3,14000000\ldots$. Note, however, that a number may have more than one decimal expansion, and that $0.9999999\ldots$ refers to the same number as $1.0000000\ldots = 1$.

We don't really need real numbers in the 'real world', but a lot of what we might want to describe becomes a lot smoother if we are allowed to use them (the trajectory of a ball is much easier thought of as a line than a sequence of points with rational coordinates), and they allow us to be precise when referring to the circumference of a circle, for example.

The set $\mathbb{R}$ is infinite in size—but mathematically speaking, it is strictly larger than $\mathbb{Q}$. It is *uncountably infinite*. See Section 4.0.1 for more details.

No real-world computer can implement all the real numbers. This is no surprise given that there are infinitely many of them. But this also means that every implementation of (some of) the real numbers will only allow limited precision.[17] Programming languages typically have some kind of *floating point number* type to approximate real (and so also) rational numbers, such as `float` in Java. It's not unusual for there to be a more precise type, such as `double` in Java. Note that operations on such numbers typically incur *rounding errors* (for these operations to be precise it would be necessary to change the range of numbers which are representable by adding more digits—for example, $.5/2.0 = .25$, and we need to go from 1 digit after the decimal point to 2). In Java there is also *bignum* which allows for arbitrary precision (since the maximal allowable length of the number can be extended), provided the number has a finite decimal expansion, but these come at a price in memory and time performance (and a program that keeps adding digits will eventually run out of memory). Floating point numbers are given by a *significand* and an *exponent* (because this increases the range of numbers that can be represented), where for a given base, the number described is

$$\text{significand} \times \text{base}^{\text{exponent}}.$$

[17] There are some languages where it is possible to carry out calculations to a pre-defined precision, but these are not main stream, and significant overhead is required to make this work properly.

0.1.5 Numbers

We typically think of the sets of numbers introduced here as being subsets of each other, with

$$\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}.$$

Mathematically speaking, this is not strictly correct, but instead we have a function that embeds the integers, say, in the rationals, in such a way that carrying out operations from the integers also works if we think of the numbers as rationals.

Sometimes in these notes we do not want to specify which set of numbers we mean, and then we assume there is a set N with an addition and a multiplicaiton operation.

0.2 Sets

Sets are very important in mathematics—indeed, modern mathematics is built entirely around the notion of sets.

A **set** is a collection of items. Collections are required in order to

- make it clear what one is talking about (ruling some things in and others out);
- precisely define various collections of numbers—and in general, much of algebra is concerned with structures given by
  - an *underlying set* (for examples see Section 0.1 for various sets of numbers which, however, aren't formally defined here),
  - operations on the set (such as addition and multiplication, for various collections of numbers) and
  - the properties of these operations, typically given by equalities that have to hold (for example $x + y = y + x$ is true for all x and y both in $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ or $\mathbb{R}$);[18]
- define *functions* (see following section)—instructions for turning entities of one kind into entities of another.

Sets have *members* and indeed a set is given by describing all the members it contains. We write

$$s \in S$$

if s is an member of the set S. This sounds simpler than it turns out to be. Members are often also referred to as *elements*.

0.2.1 Set theory

Deciding which collections of entities may be considered sets is not as easy as it might sound. Originally mathematicians thought that there would not be any problems in allowing any collection to be considered a set, but very early into the 20th century Bertrand Russel described the *paradox* named after him:

If we are allowed to form the set of all sets which contain themselves as members then we have a contradiction.[19] Theories that contain contradictions are called *inconsistent*, and they are not very useful since (at

[18] These properties are studied in more detail in Section 2.3.2.

[19] Ask yourself whether the given 'set' contains itself.

least according to classical logic) *every statement* may be deduced in an inconsistent theory. But if every statement is valid then the theory is of no use.

This caused something of a crisis, and prompted the creation of **set theory** as a field within mathematics. Set theory is concerned with the question of describing how sets may be built in a way that does not lead to contradictions. Mathematicians need to build fairly complicated sets, and making sure that all their constructions are allowed in the underlying set theory is not easy. The sets we require on this course unit are nothing like as complicated and so we do not have to worry about proper set theory here (and you should not refer to what is described in this section as 'set theory').

0.2.2 Operations on sets

The most fundamental operations on sets we may use is to *compare*[20] them.

Definition 3. A set S is a **subset** of the set T, written

$$S \subseteq T,$$

if and only if every member of s is also an member of T.

In this situation we have

$$s \in S \qquad \text{implies} \qquad s \in T,$$

or

$$\text{for all } s \in S \qquad\qquad s \in T.$$

Note that the usage of key phrases such as 'implies', 'there exists', 'for all' is described in detail in Chapter 2.2.1.

If $S \subseteq T$ and $T \subseteq S$ then $S = T$ because they contain precisely the same members.

Definition 4. We say that a set S **is a proper subset of the set** T if and only if

- S is a subset of T, that is $S \subseteq T$ and
- there exists a member $t \in T$ with $t \notin S$,

Sometimes the notations

$$S \subset T \qquad \text{or} \qquad S \subsetneq T$$

are used in this situation.

When we have sets we may build new sets by putting their combined members into one set, or by considering only those members contained in both sets. Because constructing new sets is non-trivial and may lead to problems it is usually better first to find an 'ambient' set that contains both the given sets.

Given a set X, for S and T subsets of X, we define

[20] A more general notion of comparisons between sets is studied in Section 4.0.1.

- their **union**, $S \cup T$, to be

$$\{x \in X \mid x \in S \text{ or } x \in T\};$$

- their **intersection**, $S \cap T$, to be[21]

$$\{x \in X \mid x \in S \text{ and } x \in T\}.$$

Note that we may now define the union or intersection of finitely many subsets of X by applying the operation to two sets at a time, that is, for example,

$$S_1 \cup S_2 \cup S_3 \cup \cdots \cup S_n = (\cdots((S_1 \cup S_2) \cup S_3) \cup \cdots \cup S_n).$$

Again we have used $\cdots$ here, and to be more precise we should adopt the mathematical notation

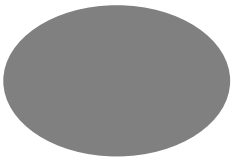
$$\bigcup_{i=1}^{n} S_i$$

instead, which spells out that we are forming the union of all the sets from S_1 to S_n.

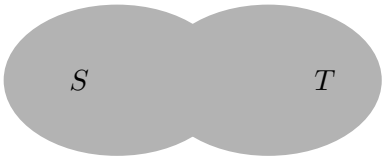
But in fact, given an arbitrary collection of subsets of X we may define their union and their intersection to obtain another subset of X.

Optional Exercise 2. How could you describe an 'arbitrary collection of subsets of X? Can you come up with a definition of the union of all these sets?

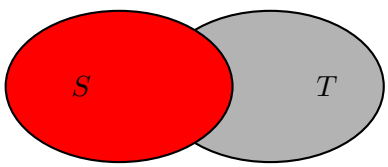
It is sometimes useful to draw such constructions in the form of a **Venn diagram**.



This is a picture of a generic set. The union of two generic sets, S and T, can then be drawn as follows.



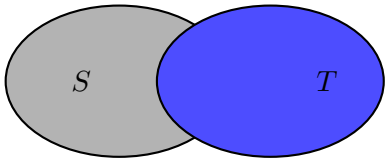
But this is a bit imprecise if we do not draw the boundaries of the sets, so it is more common to draw the boundaries of all the sets involved. We assume we have a set S, here shown in red,[22]



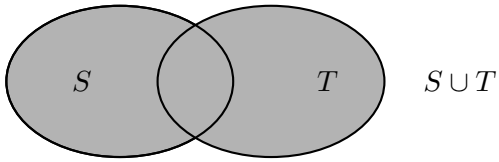
[21] Now we want all elements of X which belong to both, S and T.

[22] You will see the colours only in the electronic but not in the printed version.

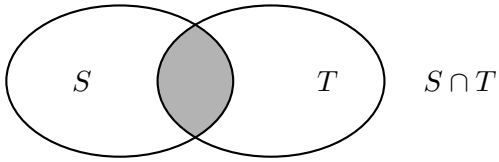
and a set T, here shown in blue



for which we form the union $S \cup T$ (here in grey).



The picture for the intersection, again drawn in grey:



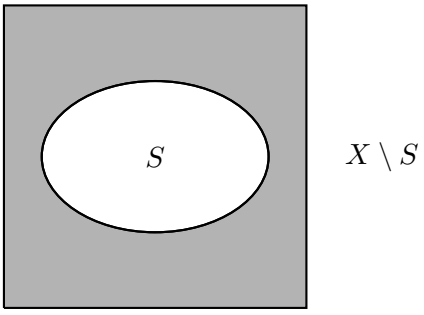
There is one further important operation on sets.

Definition 5. Let S be a subset of a set X. The **complement of S relative to X**, $X \setminus S$, is given by

$$\{x \in X \mid x \notin S\}.$$

Some people write $X - S$ for this set, and some people write S' or $\overline{S}$. The latter two require that it is clearly understood which ambient set (here X) is meant. It has the advantage that some properties can be formulated very concisely in that notation. We do *not* use the primed version for complement in these notes—instead, we use it to give us variable names (so S, S' and S'' might be names for different sets).

If we want to draw the complement then we *have* to draw the ambient set X. (We didn't have to do this for the examples so far.[23]) We do this by drawing a square, with S living inside the square.

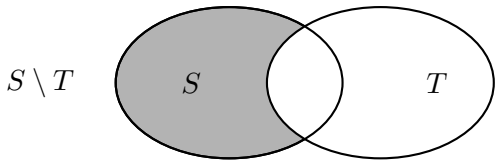


These are all the operations required to build new sets from given ones. It is now possible, for example, to define the **set difference**, $S \setminus T$, of all members of S that do not belong to T,

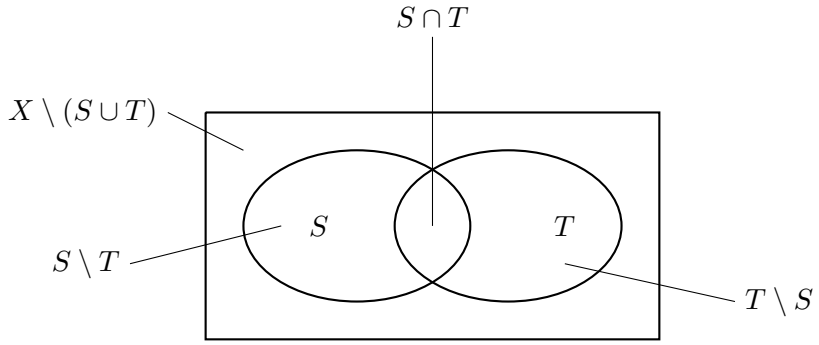
$$\begin{aligned} S \setminus T &= \{s \in S \mid s \notin T\} \\ &= S \cap (X \setminus T), \end{aligned}$$

[23] We could have drawn a box around the diagrams given above, but this doesn't really add anything.

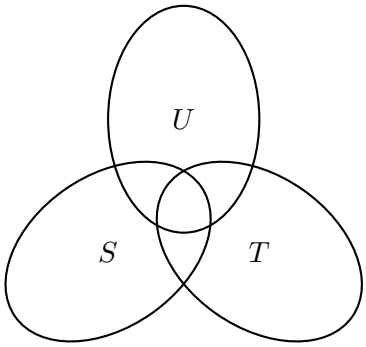
drawn in grey below.



Example 0.1. We can use these operations to give names relative to S and T to all the regions in the following picture. This means we know how to determine the elements of all these regions, provided we know when an element is in S, and when it is in T.



If we have more than two sets to start with then there are many more sets one could describe, but we now have to tools to do so for all of them.



Exercise 4. Identify all regions in the above picture and give their description based on operations applied to S, T, and U.

Exercise 5. Assume that S and T are subsets of a set X.

(a) Show that the complement relative to X of the union of S and T is the intersection of the complements (relative to X) of S and T. *Hint: Turn the sentence into an equality of sets.*

(b) Show that the union of two sets may be written using only the complement and the intersection operations. *Hint: Use your equality from the previous part.*

(c) Given an argument that we may describe precisely the same sets using ($\cup$, $\cap$ and $\setminus$) as using ($\cap$ and $\setminus$).

A useful operation assigns to a finite set the number of elements in that set, which is written as[24][25]

$$S \longmapsto |S|.$$

0.2.3 Describing sets

Describing sets precisely is harder than it may sound. If a set has finitely many elements then, in principle, we could list them all. But if there are a lot of them this is rather tedious and time-consuming. People often resort to using ... to indicate that there are members that are not explicitly named, and they hope that it is clear from the context what those members are. Take for example

$$\{0,\, 1,\, 2,\, \ldots,\, 100,000\}.$$

But whenever this notation is used there is room for confusion. It is *much* better to give a more precise description such as

$$\{n \in N \mid\ n \leq 100,000\}.$$

The idea behind this kind of description is that one describes the set in question as a *subset of a known set* (here $\mathbb{N}$), consisting of all those members satisfying a particular *property* (here being less than or equal to $100,000$). In logic such a property is known as a *predicate*. It is almost inevitably the case that any set we might want to describe is a subset of a set already known, so this technique works remarkably often.

Example 0.2. Let's assume we want to describe the set of even natrual numbers. We could write

$$\{0,\, 2,\, 4,\, 6, \ldots\},$$

but this leaves it to the reader to make precise which elements belong to the set and which ones don't. This is strongly discouraged. Instead we could write the preferable

$$\{n \in \mathbb{N} \mid\ n \text{ even}\},$$

but that assumes that the reader knows how the even property is defined. If we want to leave no room for doubt we could apply the definition (see Definition 2) and write

$$\{n \in \mathbb{N} \mid\ n \bmod 2 = 0\}.$$

This leaves no room for doubt which members belong to our set—indeed, it gives us a test that we can apply to some given natural number to see whether it belongs to our set.

Because we may form intersections and unions of sets we may also specify sets consisting of all those elements which have more than one property. All even numbers up to $100,000$ could be described as an intersection, namely

$$\{n \in \mathbb{N} \mid\ n \bmod 2 = 0\} \cap \{n \in \mathbb{N} \mid\ n \leq 100,000\},$$

[24] Some texts may use $\#S$ instead.

[25] If you are not familiar this notation to describe a function come back to this once you have read Section 0.3.

but it is more customary instead to combine both properties by using 'and', that is

$$\{n \in \mathbb{N} \mid n \bmod 2 = 0 \quad \text{and} \quad n \leq 100,000\}.$$

When looking at the real numbers there is a standard way of defining subsets which give a contiguous part of the real line:

$$[x, y] = \{r \in \mathbb{R} \mid x \leq r \leq y\}$$

and

$$(x, y) = \{r \in \mathbb{R} \mid x < r < y\},$$

or

$$[x, y) = \{r \in \mathbb{R} \mid x \leq r < y\},$$

but also

$$(-\infty, y] = \{r \in \mathbb{R} \mid r \leq y\}.$$

Sets of this form are known as 'real intervals'.

We may also use the idea of defining sets using properties to describe all those elements of a given set which satisfy at least one of several properties, for example

$$\{n \in \mathbb{N} \mid n \bmod 2 = 0 \quad \text{or} \quad n \bmod 2 = 1\}$$

is the union of two sets, namely

$$\{n \in \mathbb{N} \mid n \bmod 2 = 0\} \cup \{n \in \mathbb{N} \mid n \bmod 2 = 1\},$$

and this set is equal to $\mathbb{N}$.

Example 0.3. It is also possible to use this idea to specify the elements that *do not* have a particular property. The odd natural numbers are those that are not even.

$$\begin{aligned}\{n \in \mathbb{N} \mid n \bmod 2 \neq 0\} &= \mathbb{N} \setminus \{n \in \mathbb{N} \mid n \bmod 2 = 0\} \\ &= \{n \in \mathbb{N} \mid n \bmod 2 = 1\}.\end{aligned}$$

Example 0.4. Of course these ideas do not merely apply to natural numbers. We can start from any set that has been previously defined. For example, if we want to descibe the rational numbers as a subset of $\mathbb{R}$ we may use

$$\{r \in \mathbb{R} \mid \text{there exists } m \text{ and } n \text{ in } \mathbb{N} \text{ such that } r = m/n\},$$

Note that there is no guarantee that there are any elements in the set we describe; nothing stops us from specifying

$$\{n \in \mathbb{N} \mid n \bmod 2 = 0 \quad \text{and} \quad n \bmod 2 = 1\},$$

which is a rather complicated description of the empty set.

Example 0.5. It is possible to use infinitely many restricting properties. Given a natural number k, the multiples of k can be written as

$$\{n \in \mathbb{N} \mid n \bmod k = 0.\}.$$

So the set of natural numbers which are *not* multiples of k is

$$\{n \in \mathbb{N} \mid n \bmod k \neq 0\}.$$

If we want to write the set of prime numbers then we want the set of all those numbers which are not a multiple (with a factor of at least 2) of k for any $k \in \mathbb{N}$.

This suggests that we can use the intersection of all the sets of multiples of k, but we have to make sure that we leave $1 \cdot k$ in. Hence the set of prime numbers is

$$\bigcap_{k \in \mathbb{N} \setminus \{0,1\}} \{n \in \mathbb{N} \mid n \bmod k \neq 0 \text{ or } n = k\}$$

Instead of restricting the elements of a known set to describe a new set it is sometimes possible instead to provide instructions for *constructing the elements* of the new set. This is the second important technique for describeing sets.

Example 0.6. An alternative way of describing the even numbers is to recognize that they are exactly the multiples of 2, and to write

$$\{2n \mid n \in \mathbb{N}\}.$$

The odd numbers may then be described as

$$\{2n + 1 \mid n \in \mathbb{N}\}.$$

But for a better answer, we should add something here. Read on to find out what.

We can think of this as *constructing* a new set, but usually this only makes sense when describing a subset of a previously known set. Certainly the notation assumes that we know what we mean by $2n$, or $2n + 1$—this implies we know where the addition and multiplication operations that appear in these expressions are to be carried out. In this examples it is in $\mathbb{N}$, so it would be better to write

$$\{2n + 1 \in \mathbb{N} \mid n \in \mathbb{N}\}.$$

This may seem obvious, since $\mathbb{N}$ is explicitly named as the set n belongs to, but assume that in order to describe the rational numbers we wrote

$$\{m/n \mid m, n \in \mathbb{N}\ , n \neq 0\}.$$

But what does this mean? Where do we take m/n? This is not defined in the collection $\mathbb{N}$ of numbers where m and n live, and so it is not clear what we mean here. Maybe this is a set of formal fractions? We could clarify this by writing

$$\{m/n \in \mathbb{R} \mid m, n \in \mathbb{N},\ n \neq 0\},$$

from which it is clear that we we mean to collect all the results of calculating $m \cdot n^{-1}$ within the real numbers.

Example 0.7. To describe the integer multiples of π (for example if we want to have all the points on the real line for which sin takes the value 0) we might write,

$$\{n\pi \mid n \in \mathbb{N}\}.$$

Again we have to deduce from the context where $n\pi$ is meant to be carried out. If we write

$$\{n\pi \in \mathbb{R} \mid n \in \mathbb{N}\},$$

then everything is made explicit.

Exercise 6. For the sets given below, give a description using a predicate (as in Example 0.2), and also give a description where you generate the set (as in Example 0.6).

(a) Describe the set of all integers that are divisible by 3.

(b) Describe the set of all integers that are divisible by both, 2 and 3.

(c) Describe the set of all integers that are divisible by 2 or by 3. To generate this set you need to use the union operation.

(d) Describe the set of all integers that are divisible by 2 or by 3 but not by 6. To generate this set you need to use the union, and the relative complement, operations.

(e) all real numbers r for which $\cos r = 0$.

0.2.4 Constructions for sets

There is one other fairly common construction for sets.

Definition 6. Given sets X and Y their[26] **product** is the set

$$\{(x, y) \mid x \in X \text{ and } y \in Y\}.$$

This means that the elements of the product are pairs whose first component is an element of X and whose second component is an element of Y.

For example, the product of the set $\{0, 1\}$ with itself is the set with the elements

$$(0, 0),\ (0, 1),\ (1, 0),\ (1, 1).$$

Example 0.8. A more familiar example is a deck of cards: You have four suits, clubs ♣, spades ♠, hearts ♡ and diamonds ♢, and you have standard playing cards, say 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A in a 52-card deck. Each of those cards appears in each of the suits, so you have four Queens, one each for clubs, spaces, hearts and diamonds. In other words, your 52-card deck can be thought of as the product

$$\{\clubsuit, \spadesuit, \heartsuit, \diamondsuit\} \times \{2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A\}.$$

We can picture the result as all combinations of elements from the first set with elements from the second set.

[26] This is also known as their **Cartesian product**.

	2	3	4	5	6	7	8	9	10	J	Q	K	A
♣	2♣	3♣	4♣	5♣	6♣	7♣	8♣	9♣	10♣	J♣	Q♣	K♣	A♣
♠	2♠	3♠	4♠	5♠	6♠	7♠	8♠	9♠	10♠	J♠	Q♠	K♠	A♠
♡	2♡	3♡	4♡	5♡	6♡	7♡	8♡	9♡	10♡	J♡	Q♡	K♡	A♡
♢	2♢	3♢	4♢	5♢	6♢	7♢	8♢	9♢	10♢	J♢	Q♢	K♢	A♢

Example 0.9. A very important set that is a product is the *real plane*

$$\mathbb{R}^2 = \{(r, r') \mid r, r' \in \mathbb{R}\}.$$

This is the set we use when we draw the graph of a function from real numbers to real numbers (see Section 0.3.3), where we use the first coordinate to give the argument, and the second argument to give the corresponding value.

Similarly, the n-dimensional vector space based on $\mathbb{R}$ has as its underlying set the n-fold product of $\mathbb{R}$ with itself,

$$\mathbb{R}^n = \{(r_1, r_2, \ldots, r_n) \mid r_1, r_2, \ldots r_n \in \mathbb{R}\}.$$

Note that here we construct a new set, and we define what the elements of the set are (namely pairs of elements of the given sets) and we do not have to identify an ambient set.

When we describe operations on sets as functions (see the following section) we require the product construction. Addition, for example, is a **binary operation**[27] on, say, the natural numbers $\mathbb{N}$. As a function (also see following section) it takes two elements, say m and n, of $\mathbb{N}$, that is

$$\text{an element } (m, n) \text{ of } \mathbb{N} \times \mathbb{N},$$

and returns

$$\text{an element } m + n \text{ of } \mathbb{N}.$$

The type of this operation is

$$\mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}.$$

But, of course, we may also consider addition for different sets of numbers, giving operations, for example

$$\mathbb{Z} \times \mathbb{Z} \longrightarrow \mathbb{Z} \qquad \mathbb{Q} \times \mathbb{Q} \longrightarrow \mathbb{Q} \qquad \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}.$$

Another general operation sometimes applied to sets is the **disjoint union** but we do not describe this here.

Definition 7. Given a set X, its **powerset** $\mathcal{P}X$, is given by the set of all subsets of X.

We may think of the powerset of X as being given byxs

$$\mathcal{P}X = \{S \mid S \subseteq X\}.$$

All our operations on sets were defined for elements of such a powerset. For example, given an element S of $\mathcal{P}X$, which is nothing but a subset of X,

[27] That is an operation which takes two numbers and returns a number.

we have $X \setminus S$, the complement of S with respect to X, which is another element of $\mathcal{P}X$.

We may think of the union operation as taking two elements of $\mathcal{P}X$, and returning an element of $\mathcal{P}X$, so we would write that as

$$\cup\colon \mathcal{P}X \times \mathcal{P}X \longrightarrow \mathcal{P}X.$$

Because there are so many operations on the powerset it turns out to be a useful model for various situation. In the material on logic we see how to use it as a model for a formal system in logic.

0.3 Functions

One could argue that sets are merely there to allow us to talk about functions, and while this is exaggerated sets wouldn't be much use without the ability to move between them.

0.3.1 Function, source, target, range

A function is a way of turning entities of one kind into those of another. Formally a **function**

$$f\colon S \longrightarrow T$$

is given by

- a **source set** S
- a **target set** T and
- an instruction that turns every element s of S into an element fs of T, often[28] written as

$$s \longmapsto fs.$$

Many people allow giving functions without specifying the source and target sets but this is sloppy. Every function has a type, and for our example f here the type is $S \to T$.

Some instructions can be used with multiple source and target sets. For example

$$x \longmapsto 2x$$

may be used to define a function

- $\mathbb{N} \to \mathbb{N}$,
- $\mathbb{Z} \to \mathbb{Z}$,
- $\mathbb{Q} \to \mathbb{Q}$,
- $\mathbb{R} \to \mathbb{R}$.

and

$$x \longmapsto x^2$$

could have the types (among others)

- $\mathbb{N} \to \mathbb{N}$,
- $\mathbb{Q} \to \mathbb{Q}$ or $\mathbb{Q} \to \mathbb{Q}^+$,
- $\mathbb{R} \to \mathbb{R}$ or $\mathbb{R} \to \mathbb{R}^+$.

[28] It is quite often standard to write $f(s)$ but as long as the argument is not a complicated expression this is unnecessary.

0.3.2 Composition and identity functions

Which functions we can define from one set to another depends on the structure of the sets, and on any known operations on the sets. Only one (somewhat boring) function is guaranteed to exist for every set S, namely the **identity function** id_S given by

$$\begin{aligned} \mathsf{id}_S \colon S &\longrightarrow S \\ s &\longmapsto s. \end{aligned}$$

If we have two functions

$$f \colon S \longrightarrow T \qquad \text{and} \qquad g \colon T \longrightarrow U$$

where the target of f is the source of g we may construct the **composite of f and g**

$$\begin{aligned} g \circ f \colon S &\longrightarrow U \\ a &\longmapsto g(fa). \end{aligned}$$

This works by first applying f to s and then g to the result, that is

$$\begin{array}{ccccc} s & \longmapsto & fs & \longmapsto & gfs \\ S & & T & & U \end{array}$$

Composition allows us to build more complicated functions from simple ones.

Example 0.10. One may think of a linear function on $\mathbb{R}$, of the form

$$x \longmapsto mx + b$$

to be the result of composing the following two functions $\mathbb{R} \longrightarrow \mathbb{R}$':

$$x \longmapsto mx \qquad \text{and} \qquad x \longmapsto x + b \ ,$$

since if we apply the left hand function to x, and the right hand function to the result, we find that x is mapped to $mx + b$.

Example 0.11. We use the idea of a composite a lot, but maybe you find it easier to realize this by looking at the assignment

$$x \longmapsto \sqrt{\sin x} \ .$$

This tells you to first apply the sine function to x, and to apply the square root function to the result. The idea of composing functions just makes this explicit, and it also forces you to ensure that the output of the first function is always a valid input to the second function.

Exercise 7. Define three functions (not forgetting their sources and targets) such that their composite is a function $\mathbb{R} \longrightarrow \mathbb{R}$ which maps an input to the logarithm (for base 2) of the result of adding 2 to the negative of the square of the sine of the input.

It can sometimes be useful to determine which part of the target set is reached by a function. Given a function $f\colon S \longrightarrow T$, for $s \in S$ we say that fs is **the image of s under f**, and

$$\{fs \in T \mid s \in S\}$$

is the **range of f**. It is also known as the **image of the set** S, and written $f[S]$. Note that we may also write this using a property of elements of T as

$$\{t \in T \mid \text{there exists } s \in S \text{ with } fs = t\}.$$

Functions do many useful jobs. For example, if we formally want to define the notion of a **sequence** of, say, real numbers, then we should do so as a function a from $\mathbb{N}$ to $\mathbb{R}$. The nth member of the sequence is given by an. In such cases an is often written a_n. For example, the sequence given on page 20 would have the first few values

argument	0	1	2	3	4
value	1	1/2	1/4	1/8	1/16,

and the formal definition of this function is

$$\begin{aligned} \mathbb{N} &\longrightarrow \mathbb{R} \\ n &\longmapsto \frac{1}{2^n}. \end{aligned}$$

We may also think of a function as translating from one setting to another. In Java *casting* allows us to take an integer, `int` and cast it as a floating point number, `float`. This is effectively a function which takes an `int` (which amounts to a number of bits) and translates it into what we think of as the same number, but now expressed in a different format.

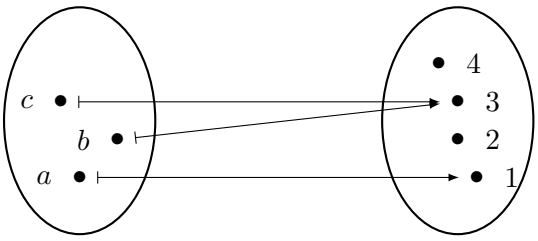
For a mathematical example, we note that we have functions connecting all our sets of numbers since

$\mathbb{N}$ is embedded in $\mathbb{Z}$ which is embedded in $\mathbb{Q}$ which is embedded in $\mathbb{R}$.

All these embeddings are functions, but they are so boring that we don't usually bother to even name them. For a slightly more interesting example take the set of all fractions. From there we have a function that maps a fraction to the corresponding rational number (and so 1/2 and 2/4 are mapped to the same number), allowing us to translate from the presentation as fraction to the numbers we are really interested in.

If you have a customer database you could print a list of all of your customers. You have effectively constructed a function that takes an entry in your database and maps it to the name field. Note that if you have two customers called John Smith then that name will be printed twice, so thinking of a 'set of names' is not entirely appropriate here.

If we have small finite sets then one can define a function in a graphical way, by showing which element of the source set is mapped to which element of the target set.



This function has source set $\{a, b, c\}$ and target set $\{1, 2, 3, 4\}$. It maps a to 1 and b and c to 3. Note that in order for such a diagram to describe a function, every element of the source set must be mapped to precisely one element of the target set.

0.3.3 The graph of a function

It can be useful to think of a function via its graph. The **graph of a function** is the set of pairs consisting of an element of the source set with its image under the function.[29] Given a function

$$f\colon S \longrightarrow T$$

the **graph of** f is the set

$$\{(s, fs) \in S \times T \mid s \in S\}$$

which is a subset of the product of S and T.

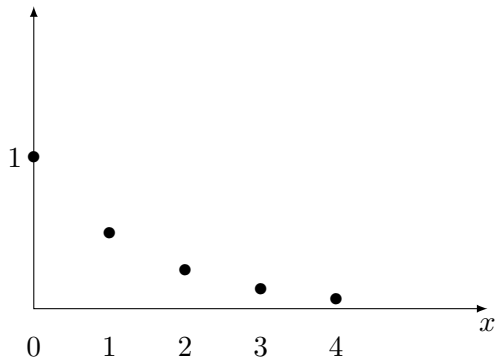
See page 73 for a characterization of those subsets of $S \times T$ which are the graph of a function from S to T.

When we try to picture a function we usually draw its graph. You will have seen these before. Various examples, for functions from $\mathbb{R}$ to $\mathbb{R}$, are given in the following section.

This idea also works for functions between other sets of numbers. Let's return to the example of the function from above, which is given by

$$\begin{aligned} \mathbb{N} &\longrightarrow \mathbb{R} \\ x &\longmapsto \frac{1}{2^x}. \end{aligned}$$

Its graph can be drawn as follows.



0.3.4 Important functions

When we are interesting in judging how long a computer program will take when applied to a large problem we typically count the number of instructions that will have to be carried out. This is typically a function of the size of the problem. For example, sorting a list of n integers will take longer the larger n is.

There are a number of functions that typically appear in such considerations.[30] In computer science it would be sufficient for these purposes to

[29] And indeed, a standard way of defining functions in set theory is via their graphs.

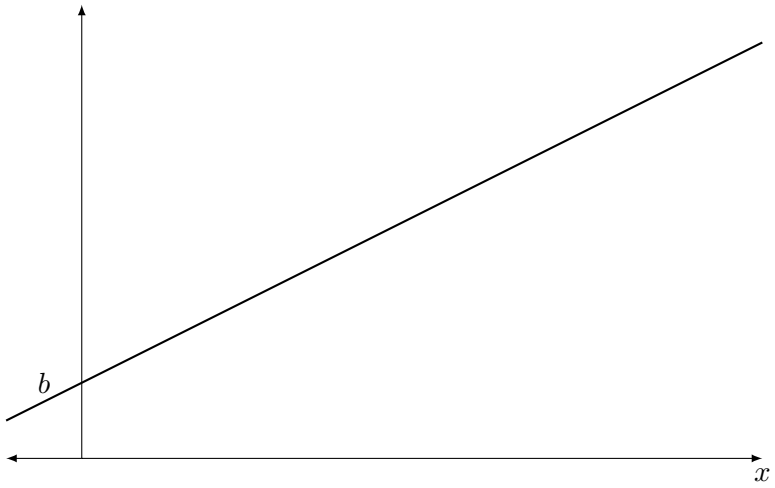
[30] You will meet them again when you look at this in more detail in COMP11212 and COMP26120.

consider these functions as going from $\mathbb{N}$ to $\mathbb{N}$, but it is often more convenient to draw their graphs as functions from $\mathbb{R}^+$ to $\mathbb{R}^+$. In what follows we consider functions that commonly appear in that setting.

There are linear functions, which are of the form

$$\begin{aligned} \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\longmapsto mx + b \end{aligned}$$

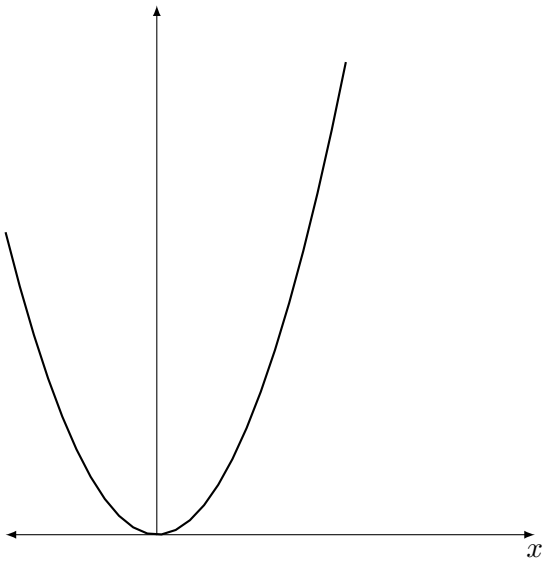
and look like this.



A typical quadratic function is given by

$$\begin{aligned} \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\longmapsto ax^2 + bx + c \end{aligned}$$

and (assuming that a, b and c are all in $\mathbb{R}^+$) looks like this:



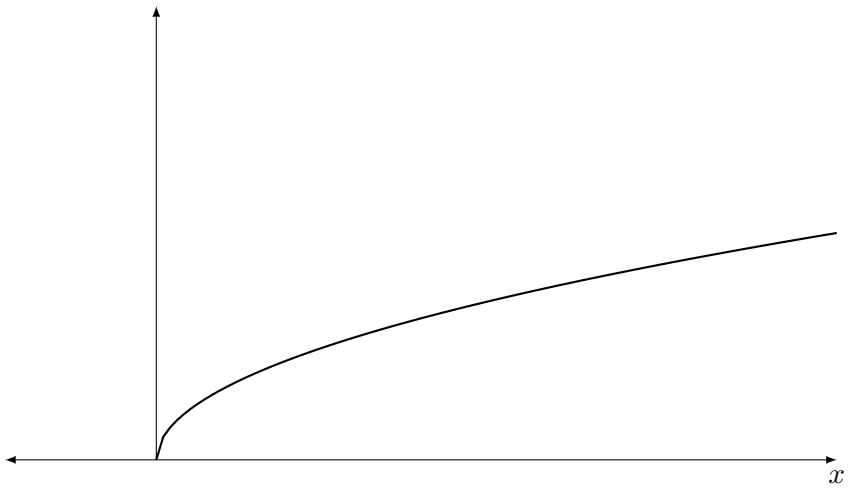
Other polynomial functions, that is functions of the form

$$\sum_{i=1}^{n} a_i x^i$$

may also feature.

Sometimes we wish to consider functions which involve the argument being taken to a power other than a natural number, for example

$$\begin{aligned} \mathbb{R}^+ &\longrightarrow \mathbb{R}^+ \\ x &\longmapsto x^{1/2} = \sqrt{x}. \end{aligned}$$



Some of these functions are defined for non-negative numbers only, so their source is $\mathbb{R}^+$, rather than all of $\mathbb{R}$. Note that for fixed $x \in \mathbb{R}^+$ this function only gives the *positive* solution of the equation

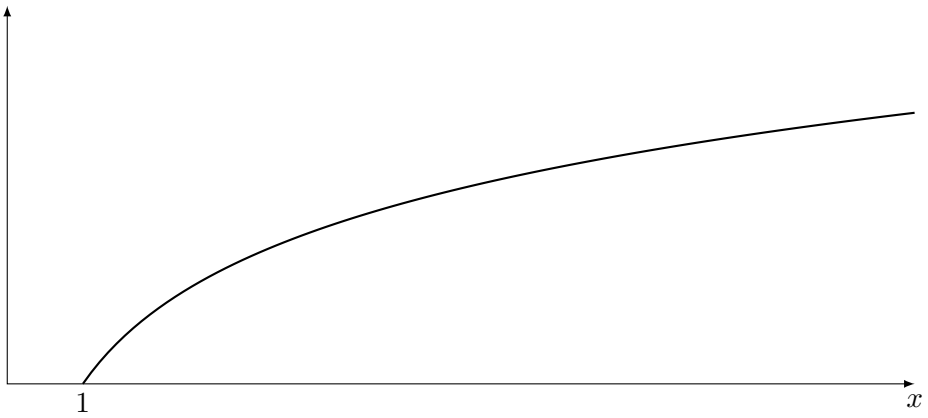
$$y = x^2.$$

If you want to refer to both of these[31] you have to write $\pm\sqrt{x}$.

Apart from these polynomial functions, important examples that come up in computer science are concerned with logarithmic functions. In computer science one typically wishes to use logarithms to base 2. They are typically written as

$$\begin{aligned} [1, \infty) &\longrightarrow \mathbb{R}^+ \\ x &\longmapsto \log x \end{aligned}$$

and look like this.

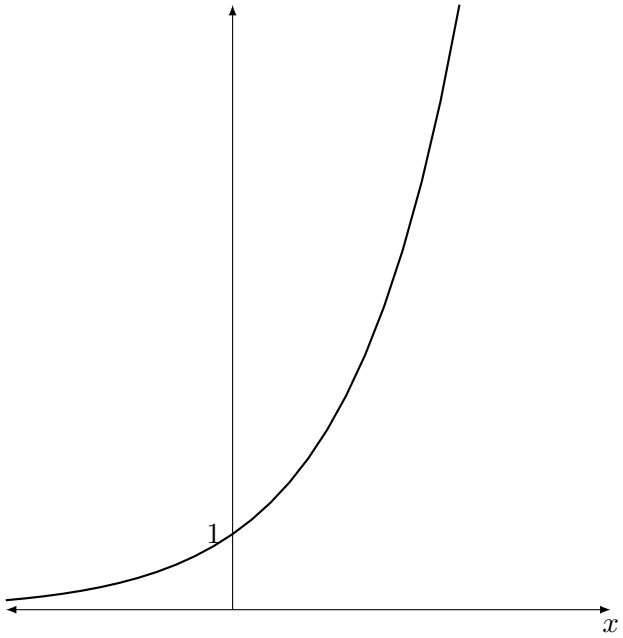


[31] If you have been taught otherwise then this is at odds with notation used at university level and beyond.

And then there are exponential functions. Because of the speed with which these grow having a program whose number of instructions is exponential in the size of the problem is a serious issue since it means that it is not feasible to calculate solutions for larger problem sizes using this program. It is fairly usual to use 2 as a base once again. The function in question is

$$\begin{aligned} \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\longmapsto 2^x \end{aligned}$$

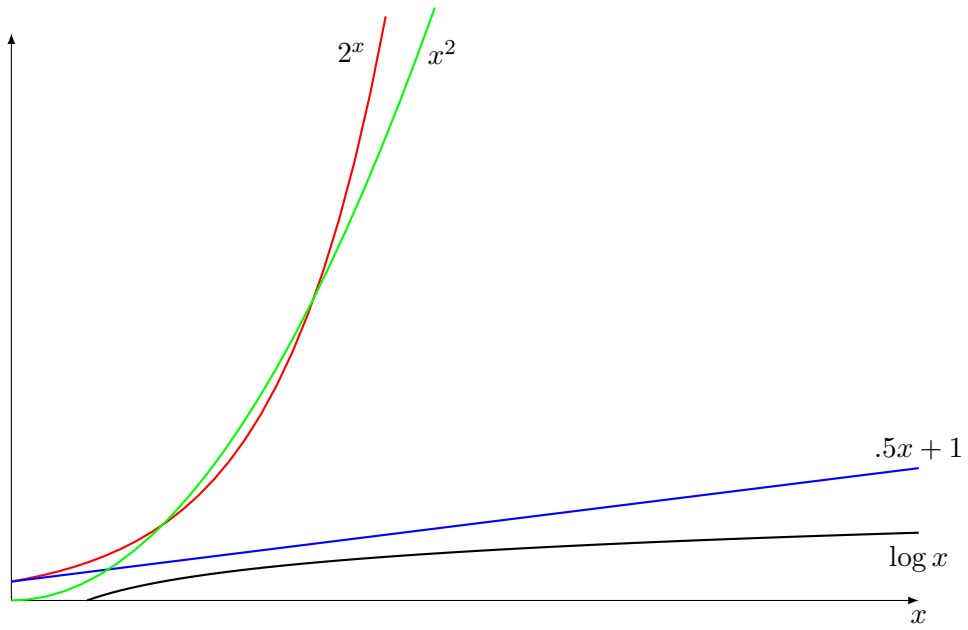
and its graph is even steeper than that of the quadratic curve above.



In all these cases typically the shape of the curve is more important than any parameters involved in defining it—so knowing that we have a quadratic function is very useful, whereas there is little added benefit in knowing a, b and c in $ax^2 + bx + c$.

If one has a problem size of 1,000,000, for example, then it is important to know how fast the function grows to see how many instructions will have to be carried out for that size (and so how long it will take for the program to finish, or it if is possible for this program to finish at all).

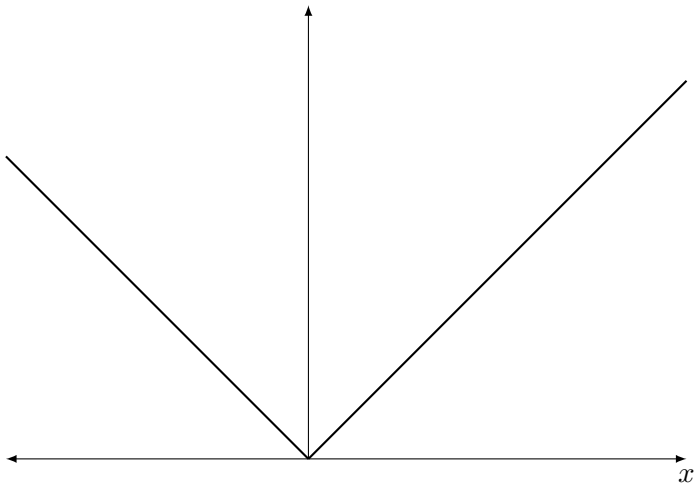
If we draw the functions from above in the same grid we can compare them.



0.4 Constructions for functions

An important way of constructing new functions from old ones is what is known as **definition by cases**. What this means is that one pieces together different functions to give a new one.

Example 0.12. Assume we want to give a proper definition of the 'absolute' function $|\cdot| \colon \mathbb{R} \longrightarrow \mathbb{R}^+$ for real numbers. The value it returns depends on whether the input is negative, or not. The graph of this function is depicted here.



We can write the corresponding assignment as

$$x \longmapsto \begin{cases} -x & x < 0 \\ x & \text{else.} \end{cases}$$

Example 0.13. If you want to give an alternative description of the function

$$\begin{aligned} \mathbb{Z} &\longrightarrow \mathbb{Z} \\ n &\longmapsto n \bmod 2, \end{aligned}$$

which maps even numbers to 0, and odd numbers to 1, you could instead write

$$x \longmapsto \begin{cases} 0 & x \bmod 2 = 0 \\ 1 & \text{else} \end{cases}$$

or, if you don't want to put the mod function into the definition, you could write

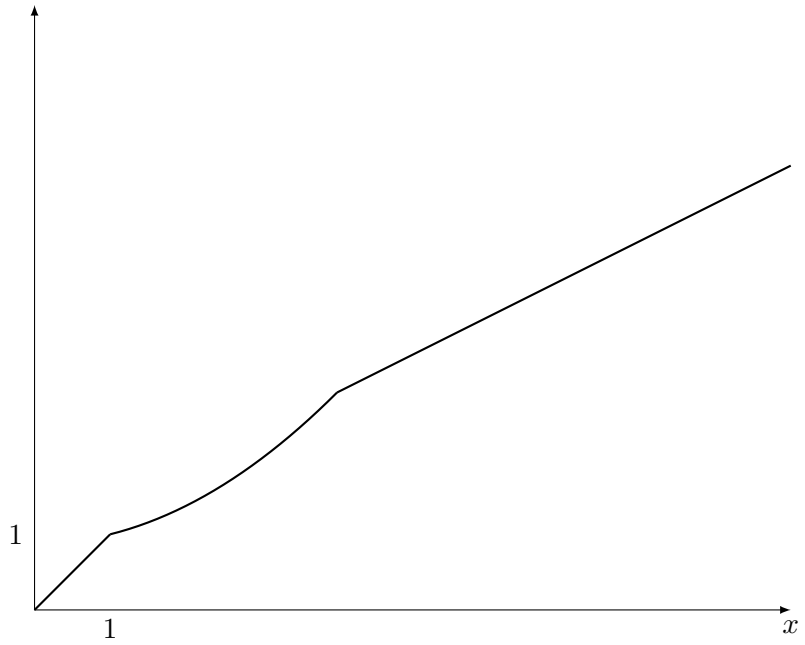
$$x \longmapsto \begin{cases} 0 & x \text{ even} \\ 1 & \text{else.} \end{cases}$$

What is important is that

- you give a value for each element of the source and
- you don't give more than one value for any element of the source.

In other words, on the right you must split your source sit into disjoint parts, and say what the function does for each of those parts.

Example 0.14. You might need this when you are trying to describe the behaviour of an entity which changes. For example, assume you are given the following graph:



This function $\mathbb{R}^+ \longrightarrow \mathbb{R}^+$ is given by the assignment

$$x \longmapsto \begin{cases} x & x \in [0,1] \\ \frac{1}{8}x^2 + \frac{7}{8} & x \in (1,4] \\ \frac{1}{2}x + \frac{7}{8} & \text{else.} \end{cases}$$

Exercise 8. Write down formal definitions of the following functions.

(a) The function which takes two integers and returns the negative of their product.

(b) The function from $\mathbb{R} \times \mathbb{R}$ to $\mathbb{R}$ which returns its first argument.

(c) The function from $\mathbb{Z} \times \mathbb{Z}$ to $\{0, 1\}$ which is equal to 1 if and only if both arguments are even.

(d) The function from $\mathbb{R}$ to $\mathbb{R}$ which behaves like the sine function for negative arguments, and like the power of 2 for non-negative arguments.

(e) Draw a picture of the set

$$\{Abdul, Bella, Clara, Dong\} \times \{red, blue, green\}.$$

Define a function that from that set to $\{0, 1\}$ which is 1 if and only if its first argument has more letters than its second.

Apart from this, the constructions we have for sets are also meaningful for functions.

If we have functions $f\colon S \longrightarrow S'$ and $g\colon T \longrightarrow T'$. Then we can define a function

$$S \times T \to S' \times T',$$

which we refer to as

$$f \times g$$

by setting

$$(s, t) \longmapsto (fs, gt).$$

Optional Exercise 3. Can you think of something that would allow you to extend the powerset construction to functions?

The following exercises draws on functions, as well as on the definition of the powerset from the previous section.

Exercise 9. Given a set X, define the following functions. Don't forget to write down their source and target.

(a) A function f from X to its powerset with the property that for every $x \in X$ we have $x \in fx$.

(b) A function from the product of the powerset of X to itself, to the powerset of X, with the property that a pair of sets is mapped to the set consisting of all those elements of x which is either in the first set, or in the second set, but not in both.

(c) Define a function from the product of X with its powerset to the set $\{0, 1\}$ which returns 1 if and only if the first component of the argument is an element of the second component.

(d) Define a function from the powerset of $\mathbb{N}$ to $\mathbb{N}$ which adds up all the elemnts in the given set.

Chapter 1

Complex Numbers

The real numbers allow us to solve many equations, but the typical example of something that has no solution is

$$x^2 = -1.$$

One way of looking at the complex numbers is that they remedy this problem. But assuming this is all they do would sell them far short.

Note that in order to solve exercises in this chapter you should only use properties given by Facts 1 to 7 in Chapter 0.

1.1 Basic definitions

We begin by giving some basic definitions.

Definition 8. The **set of complex numbers** $\mathbb{C}$ consists of numbers of the form

$$a + bi,$$

where a, b in $\mathbb{R}$. Here a is known as the **real part** and b as the **imaginary part** of the number.

At first sight it is not entirely clear what exactly we have just defined. One may view $a + bi$ as an expression in a new language.

As usual, if one of a or b is 0 it is customary not to write it, so the complex number a is equal to $a + 0i$ and the complex number bi is equal to $0 + bi$. Similarly, if $b = 1$ then it is customary to write $a + i$ instead of $a + 1i$.

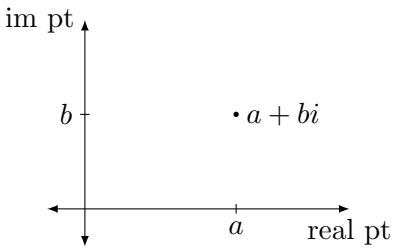
We may think[1] of a real number r as being a complex number whose imaginary part is 0, so it has the form $r + 0i$. In that way the complex numbers can be thought to include the real numbers (just as we like to think of the real numbers as including the rational numbers).

This gives a function from $\mathbb{R}$ to $\mathbb{C}$ defined by

$$r \longmapsto r + 0i.$$

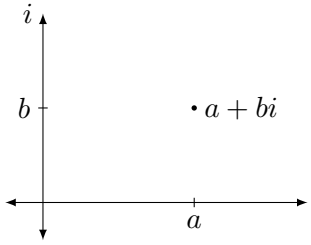
[1] Compare this with casting a value of one datatype to another in Java.

Complex numbers are usually drawn as points within the plane, using the horizontal axis for the real and the vertical axis for the imaginary part.



Above we have added labels for orientation, but usually this is done a bit differently.

Instead of marking the real and the imaginary part on the axes it is more common to mark the 'imaginary axis' with i, giving a picture in the *complex plane*.



1.2 Operations

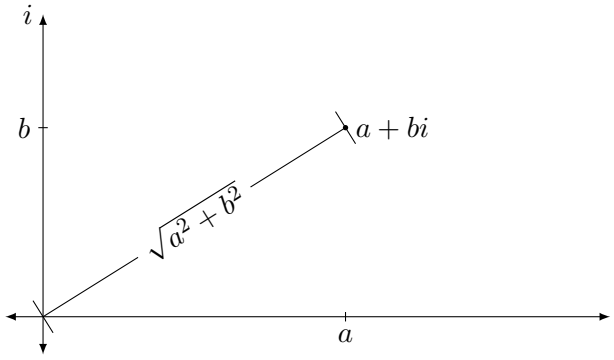
There are quite a few operations one defines for complex numbers.

1.2.1 The absolute

The[2] **absolute** $|a + bi|$ **of a complex number** $a + bi$ is given by

$$\sqrt{a^2 + b^2}.$$

We may think of this as the length of the line that connects the point 0 with the point $a + bi$:



Note that this extends the notion of absolute for real numbers in the sense that

$$|a + 0irvert = \sqrt{a^2 + 0} = \sqrt{a^2} = |a|$$

where we use the absolute function for real numbers on the right.[3]

One can calculate with the complex numbers based on the following operations.

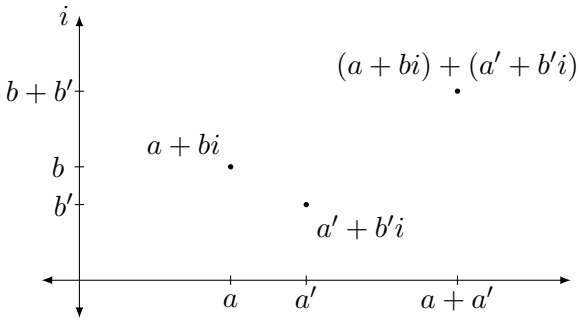
[2] This is also known as the *modulus* of a complex number.

[3] And indeed note that we could use $\sqrt{r^2}$ as a definition of the absolute for a real number r.

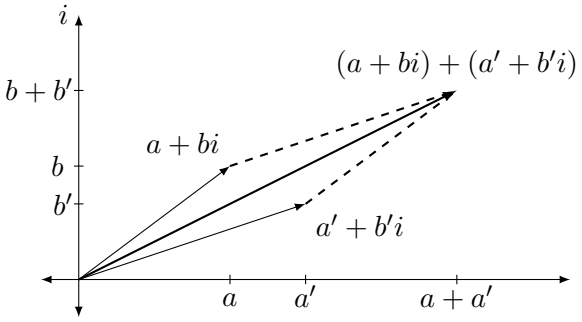
1.2.2 Addition

We set

$$(a+bi)+(a'+b'i) = (a+a')+(b+b')i.$$



To understand addition it is useful to think of the numbers in the complex plane as[4] *vectors*, then addition is just the same as the addition of vectors:



If you prefer, you may think of this as taking the vector for $a'+b'i$ and shifting it so that its origin coincides with the end point of the vector for $a+bi$.[5]

We note that if we have two complex numbers whose imaginary part is 0, say a and a', then their sum as complex numbers is $a+a'$, that is their sum as real numbers.

Note that 0 is the unit for addition[6], that is adding 0 to a complex number (on either side) has no effect.[7] In other words we have

$$\begin{aligned} 0+(a+bi) &= (0+a)+(0+b)i && \text{def addition} \\ &= a+bi && \text{addition for reals} \\ &= (a+0)+(b+0)i && \text{addition for reals} \\ &= (a+bi)+0. && \text{def addition} \end{aligned}$$

So what about inverses for addition? For the real numbers every element r has an *inverse* for addition in the form of $-r$: It is a number[8] which,[9] if added to r on either side, gives the unit for addition 0. We may use the inverse for addition for the reals to define an inverse for addition for complex numbers by setting

$$-(a+bi) = -a+(-b)i = -a-bi.$$

[4]Vectors will be taught in detail in the second half of Semester 2.

[5]Note that you may just as well think of shifting the vector for $a+bi$ such that its origin coincides with the end point of the vector for $a'+b'i$.

[6]Look at the unit for addition given by Facts 1, 3, 5 and 6.

[7]For a formal definition of the unit of an operation see 12.

[8]Again, compare Facts 3. 5 and 6 from the previous chapter.

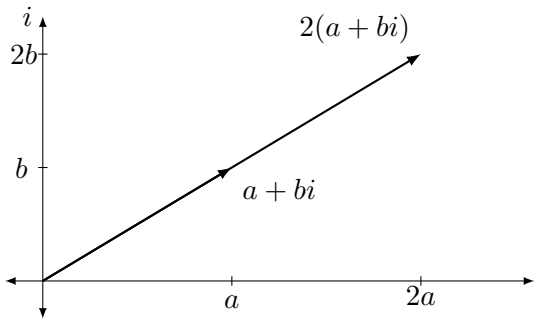
[9]For a formal definition of the inverse for a given element with respect to a given operation see 13 in the following chapter.

Exercise 10. Check that $-(a+bi)$ is indeed the inverse with respect to addition of $a+bi$. *Hint: The paragraph above this exercise tells you what you need to check.*

Note that there is no easy connection between the absolute and addition; the best we may establish for complex numbers z and z' is that

$$|z+z'| \leq |z| + |z'|.$$

Note that $(a+bi)+(a+bi) = 2a+2bi$, and that we may think of this as stretching $a+bi$ to twice its original length, and write it as $2(a+bi)$:



In general, given a real number r and a complex number $a+bi$ we may define

$$r(a+bi) = ra + rbi.$$

Exercise 11. Draw the following numbers in the complex plane: 2, -2, $2i$, $-2i$, $3+i$, $-(3+4i)$, $(-1+2i)+(3+i)$, $(1+2i)+(3-i)$, $(1+2i)-(3+i)$. For each quadrant pick one of these numbers (you may pick at most two numbers lying on an axis, and they have to be on different ones), and calculate its absolute. Describe in general how to draw the following, given a complex number z:

(a) $-z$,

(b) $2z$,

(c) $3z$,

(d) rz, where r is an arbitrary real number.

Exercise 12. Consider the function f from $\mathbb{R}^2$ to $\mathbb{C}$ which is defined as follows:

$$(a,b) \longmapsto a+bi.$$

Show that $f(a,b) + f(a',b') = f((a,b)+(a',b'))$ for all $(a,b),(a',b') \in \mathbb{R}^2$.

1.2.3 Multiplication

We define a multiplication operation on complex numbers by setting

$$(a+bi)(a'+b'i) = aa' - bb' + (ab'+ba')i.$$

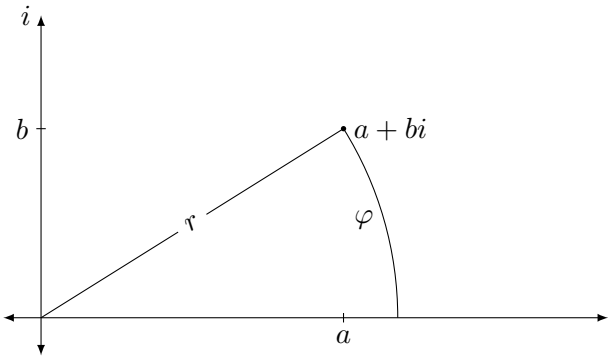
Exercise 13. Show that 1 is the unit for multiplication. *Hint: Check the calculation carried out above which shows that 0 is the unit for addition. Also look at Fact 1 which tells you what it means for 1 to be the unit for multiplication of natural numbers.*

Note that if one of the numbers has imaginary part 0 then we retain the multiplication with a real number defined above, that is

$$a(a' + b'i) = aa' + ab'i.$$

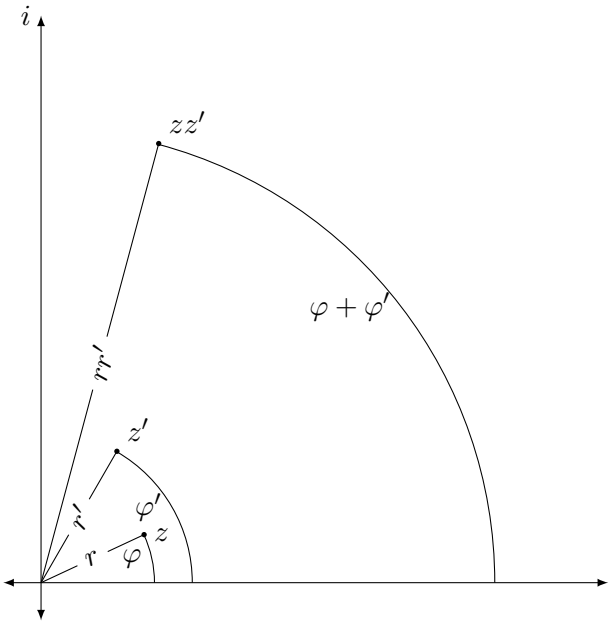
There is a geometric interpretation of multiplication, but it is a bit more complicated than that for addition. We here only give a sketch of this.

Instead of giving the coordinates a and b to describe a point in the complex plane one also could give an **angle** and a **length**—this is known as describing a number using **polar coordinates**.



This angle is often called the **argument** of the complex number, and the length is nothing but the absolute from above.

Knowing r and φ in this picture gives precisely as much information as having a and b. Describing multiplication is much easier when we do it with respect to these polar coordinates: the product of numbers z and z' given by φ and r and φ' and r' respectively has the absolute rr' and the argument $\varphi + \varphi'$.



To move from polar coordinates to the standard form there is a simple formula: The complex number given by r and φ is

$$r(\cos\varphi + \sin\varphi i).$$

In the other direction one can use the arctangent function arctan, the partial inverse of the tangent function, to calculate φ given a and b, but a few case distinctions are required.

Optional Exercise 4. Write out the definition of the function that gives the corresponding angle, that is the argument, for a complex number $a+bi$. Then prove that, starting from a complex number, calculating the angle and the absolute, and then calculating the real and imaginary part from the result, gives back the number one started with. Show that with these calculations it is the case that the argument of zz' is the argument of z plus the argument of z'.

Also note that there is a nice connection between the absolute and multiplication since we have for complex number z and z' that

$$|zz'| = |z||z'|.$$

Exercise 14. Prove the preceding statement, that is $|zz'| = |z||z'|$.

We note that according to this definition we have

$$ii = (0+1i)(0+1i) = 0 - 1\cdot 1 + (0\cdot 1 + 1\cdot 0)i = -1,$$

so in the complex numbers we may solve equations that are not solvable in $\mathbb{R}$.

Exercise 15. Pick four numbers in at least three different quadrants of the complex plane. Calculate, and then draw, their product with the number i. Describe in general where to draw iz given the position of z.

Optional Exercise 5. What happens if we keep multiplying i with itself? What does that tell you about solutions to the equation $x^4 = 1$? What about solutions for $x^n = 1$ more generally?

Exercise 16. Consider the function f from $\mathbb{R}^2$ to $\mathbb{C}$ which is defined as follows:

$$(a,b) \longmapsto a + bi.$$

Define addition and multiplication on $\mathbb{R}^2$ based on these operations for complex numbers. *Hint: You may want to consult Exercise 12.*

We have seen that with regards to subtraction, every complex number z has an inverse in the form of $-z$. What about multiplication?

Let $a+bi$ be a complex number other than 0, that is at least one of a and b is non-zero. Then[10] its multiplicative inverse is given by

$$\frac{a}{a^2+b^2} - \frac{b}{a^2+b^2}i.$$

[10]Note that our condition means that $a^2 + b^2 \neq 0$ and therefore we may form the fractions given here.

Sometimes the notation

$$(a + bi)^{-1}$$

is used for this number. More generally, if we have a complex number z then it's inverse, if it exists, is written as z^{-1}. The expression

$$z/z'$$

is again a shortcut for

$$z(z')^{-1},$$

and when you write that you need to make sure that z' actually has an inverse.

Recall that we avoid talking about division as an operation on this unit, so if you want to remove a factor from an equation please try to talk about multiplying with the multiplicative inverse, and think about whether this exists!

Exercise 17. Show that the number given above,

$$\frac{a}{a^2 + b^2} - \frac{b}{a^2 + b^2}i,$$

is indeed the inverse of $a + bi$ with respect to multiplication. *Hint: Check Fact 5 from the previous chapter to see what you have to prove.*

Calculate the inverse of the complex number $a = a + 0i$. How does that compare with the inverse for a when viewed as an element of $\mathbb{R}$?

Exercise 18. Assume you are given a complex number by its absolute and argument. Give its multiplicative inverse using the same description (that is, absolute and argument).

In summary we have seen that just as for the real numbers, we may define addition and composition for complex numbers, and in such a way that if we treat real numbers as particular complex ones, then the operations agree. Indeed, it is also possible to define exponentiation and logarithms for complex numbers but this idea leads us too far afield.

1.2.4 Conjugation

There is a further operation that you may find in texts that deal with complex numbers, namely **conjugation**. The **conjugate** $\overline{z}$ of a complex number $z = a + bi$ is given by $a - bi$.

Exercise 19. Express the conjugate using the absolute and argument of a complex number. In other words, if you have a complex number given by its absolute r and its argument ϕ, what are the absolute and argument of its conjugate? *Hint: If you find this difficult draw a few examples in the complex plane.*

Exercise 20. Show that $z\overline{z} = |z|^2$.

1.3 Properties

The complex numbers have various properties which make them a nice collection of numbers to work with. You are allowed to use the following in subsequent exercises on complex numbers.

Fact 8. *Addition and multiplication of the complex numbers have all the properties of the real numbers as given in Fact 6.*

Note that when it comes to solving equations, the complex numbers are even better behaved than the real ones: Every polynomial equation, that is an equation of the form

$$\sum_{i=0}^{n} c_i x^i = c_n x^n + c_{n-1} x^{n-1} + \cdots + c_1 x + c_0 = 0,$$

where the c_i are complex numbers has at least one solution[11] in $\mathbb{C}$, whereas this is not true in $\mathbb{R}$ even if the c_i are all elements of $\mathbb{R}$. This means that the complex numbers are particularly suitable for various constructions that depend on having solutions to polynomials.

In analysis, which includes the study of functions, their derivatives and their integrals, the theory of functions of complex numbers is much smoother than that of real ones. In order to calculate various (improper) integrals for functions of real variables one may apply methods that require functions of complex variables.

The fact that complex numbers may be thought of as having two parts, and that we have various operations for these, means they are particularly suited to a number of application areas where these operations may be interpreted.

1.4 Applications

Complex numbers may appear artificial, and having numbers with an 'imaginary' part may suggest that these are merely figments of some mathematicians' imagination.

But it turns out that they are not merely some artefact whose only use it is to deliver a number which is a square root of -1. Because complex numbers can effectively be thought of as vectors, but vectors which allow multiplication (and division) as well as addition, they are very useful when it comes to talking about quantities that need a more complex structure to express them than just one number.

In physics and various areas of engineering quantities which may be described by just one number are known as 'scalars'. Examples are distance, speed (although to describe movement one might want to include direction with speed) and energy. In a direct-current circuit, voltage, resistance and current are treated as scalars without problem. In alternating current circuits, however, there are notions of frequency and phase shift which have to be taken into account, and it turns out that using complex numbers to describe such circuits results in a very useful depiction. Moreover some calculations become much simpler when one exploits the possibilities given by modelling the circuit with complex numbers.

[11] In fact, one can show that there are n solutions, but this requires counting some solutions more than once.

Signal analysis is another area where complex numbers are often employed. Again the issue here are periodically varying quantities. Instead of describing these using a sine or cosine function of some real variable, employing the extensions of these functions to the complex numbers makes it possible to describe the amplitude and phase at the same time. In the second year course on Mobile Systems, COMP28512, these ideas are used.

There you will see for example, that by using complex numbers for a Fourier transform calculations that look complicated can be carried out via matrix multiplication.[12] When you meet this material you should remind yourself of what you know about complex numbers from these notes.

There are other areas where applications arise, such as fluid dynamics, control theory and quantum mechanics.

[12]The latter will be treated in Part 4 of this course unit.

Chapter 2

Statements and Proofs

Mathematics is a discipline that relies on rigorous definitions and formal proofs. As a consequence in mathematics statements hold, or they do not (but we may not know which it is).[1] This is very different from the situation in the natural sciences, for example. Here a theory may be falsified by observations that contradict it, but there is no way of formally verifying it.

How does a system that seeks to provide such certainty work? In principle the thought is that it is possible to define a theory strictly from first principles (typically starting with a formal theory of sets), with rules for deriving statements from existing ones. Such a system is very rigid and syntactic[2] in nature, much like a computer language (and indeed there are computer programs that implement at least aspects of this). Statements that may be formally derived in the system are known as theorems. In principle it should be possible to fit all of mathematics into a formal system like this.[3]

But in practice this is not what mathematicians do. There are two reasons for this. Starting from first principles it takes a very long time to build up the apparatus required to get to where one may even talk about entities such as the real numbers with complete rigour. Secondly the resulting statements are very unwieldy and not human-readable. Hence mathematicians carry out their work in some kind of *meta-language* which in principle can be translated into a formal system. Increasingly there are computer-verified proofs in various areas, in particular in theoretical computer science.

In this and the following chapter of the notes we look at both these ideas—proofs as they are customarily carried out by mathematicians and a formal system.

2.1 Motivation

You are here to study computer science rather than mathematics, so why should you worry about proving statements? There are two reasons one might give here.

For one there is the area of *theoretical computer science* which arguably is also an area in mathematics. The aims of this part of computer science

[1] There are also issues to do with whether a given formal system allows us to construct a proof or a counterexample.

[2] This means concerned with symbols put together according to some rules without any concern what they might mean.

[3] But there is a famous result by the logician Kurt Gödel, his *Incompleteness Theorem*, which tells us that any system sufficiently powerful for most of mathematics cannot prove its own consistency.

are to make formal statements and to prove them. Here are some examples of the kind of statement that are of concern in this area.

- This abstract computational device has the same computational power as another.
- This computation is equivalent to another.[4]
- This abstract computational system behaves in a particular manner over time.
- This problem cannot be solved by a computer, or, equivalently, there is no algorithm (or decision procedure) for it (see COMP11212).
- The best possible algorithm for this problem requires a number of steps that is a quadratic function in the size of the problem (see COMP26120).
- This program will terminate and after it has done so its result will satisfy a particular condition.
- This circuit implements a particular specification.

You can see that while the first few statements sound fairly abstract the latter two look as if they might be closer to real-world applications.

Secondly, under certain circumstances it is important to make absolute statements about the behaviour of a computational device (a chip or a computer program for example). Formally proving that programs behave in a particular way is labour-intensive (and creating a formal model of the real world in which the device lives is potentially error-prone).

In safety-critical systems, however, the benefits are usually thought to outweigh the cost. For example in an aircraft it is vital that the on-board computer behaves in a particular way. Emergency course corrections have to be made promptly and correctly or the result may be fatal for those on board. When NASA sends an explorer onto Mars, or the Voyager space craft to fly through the solar system (and to eventually leave it) then it is vital that a number of computer-controlled manoeuvres are correctly implemented. Losing such a craft, or rendering it incapable of sending back the desired data, costs large amounts of money and results in a major setback.

But even outside such applications computing is full of statements that are at least in part mathematical. Here are some examples.

- The worst case complexity of this algorithm is $n \log n$ (the kind of statement that you will see in COMP26120).
- This recursive procedure leads to exponential blow-up.
- A simple classification rule is to choose the class with the highest posterior probability (from COMP14112).
- Time-domain samples can be converted to frequency domain using Fourier Transforms, which are a standard way of representing complex signal $g(t)$ as a linear sum of basic functions $f_g(t)$) (from COMP28512).

The aim of this course unit is to prepare you for both these: studying areas of theoretical computer science and making sense of mathematical statements that appear in other parts of the field.

[4]What it might mean for two computations to be equivalent is a whole branch of theoretical computer science.

2.2 Precision

Something the language of mathematics gives us is precision. You need to become familiar with some aspects of this. In particular, there are some key phrases which sound as if they might be parts of every-day language, which have a precise meaning in a mathematical context.

2.2.1 Key phrases

Vocabulary that helps us with this are phrases such as

- 'and',
- 'or',
- 'implies' (and the related 'if and only if'),
- 'there exists' and
- 'for all'.

The aim of this section is to introduce you to what these phrases mean, and how that is reflected by by proving statements involving them.

And. Formally we use this word to connect several statements, or properties, and we demand that all of them hold.

When you enter several words into the Google search box you ask it to return pages which contain *all* the listed words—you are demanding pages that contain *word* 1 and *word* 2 and ...

If you are running database queries you are often interested in all entries that combine several characteristics, for example, you might want all your customers from a particular country for whom you have an email address so that you can make a special offer to them.

These are all informal usages, but they have fundamentally the same meaning as more mathematical ones. A very simple example is the definition of the intersection of two subsets S and T of a set X.

$$S \cap T = \{x \in X \mid x \in S \text{ and } x \in T\}.$$

In order to prove that an element x is in this intersection we have to prove both, that

- x is in S and that
- s is in T.

It is a good idea to structure proofs so that it is clear that these steps are carried out. Here is an example involving natural numbers. If you are unfamiliar with the notation consult Section 0.1.1. To show that 6 is an element of

$$\begin{aligned}&\{n \in \mathbb{N} \mid n \bmod 2 = 0 \text{ and } n \bmod 3 = 0\}\\ &= \{n \in \mathbb{N} \mid n \bmod 2 = 0\} \cap \{n \in \mathbb{N} \mid n \bmod 3 = 0\}\end{aligned}$$

one splits the requirement.

- Since $6 = 3 \cdot 2$ we have that 6 is a multiple of 2 and so $6 \bmod 2 = 0$ and

- since $6 = 2 \cdot 3$ we have that 6 is a multiple of 3 and so $6 \bmod 3 = 0$.

Overall this means that 6 is in the intersection as required.

This usage of 'and' may also be observed in every-day language: If I state 'it is cloudy and it is raining' then I am claiming that both of the following statements are true:

- It is cloudy.
- It is raining.

In order to argue that a statement of the form (Clause 1 and Clause 2) does not hold it is sufficient to show that one of the two clauses fails to hold. So for example in order to establish

$$x \notin S \cap T$$

it is sufficient to show *one* of

- $x \notin S$,
- $x \notin T$.

Or. We connect two statements or properties with 'or' if at least one (but possibly both of them) hold.

To use the Google search box as an example once again, if you type two entries separated by 'OR' it will look for pages which contain one of the two words.

This is also a fairly standard database query: You might be interested in all the customers for whom you have a landline or a mobile phone number, or all the ones who have ordered product X or product Y because you have an accessory to offer to them.

Again a simple example is given by sets, namely by the definition of the union of two subsets S and T of a set X, which is given by

$$\{x \in X \mid x \in S \text{ or } x \in T\}.$$

In order to show that 6 is an element of

$$\{n \in \mathbb{N} \mid n \bmod 2 = 0 \text{ or } n \bmod 3 = 0\}$$

it is sufficient to prove *one* of the two parts. It is therefore sufficient to state that

- Since $6 = 3 \cdot 2$ we have that 6 is a multiple of 2 and so $6 \bmod 2 = 0$.

It is not necessary to check the other clause. Nonetheless we do get a proof strategy here: Look at both cases separately, and stop when one of them has been established.

Again this usage is well established in informal language (although usage tends to be less strict than with 'and'). If I say 'Tomorrow I will go for a walk or a bicycle ride' then I expect one of the following two sentences to be true:

- Tomorrow I will go for a walk.
- Tomorrow I will go for a bicycle ride.

Note that in informal language 'or' often connects incompatible statements, which means that also at most one of them is true. However, in the example it is possible that I might fit in both, a walk and a bike ride and the statement does not preclude this.

In order to show that a statement of the form (Clause 1 or Clause 2) does not hold we have to show that *neither* of the clauses holds. For example, in order to show

$$x \notin S \cup T$$

we have to establish *both*.

- $x \notin S$,
- $x \notin T$.

Implies. This is a phrase that tells us that if the first statement holds then so does the second (but if the first statement fails to hold we cannot infer anything about the second).

For this notion it is harder to find examples outside of mathematics, but you might want to ensure that in your database, the existence of an address entry for a customer implies the existence of a post code. Again we may look at sets to give us a simple formal example.

If S and T are subsets of some set X then

$$S \subseteq T$$

means that given $x \in X$,

$$x \in S \text{ implies } x \in T.$$

In order to establish that $S \subseteq T$ given $x \in X$ one only has to show something in case that $x \in S$, so usually proofs of that kind are given by assuming that $x \in S$, and then establishing that $x \in T$ also holds.

To show, for example, that

$$\{n \in \mathbb{N} \mid n \bmod 6 = 0\} \subseteq \{n \in \mathbb{N} \mid n \bmod 3 = 0\}$$

we may pick n in the former set. We know that $n \bmod 6 = 0$, so n is divisible by 6 which means that there is

$$k \in \mathbb{N} \qquad \text{with} \qquad n = k6.$$

But that means that

$$n = k6 = k(2 \cdot 3) = (k2)3$$

and so $n \bmod 3 = 0$.

Typically we do not use 'implies' in informal language, but we have constructions that have a similar meaning. I might state, for example, 'if it rains I will stay at home'. So if on the day it is raining you should not expect me to meet you, you should expect me to be at home. Note that this does *not* allow you to draw any conclusions in the case where it is not raining, although many people tend to do so. ('But you said you wouldn't be coming only if it was raining ...'.) If I say 'if it is raining I will definitely stay at home' I've made it clearer that I reserve the right to stay at home even if it is not raining. Note that in the formal usage of 'implies' the meaning is completely precise.

In order to show that a claimed implication does not hold one has to find an instance where the first clause holds while the second does not. So in order to catch me out as having said an untruth in the above example, you've got to find me out of the house when it is raining at the appointed time. To look at a more mathematical example: In order to refute the claim that, for a natural number n, n being divisible by 2 implies n being divisible by 6, you need to find a number that is

- even, that is divisible by 2 and
- not divisible by 6.

The number 4 satisfies both properties.

If and only if. This phrase is merely a short-cut. When we say that

Statement 1 (holds) if and only if Statement 2 (holds)

then we mean by this that both,

Statement 1 implies Statement 2

and

Statement 2 implies Statement 1.

An example is given by the prove of equality of two sets, which is equivalent to showing that they are both subsets of each other. To show that

$$S = \{n \in \mathbb{N} \mid n \bmod 6 = 0\}$$

is equal to

$$T = \{n \in \mathbb{N} \mid n \bmod 2 = 0\} \cap \{n \in \mathbb{N} \mid n \bmod 3 = 0\}$$

we have to show that $S \subseteq T$ and that $T \subseteq S$.

It is a good idea to optically structure the proof accordingly.

$S \subseteq T$ Given $n \in S$ we know that $n \bmod 6 = 0$ which means that we can find $k \in \mathbb{N}$ with $n = k6$. This means that both

- $n = (k3)2$ and so $n \bmod 2 = 0$ and
- $n = (k2)3$ and so $n \bmod 3 = 0$

and so $n \in T$.

$T \subseteq S$ Given $n \in T$ we know that both

- $n \bmod 2 = 0$, which means that there is $k \in \mathbb{N}$ with $n = k2$ and
- $n \bmod 3 = 0$, which means that there is $l \in \mathbb{N}$ with $n = l3$.

This means that 2, which is a prime number, divides $n = l3$. By Definition 9 this means that

- 2 divides 3 (which clearly does not hold) or
- 2 divides l, which means that there exists $j \in \mathbb{N}$ with $l = j2$.

Altogether this means that $n = l3 = j2 \cdot 3 = j6$, and so n is divisible by 6.

Quite often when proving an 'if and only if' statement the best strategy is to prove the two directions separately. The only exception is when one can find steps that turn one side into the other, and every single step is reversible.

In order to show that an if and only if statement does not hold it is sufficient to establish that one of the two implications fails to hold.

For all. Again a phrase that is very common in mathematical definitions or arguments, but there are other uses. For example you might want to ensure that you have an email address for every customer in your database.

A use in a formal setting might be

$$\text{for all elements } k \text{ of } \{4n \mid n \in \mathbb{N}\} \text{ we have that } k \text{ is divisible by 2.}$$

In order to show that this is true I have to assume that I have an arbitrary element k of the given set. In order for k to be in that set it must be the case that there exists $n \in \mathbb{N}$ such that $k = 4n$. But now

$$k = 4n = 2(2n)$$

and so we have provided a witness, namely $2n$, to show that k is divisible by 2.

Such a statement should have two parts.

- For which elements are making the claim? There should be a set associated with this part of the statement (this is $\mathbb{N}$ in the above example).
- What property or properties do these elements have to satisfy? There should be a statement which specifies this (which is the remainder of the above formal statement, which is quite complex in its own right).

Looking back above at the statement that one set is the subset of another, we have, strictly speaking, suppressed a 'for all' statement.

Given subsets S and T of a set X the statement

$$S \subseteq T$$

is equivalent to

$$\text{for all } x \in X \qquad x \in S \text{ implies } x \in T.$$

Typically when proving a statement beginning with 'for all' one assumes that one has an unspecified element of the given set, and then establishes the desired property.

A nice example of such a statement is that of the equality of two functions with the same source and target. Let $f\colon S \longrightarrow T$ and $g\colon S \longrightarrow T$ be two functions. Then

$$f = g$$

if and only if

$$\text{for all } s \in S \text{ we have } fs = gs.$$

In every-day language you are more likely to find the phrase 'every' instead of 'for all'. Mathematicians like to use phrases that are a little bit different from what is common elsewhere to draw attention to the fact that they mean their statement in a formal sense. 'Every first year computer science student takes COMP11120' is a claim that is of this form. In order

to check whether it is true you have to go through all the first year students in the School and check whether they are enrolled on this unit.

In order to show that a statement beginning 'for all' does not hold it is sufficient to find one element of the given set for which it fails to hold.

In the previous example, if you can find *one* student in computer science who is not enrolled on COMP11120 then you have shown that the statement given above does not hold.

There exists. This is a phrase that is frequently found both in mathematical definitions and arguments. For example in the natural numbers,

$$\begin{aligned} &m \text{ is divisible by n} \\ &\text{if and only if} \qquad \text{there exists } k \in \mathbb{N} \text{ such that } m = kn. \end{aligned}$$

Whenever a statement is made about existence there should be two parts to it:

- Where does the element exist? There should always be a set associated with the statement. In the above example, k had to be an element of $\mathbb{N}$ (and indeed the existence of some $r \in \mathbb{R}$ with the same property would completely change the definition and make it trivial).
- What are the properties that this element satisfies? There should always be a statement which specifies this. In the example above the property is $m = kn$ (for the given m and n).

One proves a statement of this form by producing an element which satisfies it, which is also known as a *witness*. For example, to show that 27 is divisible by 9 I have to show that there exists $k \in \mathbb{N}$ with $27 = k9$. To show this I offer $k = 3$ as a witness, and a simple check verifies that this element has the desired property.

To go back to the database example you might wonder whether you have a customer in your database who lives in Italy, or whether you have a customer who is paying with cheques (so that you can inform them that you will no longer accept these as a payment method).

Again in every-day language the phrase 'there is' is more common than 'there exists'. The latter serves to emphasize that a statement including is should be considered a precise mathematical statement. 'There is a student who is enrolled on COMP25212 and MATH20302' may have implications for the timetable.

Sometimes instead of merely demanding the existence of an element we might demand its *unique existence*. This is equivalent to a quite complex statement and is discussed below.

In order to show that a statement beginning 'there exists' does not hold one has to establish that it fails to hold for *every* element of the given set. So to demonstrate that the statement above regarding students does not hold you have to check every single second year student.

2.2.2 Complex statements

The 'if and only if' statement is an example of a slightly more complex statement, where two implications are combined using 'and'. There is one further complex statement that is used fairly frequently.

Unique existence

We sometimes demand that there exists a *unique* element with a particular property. This is in fact a convenient shortcut.

$$\text{There exists a unique } s \in S \text{ with the property } P$$

holds if and only if

$$\text{there exists } s \in S \text{ with property } P \qquad \text{and}$$
$$\text{for all } s, s' \in S, \text{ if } s \text{ and } s' \text{ satisfy property } P \text{ then } s = s'.$$

For example, if $f\colon S \longrightarrow T$ is a function from the set S to the set T then we know the following:

$$\text{for every } s \in S \text{ there exists a unique } t \in T \text{ with } fs = t.$$

Uniqueness is important here: We expect that a function, given an input value, produces precisely one output value for that input. So if we have valued t and t' in T which both satisfy the statement then we have $t = fs = t'$, and so $t = t'$. This idea is used in characterizing graphs of functions on page 36.

We have the key ingredients that formal statements are made of, namely the key phrases which allow us to analyse their structure. Analysing the structure of a statement allows us to construct a blueprint for a proof of that statement. The key ideas are given in the text above, but we give a summary table here By 'counterproof' we mean a proof that the statement does not hold. In the table S, $S1$ and $S2$ are statements, possibly containing further key phrases.

statement	proof	counterproof
$S1$ and $S2$	proof of $S1$ and proof of $S2$	counterproof for $S1$ or counterproof for $S2$
$S1$ or $S2$	proof of $S1$ or proof of $S2$	counterproof for $S1$ and counterproof for $S2$
if $S1$ then $S2$	assume $S1$ holds and prove $S2$	assume $S1$ holds and give counterproof for $S2$
for all x, S	assume an arbitrary x is given and prove S for that x	give a specific x and show S does not hold for that x
there is x such that S	find a specific x and show that S holds for that x	assume you have an arbitrary x and show S does not hold for that x

Every statement we might wish to prove, or disprove, is constructed from these phrases. In order to find a blueprint for a proof, or counterproof, all we have to do is to take the statement apart, and follow the instructions from this table. We give a number of examples for how to do this in the following sections.

We look at formal proofs in the following sections, but for the moment we concentrate on working out whether a given statement is true, and giving a reason for our conviction. We rely here on definitions given in Chapter 0.

Example 2.1. Assume we are given the statement:

$$\text{For all } n \in \mathbb{N},\ 2n \text{ is even,}$$

and we are asked whether this is true or not, and why. On the left we give a running commentary on what one might think, and on the right we write down what one might write down in such a situation.

Do we believe the statement? If you cannot tell at once it's usually a good idea to test it out for the first few numbers. $2 \cdot 0 = 0$ is even, $2 \cdot 1 = 2$ is even, $2 \cdot 2 = 4$ is even - this looks good. The table on the previous page says to show a statement of the form 'for all ...' we should assume we have a natural number n.	Let n be in $\mathbb{N}$.
So far so good. What about the statement $2n$ is even? At this point one should always look up the formal definition of the concepts used in the statements. Definitions of evenness, and divisibility appear in Chapter 0.	We have to show that $2n$ is even, that is that 2 divides $2n$.
So now we have put in the definition of evenness, but that leaves us with divisibility, so we put in that definition.	Hence we have to show that there is $k \in \mathbb{N}$ with $2n = 2k$. We pick $k = n$ and so the claim is established.

Example 2.2. Assume that we have the statement

$$\text{for all } n \in \mathbb{N} \setminus \{0, 1\},\ n \text{ is prime or } n \text{ is a multiple of 2.}$$

We proceed as in the previous example.

Do we believe the statement? Well, 0 and 1 have been excluded, so let's look at the next few numbers. We have that 2 is prime, 3 is prime, 4 is a multiple of 2, 5 is prime... This looks good, but do we really believe this? Are there really no odd numbers which are not prime? The number 9 comes to mind. So we want to give a counterproof. The table above tells us that we are looking for one n such that the statement does not hold.	Let $n = 9$.
To give a counter proof we have to show that 9 does not satisfy the claim. The two statements are connected with 'or', so according to the table above we have to show that *neither* holds.	We note that 9 is not prime since $9 = 3 \cdot 3$. But 9 is not even since $9 \operatorname{div} 2 = 1$. Hence this is a counterexample to the claim.

Example 2.3. Assume we are given the statement

$$\text{there is an } n \in \mathbb{Z} \setminus \{0, 1\} \text{ such that } n + n = -(n \cdot n).$$

We proceed as before.

Do we believe the statement? If we try $2+2$ we get 4, but $-(2 \cdot 2) = -4$, and clearly we get a sign mismatch for any positive integer. But what about $n = -2$?
The table above tells us that all we have to do is find one element for which the claim is true.

If we set $n = -2$ then we have

$$\begin{aligned} n + n &= -2 + (-2) \\ &= -4 = -(2 \cdot 2) \\ &= -(n \cdot n) \end{aligned}$$

as required.

Example 2.4. Assume we are given the statement

There exists $n \in \mathbb{Z}$ such that n is a multiple of 3 and n is a power of 2..

Do we believe the statement? A bit of thinking convinces us that powers of 2 are only divisible by powers of 2, so they cannot be divisible by 3 and therefore they cannot be a multiple of 3.
But how do we show that such a number cannot exist? This is a situation where what counts as a formal proof very much depends on what properties one may use.
The cleanest proof is via the prime factorization of integers (or corollaries thereof), but that is more than I want to cover in these notes.
In a situation where you cannot see how to write down a formal proof you should write something along the lines of the first paragraph written here. Never be afraid of expressing your thoughts in plain English!

If you were to start a formal proof it would look like something on the right.

Assume that n is a power of 2, that is, there exists $k \in \mathbb{N}$ such that $n = 2^k$.
If n is a multiple of 3 then there is $l \in \mathbb{Z}$ such that $n = 3l$.
Hence $2^k = n = 3l$.

This is where you would like to use that 3 cannot divide 2^k, but this requires a fact that is not given in Chapter 0. So the best you can do is to write what I wrote on the right.

This means that 3 must divide 2^k, which is impossible.

Exercise 21. Which of the following statements are valid? Try to give a reason as best you can, following the previous examples. You should use the definitions from Chapter 0 for the notions of evenness and divisibility (and there is a formal definition of primeness below, but for this exercise you may use the one you are familiar with).

(a) For all $n \in \mathbb{N}$, n is even or n is odd.

(b) There exists $n \in \mathbb{N}$ such that n is even and n is a prime number.

(c) There exists a unique $n \in \mathbb{Z}$ such that n is even and n is a prime number.

(d) For all $n \in \mathbb{Z}$, n is divisible by 4 implies n is divisible by 2.

(e) For all $n \in \mathbb{Z}$, n is odd implies $n \bmod 4 = 1$ or $n \bmod 4 = 3$.

(f) There exists $n \in \mathbb{N}$ such that n is even implies n is odd.

(g) For all $n \in \mathbb{Z} \setminus \{0\}$ there exists m in $\mathbb{Z}$ such that $n \operatorname{div} m = 2$.

Examples for treating more complex statements, and giving more formal proofs, are given in the following sections.

2.3 Important examples

One shouldn't think of the above as mere 'phrases'—they allow us to construct formal statements and come with a notion of how to establish proofs for these. This is what mathematics is all about. We look at an even more formal treatment of these ideas in the material on logic. In this section we look at examples for such statements which give definitions which are important in their own right. The aim is for you to become familiar with the logical constructions as well as learning about the given example.

We look at a number of examples of precise statements, and how to prove or disprove them.

2.3.1 Numbers

We begin by giving examples within some sets of numbers. Recall the formal definition of one integer dividing another, see Definition 1: An integer m divides an integer n if and only if there exists an integer k such that $k \cdot m = n$.

Example 2.5. We prove the following statement for integers i, j and k:

If i divides j then i divides $j \cdot k$.

As before on the left hand side we keep a running commentary, and on the right hand side we write down what the proof should look like.

This is an 'if ... then' statement. The table above tells us we should assume the first statement holds.	Assume that i and j are integers and that i divides j.
Sooner or later we have to apply the formal definition of divides to work out what this means.	This means that there exists an integer m such that $i \cdot m = j$.
It is usually a good idea to write down what we have to prove, again expanding the definition of 'divides'.	We have to show that i divides $j \cdot k$, that is, that there exists an integer n such that $i \cdot n = j \cdot k$.

We have to establish a 'there exists' statement, and the table tells us we have to find an element for which it is true. At this point one usually has to stare at the statements already written down to see whether there is an element with the right property hidden among them.

We have $j \cdot k = (i \cdot m) \cdot k$ by the assumption, and so

$$j \cdot k = i \cdot (m \cdot k)$$

by associativity of $\cdot$, and so we have found an integer, namely $m \cdot k$ with the property that we may multiply it with i to get $j \cdot k$.

Example 2.6. We carry out another example in the same style. Assume we are asked to prove or disprove the following statement for integers i, j and k.

If i divides k, and j divides k, then $i \cdot j$ divides k.

Now we first have to work out whether we want to prove the statement, or find a counterproof. Usually it's a good idea to do some examples. 2 divides 6 and 3 divides 6, and $2 \cdot 3 = 6$ divides 6, but 2 divides 2 and 2 divides 2, whereas $2 \cdot 2 = 4$ does not divide 2, so this statement is false.

But what does a formal argument look like in this case? The table from above tells us that it is sufficient to find one way of picking i, j, and k which makes the claim false. We show how to use the counter example we found informally to formally establish that the statement does not hold.

We note that $2 \cdot 1 = 2$, and so 2 divides 2. We pick $i = j = k$. For those choices, the above establishes that i divides k and that j divides k.
But $i \cdot j = 4$ and this number does not divide $k = 2$, hence the statement is false.

Exercise 22. Prove or disprove the following statements about divisibility in the setting of the preceding definition. Assume that i, j and k and m are integers. Follow Examples 2.5 and 2.6 (you don't have to give the running commentary).

(a) If i divides k and j divides m then $i \cdot j$ divides $k \cdot m$.

(b) If i^2 divides $j \cdot k$ then i divides j and i divides k.

(c) If i divides j and j divides k then i divides k.

Here is a definition of a number being prime that will look different from the one you have seen before. The aim of this is to encourage you to follow the given formal definition, and not your idea of what it should mean.

Definition 9. An element $n \neq 1$ of $\mathbb{N}$ (or $n \neq \pm 1$ in $\mathbb{Z}$) is **prime** if and only if for all elements k and l of $\mathbb{N}$ (or $\mathbb{Z}$) it is the case that

n divides kl implies n divides k or n divides l.

Exercise 23. Establish the following claims for prime numbers=, using the definition given above.

(a) Show that if an element $n \neq 1$ of $\mathbb{N}$ is prime then

$$m \text{ divides } n \qquad \text{implies} \qquad m = 1 \text{ or } m = n.$$

(b) Show that if m and n are prime in $\mathbb{N}$, $m \neq n$, and k is any natural number then

$$m \text{ divides } k \qquad \text{and} \qquad n \text{ divides } k \qquad \text{implies} \qquad mn \text{ divides } k.$$

(c) Show that if $n \neq \pm 1$ is prime in $\mathbb{Z}$ then

$$m \text{ divides } n \qquad \text{implies} \qquad m = \pm 1 \text{ or } m = \pm n.$$

Note that the converse of (b) and (c) are also true, that is, our definition of primeness is equivalent to the one you are used to. However, the proof requires a lot more knowledge about integers than I want to ask about here.

2.3.2 Operations

Functions that appear very frequently are *operations* on a set. Usually we are interested in *binary operations on a set* S, that is functions

$$S \times S \longrightarrow S.$$

Examples of such functions are

- addition and multiplication for $\mathbb{N}$,
- addition, and multiplication for $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ or $\mathbb{C}$, as well as the derived operation of subtraction,
- union and intersection of sets as functions from $\mathcal{P}X \times \mathcal{P}X$ to $\mathcal{P}X$,
- concatenation of strings in Java,
- concatenation of lists (see Section 5.2) over some set.

Note that division almost fits into this scheme, but since we may not divide by 0 we can only define it as a function where the source has been adjusted, for example,

$$\mathbb{R} \times (\mathbb{R} - \{0\}) \longrightarrow \mathbb{R}.$$

These are operations we use all the time, and they are deserving of further study. Note that we typically write binary operations in *infix notation*, that is, we write the operation between its two arguments, such as $r + r'$, $c \cdot c'$.

For what follows we need an *arbitrary* binary operation, and for that we will use the symbol $\circledast$.

Definition 10. A binary operation $\circledast$ on a set S is **associative**[5] if and only, for all s, s', s'' in S it is the case that

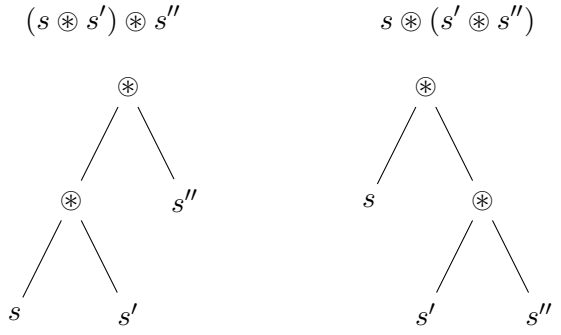
$$(s \circledast s') \circledast s'' = s \circledast (s' \circledast s'').$$

Why is this important? We use brackets to identify in which order the operations should be carried out. We can think of the two expressions as encoding a tree-like structure (known as a *parse tree*[6]), which tells us in

[5] Do not confuse this with 'operator associativity' from *Java: Just in Time*.

[6] Parse trees are studied in detail in COMP11212.

which order to carry out the operations present in the expression.



We look at an example. Recall that $m - n$ is a shortcut for calculating $m + (-n)$, Using that derived operation as an example, we illustrate how one can think of this as allowing the filling in of the various steps of the calculation:

whereas

Note that this example establishes that subtraction is *not* associative for the integers since it shows that

$$(3-4)-5 \neq 3-(4-5).$$

Knowing that an operation is associative means that both trees evaluate *to the same number* and therefore we *may leave out brackets* when using such an operation. It is safe to write

$$s \circledast s' \circledast s''$$

for such an operation.

This is important to computer scientists for two main reasons:

- When writing a program, leaving out brackets in this situation makes the code more readable to humans.
- When writing a compiler for a programming language, knowing that an operation is associative may allow significantly faster ways of compiling.

Note that if we write our operation as a binary function

$$f\colon S \times S \longrightarrow S$$

where we use prefix notation then associativity means that the following equality holds:

$$f(f(s,s'),s'') = f(s,f(s',s''))$$

Example 2.7. Assume we are given a set X. Recall from Section 0.2.4 that we may think of the union operation as a function

$$\cup\colon \mathcal{P}X \times \mathcal{P}X \longrightarrow \mathcal{P}X \ .$$

We show below that this operation is associative.

The statement we wish to show is a 'for all ...' statement. Following the table on page 60 we assume that S, S' and S'' are (arbitrary) elements $\mathcal{P}X$. We calculate

$$\begin{aligned}
(S \cup S') \cup S'' &= \{x \in X \mid x \in S \text{ or } x \in S'\} \cup S'' && \text{def union}\\
&= \{x \in X \mid (x \in S \text{ or } x \in S') \text{ or } x \in S''\} && \text{def union}\\
&= \{x \in X \mid x \in S \text{ or } x \in S' \text{ or } x \in S''\} && \text{common sense}\\
&= \{x \in X \mid x \in S \text{ or } (x \in S' \text{ or } x \in S'')\} && \text{common sense}\\
&= S \cup \{x \in X \mid x \in S' \text{ or } x \in S''\} && \text{def union}\\
&= S \cup (S' \cup S'') && \text{def union}
\end{aligned}$$

Note that we have justified each step in the equalities used above—this ensures that we check we only use valid properties, and tells the reader why the steps are valid.

Note that we had to invoke 'common sense' here—usually this means that we are relying on definitions that are not completely rigorous mathematically speaking. What we have done in the definition of the union of two sets is to rely on the meaning of the English language. Only when we are down to that is it allowable to use 'common sense' as a justification (you might also call it 'the semantics of the English language'). In formal set theory there is formal logic to define the union of two sets, but we do not go to this level of detail here.

Example 2.8. What does proof look like that an operation is not associative? One is already given above, for subtraction as a derived operation for integers.

We are trying to find a counterproof for a statement of the form 'for all ...'. According to the table on page 60 all we have to do is find instances of the various variables such that the statement doesn't hold.

Above we have found a counter example, and below we illustrate how one might write this up.

We calculate $(3-4)-5 = -1-5 = -6$, whereas $3-(4-5) = 3-(-1) = 3+1 = 4$, and so subtraction is not an associative operation for integers.

Since we so far do not have formal definitions of addition and multiplication for $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ it is impossible to formally prove that these are indeed associative. You may use this as a fact in your work for the following exercise.[7]

Exercise 24. Work out whether the following operations are associative.

(a) Intersection for sets.

(b) Addition for complex numbers.

(c) Subtraction for complex numbers.

(d) Multiplication for complex numbers.

(e) Define the average ave of two real numbers r, r' as

$$\text{ave}(r, r') = \frac{r + r'}{2}.$$

Is this operation associative? Would you apply it to calculate the average of three numbers? If not, can you think of a better averaging function?

(f) Multiplication of real numbers where every number is given up to *one* post-decimal digit, and where rounding takes place every time after a multiplication has been carried out.[8]

(g) The `+` operator for strings in Java.

(h) The `&&` operator for boolean expressions in Java.

(i) Let S be a set and let $\mathsf{Fun}(S, S)$ be the set of all functions with source S and target S. Show that composition is an associative operation on the set $\mathsf{Fun}(S, S)$.

Some operations allow us even greater freedom: Not only is it unnecessary to provide brackets, we may also change the *order* in which the arguments are supplied.

Definition 11. A binary operation $\circledast$ on a set S is **commutative** if and only if, for all s and s' in S we have

$$s \circledast s' = s' \circledast s.$$

If an operation is commutative then it does not matter in which order arguments are supplied to it. Hence the two trees below will evaluate to give the same result.

[7] Note that formal definitions, and proofs, of these properties for the natural numbers are given in Section 5.1.

[8] When programming there is usually limited precision, and rounding has to take place after each step of the computation. While a computer has more precision, say for floating point numbers, the problems that occur are the same as here.

Example 2.9. Again we look at the union operation on the powerset $\mathcal{P}X$ for a given set X.

Once more this is a statement of the 'for all ...' kind. Following the table on page 60 once again we assume that we have (arbitrary) elements S and S' of $\mathcal{P}X$. The union of S and S' is defined as follows:

$$S \cup S' = \{x \in X \mid x \in S \text{ or } x \in S'\}$$

and, once again invoking 'common sense', this is the same as

$$\{x \in X \mid x \in S' \text{ or } x \in S\} = S' \cup S.$$

Example 2.10. Consider the following operation for complex numbers:[9] Given z and z' in $\mathbb{C}$ we set

$$z \circledast z' = \overline{z}z'.$$

The question is whether this operation is commutative.

First of all we have to work out whether we think it is true, and should try to prove it, or whether we should aim for a counterproof.

There are two approaches here: You can write down what this operation does in terms of real and imaginary parts, or you can think for a moment about what the conjugate operation $\bar{\cdot}$ does. It affects the imaginary part only, so if we have the product of one number with imaginary part 0, and one with imaginary part other than 0, there should be a difference. This suggests we should try a counterproof, that is, we should find one choice for z, and one for z', such that the statement becomes false.

The simplest numbers fitting the description given above, and which are distinct from 0, are 1 and i. We check

$$i \circledast 1 = \overline{i} \cdot 1 = -i \cdot 1 = -i,$$

and

$$1 \circledast i = \overline{1} \cdot i = 1 \cdot i = i.$$

Since $-i \neq i$ we have established that the given operation is not commutative.

Exercise 25. Work out whether the following operations are commutative. If you think the answer is 'yes', give a proof, if 'no' a counter example.

(a) Multiplication for complex numbers.

(b) Subtraction for integers.

(c) Division for real numbers different from 0.

(d) Set difference on some powerset.

(e) The ave function from the previous exercise.

(f) The `+` operator for strings in Java.

(g) The `&&` operator for boolean expressions in Java.

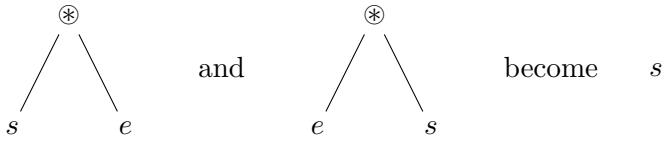
Some operations have an element which does not have any effect when combined with any other.

[9] This appeared in a past exam paper.

Definition 12. Let $\circledast$ be a binary operation on a set S. An element e of S is a[10] **unit for** $\circledast$ if and only if it is the case that for all elements s of S we have

$$s \circledast e = s = e \circledast s.$$

If we want to picture this using a tree then it is saying that



This looks odd, but if you think of the first two trees as being part of a larger tree then this becomes a useful simplification rule.

Example 2.11. We have already seen a number of examples of units. The number 0 is the unit for addition on all the sets of numbers we cover in these notes. This is one of the statements from Fact 1 (and corresponding facts about the other sets of numbers), since for all $n \in \mathbb{N}$ we have

$$n + 0 = n = 0 + n.$$

Example 2.12. Consider the subtraction operation for integers. Does this operation have a unit? Once again, we first have to decide whether we should try to give a proof or a counterproof.

The statement in question is of the kind 'there exists ...'. To prove such a statement we have to give an element with the required property. In a situation where we're not sure what such an element might look like, it is often possible to derive properties it needs to have. This is the strategy we follow here.

If the number we have e were a unit for subtraction we would require

$$n - e = n$$

for all elements n of $\mathbb{Z}$. The only number which satisfies this is $e = 0$, but if we calculate

$$0 - 1 = -1,$$

we see that this element cannot be the unit since we would require that number to be equal to 1 to satisfy $e - n = n$ for all $n \in \mathbb{N}$. Hence the given operation does not have a unit.

Note that the subtraction operation satisfies none of our properties! For this reason it is quite easy to make mistakes when using this operation, and that is why it is preferable to not to consider subtraction a well-behaved operation.

It is usually harder to establish that an operation does not have a unit, so we give another example for this case.

[10]This is sometimes also known as the identity for the unit, but that terminology might create confusion with the identity function for a set.

Example 2.13. Let's consider the following set difference operation on $\mathcal{P}X$ for a given set X. Recall that for S, S' in $\mathcal{P}X$ this is defined as

$$S \setminus S' = \{s \in S \mid s \notin S'\}.$$

Does this operation have a unit? As in the previous example we derive properties that such a unit would have to have.

In order for $S \setminus S' = S$ to hold it must be the case that none of the elements of S occurs in S'. In particular if U were the unit we must have

$$X \setminus U = \{x \in X \mid x \notin U\} = X,$$

which means that U must necessarily be empty. But for the empty set we have

$$\emptyset \setminus X = \{x \in \emptyset \mid x \notin X\} = \emptyset,$$

but for $\emptyset$ to be the unit this would have to be equal to X.

This means that no element of $\mathcal{P}X$ can satisfy the requirements for a unit for this operation.

Exercise 26. Identify the unit for the following operations, or argue that there cannot be one:

(a) Union and intersection of subsets of a given set X.

(b) Multiplication for integers, rational, real and complex numbers.

(c) The operation from Example 2.10.

(d) The ave operation for the preceding two exercises.

(e) The `+` operator for strings in Java.

(f) The `&&` operator for boolean expressions in Java.

Note that mathematicians call a set with an associative binary operation which has a unit a **monoid**.

Exercise 27. Prove that there is at most one unit for a binary operation $\circledast$ on a set S. *Hint: Assume you have two elements that satisfy the property defining the unit.*

Exercise 28. Consider the set $\mathsf{Fun}(S, S)$ of all functions from some set S to itself. This has a binary operation in the form of function composition. If you have not already done so in Exercise 24 then show that this operation is associative. Find the unit for the operation. Conclude that we have a monoid. Further show that the operation is not commutative in general.

Some operations come in pairs, for example, addition and subtraction (for any of our system of numbers) are related. However, mathematically speaking, it is preferable to think about inverses rather than such an accompanying operation. Above we have pointed out that $m - n$ is merely a shortcut for calculating $m + (-n)$, that is, we should add the inverse of n for the addition operation to m. So instead of trying to find some rules that the subtraction operation satisfies it turns out that it's much easier to define inverses, and their properties.

Definition 13. Let $\circledast$ be an associative binary operation with unit e on a set S. We say that the element s' is an **inverse for $s \in S$ with respect to** $\circledast$ if and only if we have

$$s \circledast s' = e = s' \circledast s.$$

It is standard to write s^{-1} for the inverse of s, but that convention changes if one uses the symbol $+$ for the operation. In that case one writes $-s$ for the inverse of the element s with respect to the operation $+$.

For addition on the integers, for example, the inverse of an element n is $-n$. If we restrict addition to the natural numbers then there is a unit, but inverses do not exist.[11]

For multiplication we only get inverses when we have at least the rational numbers to work from (but note that 0 does not have a multiplicative inverse). For the rational or real numbers the inverse of an element $r \neq 0$ is r^{-1}. In Chapter 1 we have proved that inverses exist for both, addition and multiplication,[12] and we have shown how to calculate them for a given element.

Example 2.14. The proof that inverses for addition exist for integers, rationals, or reals, is very short: Given such a number r, we are so used to the fact that $r + (-r) = 0 = -r + r$ that it hardly feels as if this is a proof!

Exercise 44 gives an example of an operation where inverses exist, where the underlying set is not a set of numbers.

Example 2.15. To show that a given operation does not have inverses for every element one has to give an element which does not have one.

Assume that X is a set onsider the intersection operation,

$$ass \cap \mathcal{P}X \times \mathcal{P}X\mathcal{P}X(S, S')S \cap S'.$$

The unit for this operation is given by X. We show that the empty set does not have an inverse:

If S were an inverse for $\emptyset$ with respect to $\cap$ it would have to be the case that $S \cap \emptyset = X$. But $S \cap \emptyset = \emptyset$, and so as long as X is non-empty, an inverse cannot exist.

Note that if s^{-1} is the inverse for s with respect to $\circledast$ then s is the inverse of s^{-1} with respect to that operation since the definition is symmetric.

There are many more binary operations (typically not carried out on numbers) where inverses do not exist.

Exercise 29. For the following operations, give an argument why inverses do not exist.

(a) Union of subsets of a given set.

(b) The ave function from the previous exercises.

(c) The `+` operation for strings in Java.

[11] Think about why the only natural number with an additive inverse is 0.

[12] But remember that the complex number 0 does not have a multiplicative inverse.

(d) The && operation for boolean expressions in Java.

Exercise 30. Let S be a set with an associative binary operation $\circledast$, and assume that $e \in S$ is the unit for that operation.

(a) Show that the inverse for the element $s_1 \circledast s_2$ is given by $s_2^{-1} \circledast s_1^{-1}$.

(b) Show that every element has at most one inverse. *Hint: Assume that there are two inverses and prove that they have to be the same.*

One should think of subtraction as *adding the inverse* of the given element, that is, for example for integers m and n,

$$m - n = m + (-n).$$

In that sense subtraction isn't a new operation at all: It can be defined completely in terms of addition, and the properties of that operation. Similarly, division is multiplying by the appropriate multiplicative inverse.

Note that mathematicians call a set with an associative binary operation with a unit, and where element has an inverse, a **group**. Groups are very nice mathematical entities, but most of the sets with a binary operation you will see will not have the full structure of a group (typically lacking inverses).

Optional Exercise 6. Assume that A is a set with a binary operation $\circledast$ which is associative and has a unit. Consider the set $\mathsf{Fun}(X, A)$ of all functions from some set X to A. Show that the following definition gives an associative operation on $\mathsf{Fun}(X, A)$. For $x \in X$ set

$$(f \circledast g)x = fx \circledast gx.$$

This is known as defining an operation *pointwise* on as set of functions. Find the unit for this operation and show that it is one. If the operation on A is commutative, what about the one on $\mathsf{Fun}(X, A)$?

2.3.3 Functions

Functions allow us to transport elements from one set to another. Section 0.3 gives a reminder of what you should know about functions before reading on.

The **graph of a function**[13] $f\colon S \longrightarrow T$ is defined as

$$\{(s, fs) \in S \times T \mid s \in S\}.$$

This is the set we typically draw when trying to picture what a function looks like, at least for functions from sets of numbers to sets of numbers. The typical case for that is for S and T to be subsets of $\mathbb{R}$.

We can characterize all those subsets of $S \times T$ which are the graph of a function of the type $S \to T$.

Proposition 2.16. *A subset G of $S \times T$ is the graph of a function from S to T if and only if*

for all $s \in S$ there exists a unique $t \in T$ with $(s, t) \in G$.

[13] See also Section 0.3.3 for an introduction to the idea and Section 0.3.4 for some examples.

This statement requires a proof. We give one here as another example for how to use the key phrases in the statement to structure the proof.

We have an 'if and only if' statement, and we split the proof into two parts accordingly.

- Assume that[14] G is the graph of a function. We would like to have a name for that function, so we call it f. We have to show it has the given property. This is a statement of the form 'for all ...', so following the table on page 60 we assume that we have an arbitrary $s \in S$. We now have to establish the remainder of the given statement. This is a 'unique existence' property, which means we have to show two things:
  - **Existence.** We know that (s, fs) is in the graph G of f, and so we have found a witness in the form of $t = fs$ for the existence part.
  - **Uniqueness.** Assume we have t and t' in T with (s, t) and (s, t') both elements of G. But by definition of G we know that the only element with first component s is (s, fs), and so we must have $t = fs = t'$ and we have established the uniqueness part.
- Assume that[15] G is a subset of $S \times T$ satisfying the given condition. We have to show that G is the graph of a function, and the only way of doing this is to
  - define a function f and
  - show that G is the graph of f.

  We do those steps in turns.
  - Define a function $f\colon S \longrightarrow T$ by setting

    $$f\colon s \longmapsto t \qquad \text{if and only if} \qquad (s, t) \in G.$$

    We have to check that this definition produces precisely *one* output in T for every input. By existence we know there is at least one element of T for every $s \in S$. But by uniqueness we know that if (s, t) and (s, t') are in G then $t = t'$, so there is at most one element for every $s \in S$.
  - The graph of the function f is given as follows.

    $$\begin{aligned}\{(s, fs) \in S \times T \mid s \in S\} &= \{(s, t) \in S \times T \mid (s, t) \in G\} \quad \text{def } f \\ &= G\end{aligned}$$

This completes the proof.

Some functions have particular properties that are important to us.

Definition 14. A function $f\colon S \longrightarrow T$ is **injective** if and only if

$$\text{for all } s \text{ and } s' \text{ in } S \qquad fs = fs' \qquad \text{implies} \qquad s = s'.$$

Under these circumstances we say that f is an **injection**.

[14] This direction is sometimes known as the 'forward' or 'if' direction.

[15] This direction is sometimes known as the 'backwards' or 'only if' direction.

One way to paraphrase[16] this property is to say that two different elements of S are mapped to two different elements of T. This means that knowing the result $fs \in T$ of applying f to some element s of S is sufficient to recover s.

For example the function d from $\mathbb{N}$ to $\mathbb{N}$ given by

$$d\colon n \longmapsto 2n$$

is injective since given n, n' in $\mathbb{N}$ we know that

$$2n = dn = dn' = 2n' \qquad \text{implies} \qquad n = n'.$$

On the other hand the function from $\mathbb{R}$ to $\mathbb{R}$ given by

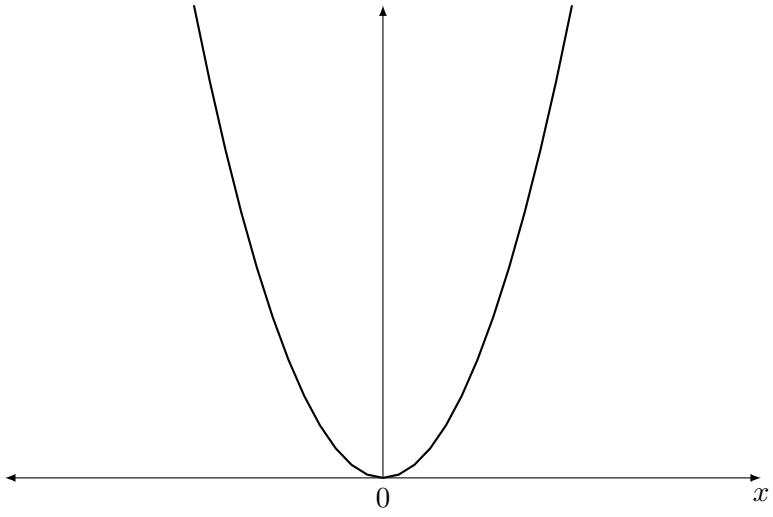
$$r \longmapsto r^2$$

is not injective. In order to establish this we merely have to produce two elements of $\mathbb{R}$ which are mapped to the same image. Since $1^2 = 1 = (-1)^2$ we know that 1 and -1 have the same image under this function and so it is not injective.

This property is important, for example, when casting an element of some datatype to another. We expect that if we cast an `int` to a `double` in Java that two different `int` values will be cast to different `double` values. This operation should be performable without losing any information.

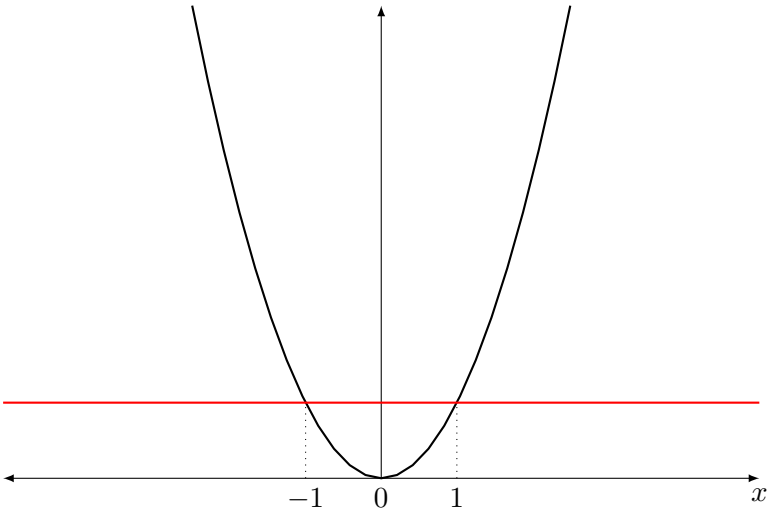
We can also think of an injection as a 'unique relabelling' function: Every element from the source set S is given a new label from the target set T in such a way that no two elements of S are given the same label.

The graph of a function can be useful when determining whether a function is injective. For the squaring function described above the graph looks like this.



Whenever we can draw a horizontal line that intersects the graph of our function in more than one place then the function is not injective:

[16] But usually not a good way of attempting a proof.



The x-coordinates of the two intersection points give us two different elements of $\mathbb{R}$ where the function takes the same value.

Note that one has to be careful when using the graph to determine whether a function is injective: Since most examples have an infinite graph it is impossible to draw all of it, so one has to ensure that there isn't any unwanted behaviour in the parts not drawn.

Example 2.17. We show that the function whose graph is given above is not injective. Consider the function

$$\begin{aligned} f\colon \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\longmapsto x^2 \end{aligned}.$$

Injectivity is a 'for all ...' statement. To give a counterproof it is therefore sufficient to find witnesses s and s' such that the given statement does not hold. The graph tells us how to find those witnesses.

Let $s = -1$, and let $s = 1$. Then $fs = (-1)^2 = 1$, and $fs' = 1^2 = 1$, and since $s = -1 \neq 1 = s'$ we have found a counterexample.

Showing that a function is injective requires more of an argument.

Example 2.18. Consider the function

$$\begin{aligned} f\colon \mathbb{N} &\longrightarrow \mathbb{N} \\ n &\longmapsto 2n \end{aligned}.$$

We show that this function is injective.

Since injectivity is a 'for all ...' statement the table on page 60 suggests that we should assume that we have n and n' (arbitrary) in $\mathbb{N}$.

We now have to show that the remainder of the statement holds. This is of the form 'if ... then', and following the same table we assume that $2n = fn = fn' = 2n'$. Dividing by 2 on both sides we may conclude that $n = n'$ as required.

If there is an injection from some set S to some set T then we may deduce that T is at least as large as S. See the Section 4.0.1 for more detail, in particular Exercise 57.

Exercise 31. Show that the following functions are injective or not injective as indicated.

(a) Injective: The function from $\mathbb{R}$ to $\mathbb{C}$ defined on page 43.

(b) Not injective: The function $x \longmapsto 2x^2 - 4x + 1$ from $\mathbb{R}$ to $\mathbb{R}$.

(c) Not injective: The function from the set of first year CS student to lab groups $M + W$, $B + X$, Y and Z.

(d) Injective: The function[17] $x \longmapsto x^3 + x + 1$ from $\mathbb{R}$ to $\mathbb{R}$.

Exercise 32. Determine which of the following functions are injective. You have to provide an argument with your answer. You should not use advanced concepts such as limits or derivatives, just basic facts about numbers. Where the function is not injective can you restrict the source set to make it injective?

(a) The sin function from $\mathbb{R}$ to $\mathbb{R}$.

(b) The log function from $[1, \infty)$ to $\mathbb{R}^+$.

(c) The function $x \longmapsto 2^x$ from $\mathbb{N}$ to $\mathbb{N}$, or from $\mathbb{R}$ to $\mathbb{R}$, you may choose.

(d) The function used by the School from the set of first year CS students to the set of tutorial groups.

(e) The function used by the University from the set of first year CS students to the set of user ids.

(f) The function from the set of first year lab groups $M + W$, X, Y and Z for COMP10120 to their lab slots in the timetable.

(g) The function $x \longmapsto x(-i)$ from $\mathbb{C}$ to $\mathbb{C}$.

(h) The function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$ which maps (n, m) to $2^n 3^m$.

(i) The function $x \longmapsto \{x\}$ from a set S to the powerset $\mathcal{P}S$.

Exercise 33. Establish the following properties.

(a) If S is a one-element set then every function which has it as a source is injective.

(b) The composite of two injective functions is injective.

(c) If $f \colon S \longrightarrow T$ and $g \colon T \longrightarrow U$ are two functions such that $g \circ f$ is injective then f is injective.

(d) Show for the previous statement that g need not be injective by giving an example.[18]

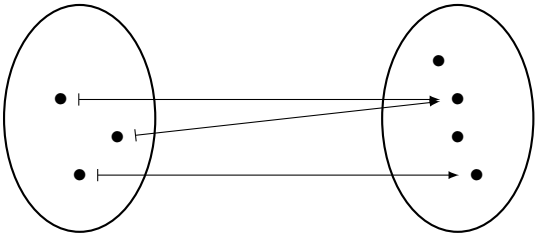
[17] Giving a formal proof is hard for this function and goes beyond this course unit—try to make an informal argument.

[18] The smallest example concerns sets with at most two elements. You may want to read the next two paragraphs to help with finding one.

(e) Assume that $f\colon S \longrightarrow S'$ and $g\colon T \longrightarrow T'$ are both injections. Show that this is also true for

$$\begin{aligned} f \times g\colon S \times T &\longrightarrow S' \times T' \\ (s,t) &\longmapsto (fs, gt). \end{aligned}$$

In the case where we have a function from one small finite set to another we can draw a different kind of picture.



If a function is given by a picture like this, then all one has to do to check injectivity is to see whether any element in the target set has more than one arrow going into it. The function given in the above picture is therefore not injective. Exercise 38 invites you to try this technique for yourself.

Exercise 34. Show that if S is a set with finitely many elements, and $f\colon S \longrightarrow T$ is an injective function from S to a set T then the image of S under f has the same number of elements as S.

Definition 15. A function $f\colon S \longrightarrow T$ is **surjective** if and only if

$$\text{for all } t \in T \qquad \text{there exists } s \in S \qquad \text{with} \qquad t = fs.$$

We also say in this case that f is a **surjection**.

In other words a function is surjective if its range is the whole target set, or, to put it differently, if its image reaches all of the target set.

We care that a function is surjective if we are using the source set to talk about members of the target set. For example the set of even numbers is given by

$$\{2n \in \mathbb{N} \mid n \in \mathbb{N}\},$$

which tells us that it is the image of the function f from $\mathbb{N}$ to $\mathbb{N}$ given by

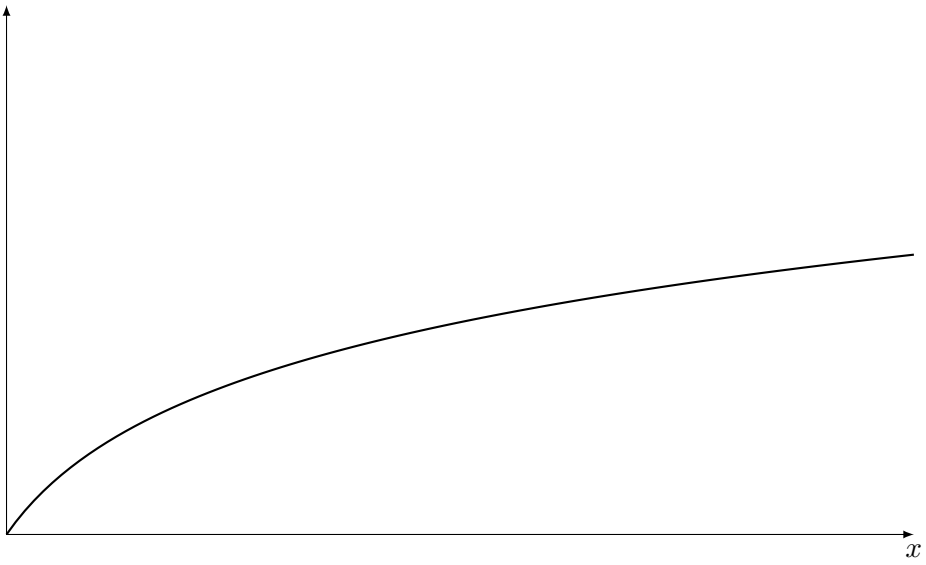
$$n \longmapsto 2n.$$

This means f is surjective, and we may use fn, where $n \in \mathbb{N}$ to access *all* even numbers.

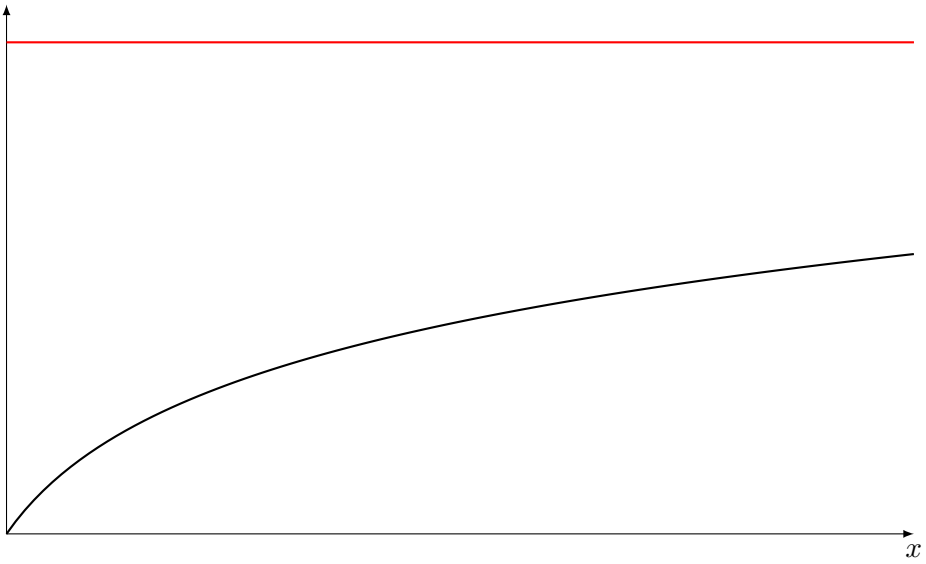
One can again take the graph of a function to help decide whether it is surjective. It can be tricky, however, to determine the answer from looking at the graph. Instead of looking whether there is a horizontal line which intersects the graph in at least two points we now have to worry about whether there is a horizontal line that intersects the graph not at all. For some functions this can by quite difficult to see.

Consider for example the function from $\mathbb{R}^+$ to $\mathbb{R}^+$ given by

$$x \longmapsto \log(x+1).$$



It is really difficult to judge whether some horizontal line will have an intersection with this graph or not.



You might argue that the problem would be solved if we drew a larger part of the graph, but then we could also move the horizontal line higher up (remember that one has to show that one can find an intersection for *every* horizontal line).

Example 2.19. The function

$$\begin{aligned} f\colon \mathbb{Z} &\longrightarrow \mathbb{Z} \\ x &\longmapsto x+1 \end{aligned}$$

is surjective. Surjectivity is a statement of the 'for all ...' kind, so following the table on page 60 we assume we are given $n \in \mathbb{Z}$.

The remainder of the surjectivity property is a 'there exists ...' statement, so according to the same table we have to find a witness, say $m \in \mathbb{Z}$. This witness has to satisfy $fm = n$. Inserting the definition of f, this means we need to pick m such that

$$m + 1 = fm = n,$$

so we pick $m = n - 1$ and this has the required property, so f is surjective.

Example 2.20. The function

$$\begin{aligned} f\colon \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\longmapsto x^2 \end{aligned}$$

is not surjective.

To show this we want to find a counterproof to a statement of the 'for all ...' kind. According to the table on page 60 means we have to find a witness $r \in \mathbb{R}$ that does not satisfy the remainder of the property.

Which property is this? It's a property of the 'there exists ...' kind, so following the same table we have to show that no r' in $\mathbb{R}$ satisfies that $(r')^2 = r$.

Putting it like this should give us the right idea: We choose $r = -1$, and then no real number r' can be squared to give r.

Alternatively, looking at the graph of this function (which is given above), we can see that any negative number would work as the required witness.

If there is a surjection from a set S to a set T then we may deduce that T is at most as large as S. See below for more detail on this idea.

Exercise 35. Show that the following functions are surjective or not surjective as indicated.

(a) Surjective: The function $n \longmapsto |n|$ from $\mathbb{Z}$ to $\mathbb{N}$.

(b) Surjective: The function used by the School from the set of first year CS students to the set of tutorial groups.

(c) Not surjective: The function used by the University from the set of all students currently in the university to the set of valid student id numbers.

(d) Not surjective: The function from $\mathbb{R}$ to $\mathbb{C}$ given on page 43.

Exercise 36. For the following functions determine whether they are surjective and support your claim by an argument. You should not use advanced concepts such as limits ore derivatives, just basic facts about numbers.

(a) The function from $\mathbb{Q}$ to $\mathbb{Q}$ given by

$$x \longmapsto \begin{cases} 0 & x = 0 \\ 1/x & \text{else.} \end{cases}$$

(b) The function from $\mathbb{R}$ to $\mathbb{R}$ given by $x \longmapsto x^2 + 2x - 1$.

(c) The function from $\mathbb{C}$ to $\mathbb{C}$ given by $x \longmapsto xi$.

(d) The function from $\mathbb{C}$ to $\mathbb{R}$ given by $x \longmapsto |x|$.

(e) The function that maps each first year CS student to their labgroup $W + M$, $B + X$, Y or Z.

(f) The function that maps each member of your tutorial group to one of the values E and W, depending on whether they were born in Europe (E) or in the rest of the world (W).

(g) The function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$ given by $(x, y) \longmapsto x$.

(h) The function from the *finite powerset* of $\mathbb{N}$,

$$\{S \subseteq \mathbb{N} \mid S \text{ has finitely many elements}\},$$

to $\mathbb{N}$ that maps S to the number $|S|$ of elements of S.

Exercise 37. Establish the following statements

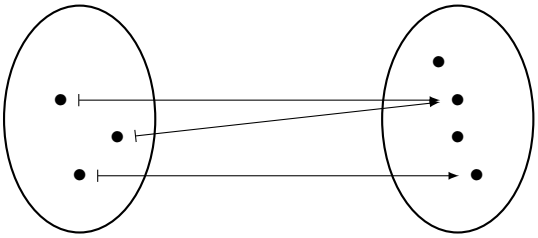
(a) The composite of two surjections is an surjection.

(b) If $f\colon S \longrightarrow T$ and $g\colon T \longrightarrow U$ are functions such that $g \circ f$ is surjective then g is surjective.

(c) Establish that in the previous statement f need not be surjective by giving an example.

(d) Assume that $f\colon S \longrightarrow S'$ and $g\colon T \longrightarrow U$ are both surjections. Show that this is also true for $f \times g$.

Again, if we are looking at functions between small finite sets then we can draw a picture like this.



For a function to be surjective all one has to check is that every element of the target set (on the right) has at least one arrow going into it. This example is not surjective.

Exercise 38. For the following functions draw a picture analogous to the above and determine whether or not it is injective and/or surjective.

(a) The function from $\{0, 1, 2, 3, 4\}$ to itself which maps the element i to i mod 3.

(b) The function from $\{0, 1, 2, 3, 4, 5, 6, 7\}$ to $\{0, 1, 2, 3\}$ which maps x to x mod 4.

(c) The function from $\{0, 1, 2, 3, 4\}$ to $\{n \in \mathbb{N} \mid n \leq 9\}$ which maps i to $2i$.

(d) The function from the set of members of your tutorial group to the set of letters from A to Z, which maps a member of the group to the first letter of their first name.

(e) The function that maps the members of your tutorial group to the set $\{\mathsf{M}, \mathsf{F}\}$ depending on their gender.

(f) The function from the set of labgroups

$$\{M+W, B+X, Y, Z\} \qquad \text{to the set} \qquad \{G23, LF31, \mathsf{both}\}$$

mapping a group to $G23$ if all their labs for COMP16121 this semester are in $G23$, to $LF31$ if all these labs take place in that room, and to both if they have labs in both these rooms.

We need two further notions for functions. First of all there is a name for functions which are both, injective and surjective.

Definition 16. A function $f\colon S \longrightarrow T$ is a **bijection** if and only if it is both, injective and surjective. We say in this case that it is **bijective**.

Example 2.21. Consider the function f from $\mathbb{Z}$ to $\mathbb{Z}$ given by

$$x \longmapsto x+1.$$

It is shown above that this function is surjective and so it remains to show that it is also injective. Assume we have n and m in $\mathbb{Z}$ with the property that

$$fn = fm.$$

If we insert the definition of f then this means that

$$n+1 = fn = fm = m+1,$$

and by deducting 1 on both sides we deduce

$$n = m.$$

This establishes that f is injective, and so it is bijective.

Recall that we may think of an injection as a function that attaches to every element from the source set S a unique label from the target set T. Since every bijection is injective we may think of it as such a relabelling function, but it is a special such since

- every element gets a unique label (true for all injections) and
- every label is used.

Exercise 39. Determine which of the following functions are bijections. Justify your answer.

(a) The function from $\mathbb{Q}$ to $\mathbb{Q}$ given by

$$x \longmapsto \begin{cases} 0 & x = 0 \\ 1/x & \text{else} \end{cases}$$

(b) The function from $\mathbb{C}$ to $\mathbb{C}$ given by $x \longmapsto xi$.

(c) The function from $\mathbb{Z}$ to $\mathbb{N}$ given by $x \longmapsto |x|$.

(d) The function from $\mathbb{C}$ to $\mathbb{R}$ given by $x \longmapsto |x|$.

Exercise 40. Show that if $f\colon S \longrightarrow T$ and $g\colon T \longrightarrow U$ are two functions and $g \circ f$ is a bijection then f is an injection and g is a surjection.

Whenever we have a bijection f there is a companion which undoes the effect of applying f.

Definition 17. A function $g\colon T \longrightarrow S$ is the **inverse of the function** of $f\colon S \longrightarrow T$ if and only if

$$g \circ f = \mathsf{id}_S \qquad \text{and} \qquad f \circ g = \mathsf{id}_T.$$

Note that if g is the inverse function of f then f is the inverse function of g since the definition is symmetric.

We illustrate how the properties of a function from S to T say something about a function one may construct going from T to S: Note that the proposition tells us that for an injective function we can find a function which satisfies one of the two equalities required for inverse functions.

Proposition 2.22. *If $f\colon S \longrightarrow T$ is an injection then we can find a function $g\colon S \longrightarrow T$ such that $g \circ f = \mathsf{id}_S$.*

Proof. To start pick an arbitrary element $s_\bullet$ of S. Now given $t \in T$ we would like to define g as follows:

$$g\colon t \longmapsto \begin{cases} s & \text{if there is } s \in S \text{ with } fs = t \\ s_\bullet & \text{else} \end{cases}$$

First of all we have to worry whether this does indeed define a function—we need to ensure that in the first case, only one such s can exist. But since f is an injection we know that $fs = fs'$ implies $s = s'$ and so s is indeed unique. Hence our definition does indeed give us a function g.

Secondly we have to check that the equations for f and g holds as promised. Given $s \in S$ we calculate

$$\begin{aligned} (g \circ f)s &= g(fs) && \text{def composition} \\ &= s && \text{def } g \\ &= \mathsf{id}_S s && \text{def identity function} \end{aligned}$$

□

Recall that we may think of a bijection as a function that attaches to every element of the source set a unique label from the target set T in such a way that all labels are used. What the exercise tells us is that we can undo the effect of the labelling: The inverse function tells us how to translate back from labels in T to elements of the original set S.

Exercise 41. Show that a function $f\colon S \longrightarrow T$ is a bijection if and only if it has an inverse function. *Hint: The proof of the preceding proposition gives you almost half of the solution.*

Exercise 42. Let $f\colon S \longrightarrow T$ be a function. Let $f[S]$ be the image of S under f in T (also known as the range of f). We may define a function f' as follows.

$$\begin{aligned} f'\colon S &\longrightarrow f[S] \\ s &\longmapsto fs. \end{aligned}$$

Show that if f is injective then f' is a bijection.

Example 2.23. Consider the function

$$\begin{aligned} f\colon \mathbb{Z} &\longrightarrow \mathbb{Z} \\ x &\longmapsto x+1. \end{aligned}$$

In Example 2.21 it is shown that this function is a bijection. The preceding exercise tells us that this function must have an inverse function, and we calculate the inverse here.

To give this inverse we need to find a function which 'undoes' what f does, and the obvious candidate for this is the function g given by

$$x \longmapsto x-1.$$

We show that g is indeed the inverse function for f. Assume that $n \in \mathbb{Z}$. We calculate

$$\begin{aligned} (g \circ f)n &= g(fn) && \text{def function composition} \\ &= g(n+1) && \text{def } f \\ &= (n+1)-1 && \text{def } g \\ &= n && \text{arithmetic} \end{aligned}$$

and so we know that $g \circ f = \mathsf{id}_Z$. We also have to show that the other composite is the identity, so again assume we have $n \in \mathbb{Z}$. We calculate

$$\begin{aligned} (f \circ g)n &= f(g(n)) && \text{def function composition} \\ &= f(n-1) && \text{def } g \\ &= (n-1)+1 && \text{def } f \\ &= n && \text{arithmetic,} \end{aligned}$$

so we also have $f \circ g = \mathsf{id}_Z$, and both equalities together tell us that g is indeed the inverse function for f.

Exercise 43. Calculate the inverse for the function from $\mathbb{C}$ to $\mathbb{C}$ given by $x \longmapsto 2xi$. Use the inverse function to show that the given function is surjective.

Exercise 44. Recall from Exercise 28 the set $\mathsf{Fun}(S,S)$ of all functions from a set S to itself. We define a subset of this set

$$\mathsf{Bij}(S,S) = \{f \in \mathsf{Fun}(S,S) \mid f \text{ is a bijection}\}.$$

Show that the composite of two bijections is a bijection. This means that we can use function composition to define a binary operation

$$\mathsf{Bij}(S,S) \times \mathsf{Bij}(S,S) \longrightarrow \mathsf{Bij}(S,S),$$

which is again a monoid. Show that the inverse function of an element of $\mathsf{Bij}(S, S)$ which is known to exist by Exercise 41 is its inverse with respect to the function composition operation. Conclude that $\mathsf{Bij}(S, S)$ is a group (under the composition operation).

Glossary

absolute, $|\cdot|$ 16, 44

Defined for various sets of numbers, here extended to complex numbers. Given a complex number $a + ib$ we have $|a + ib| = \sqrt{a^2 + b^2}$.

and 54

Connects two properties or statements, both of which are expected to hold.

anti-symmetric 176

A binary relation R on a set S is anti-symmetric if (s, s') and (s', s) both being in the relation implies $s = s'$.

argument 47

The argument of a complex number is the angle it encloses with the positive branch of the real axis.

associative 65

A binary operation is associative if and only if it gives the same result when applied two three inputs, no matter whether it is first applied to the first two, or first applied to the last two of these.

bijective 81

A function is bijective if and only if it is both, injective and surjective. A bijective function is called a bijection.

binary operation 32, 65

A function of the type $S \times S \longrightarrow S$, which takes two elements of a set S as input and produces another element of S.

$\mathbb{C}$ 43

The complex numbers as a set with a number of operations.

commutative 68

A binary operation is commutative if and only if it gives the same result when its two inputs are swapped.

complement 26

The complement of a set S is always taken with respect to an underlying set, and it consists of those elements of the underlying set which do not belong to S.

complete binary tree with labels from a set S 134

This is a tree where each node has a label from S and where each node has either 0 or 2 children. Formally this is another recursively defined notion.

composite of functions 34

The composite of two functions is defined provided the target of the first is the source of the second. It is the function resulting from taking an element of the source of the first function, applying the first function, and then applying the second to the result.

composite of partial functions 149

Similar to the composite of two functions, but the result is undefined if either of the two functions is not defined where required.

conjugate, $\overline{\cdot}$ 49

The conjugate $\overline{z}$ of a complex number number $z = a + ib$ is $a - ib$.

countable 111

A set is countable if and only if there is an injective function from it to $\mathbb{N}$.

countably infinite 111

A set is countably infinite if it is both, countable and infinite.

definition by cases 40

A way by piecing together functions to give a new function.

degree of a polynomial 174

The degree of a polynomial is the largest index whose coefficient is unequal to 0.

directed graph 145

A set (of *nodes*) connected by edges; can be described using a binary relation on the set.

divides 17

A number m divides a number n in some set of numbers if there exists a number k with the property that $n = km$.

divisible 17

We say for natural numbers (or integers) that n is divisible by m if and only if n leaves remainder 0 when divided by m using integer division.

domain of definition 148

For a partial function it is the set consisting of all those elements of the source set for which the partial function is defined.

dominate 112

A function f from a set X to $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ or $\mathbb{R}$ dominates another g with the same source and target if and only if the graph of f lies entirely above the graph of g (graphs touching is allowed).

equivalence class with respect to R generated by s, $[s]$ 161

The set of all elements which are related to s by the equivalence relation R.

equivalence relation 159

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

equivalence relation generated by a binary relation R 160

The transitive closure of the symmetric closure of the reflexive closure of R.

even 17

An integer (or natural number) is even if and only if it is divisible by 2.

eventually dominate 114

A function f from $\mathbb{N}$ to $\mathbb{N}$ eventually dominates another g with the same source and target if and only if there is some number beyond which the graph of f lies above that of g (graphs touching is allowed). The analogous definition works for functions with source and target $\mathbb{Z}$, $\mathbb{Q}$ or $\mathbb{R}$.

for all 58

Expresses a statement or property that holds for all the entities specified.

function 33

A function has a source and a target, and contains instructions to turn an element of the source set into an element of the target set. Where partial functions are discussed sometimes known as **total function**.

graph of a function 36, 73

The graph of a function f with source S and target T consists of all those pairs in $S \times T$ which are of the form (s, fs).

greatest element, $\top$ 181

An element which is greater than or equal to every element of the given poset.

greatest lower bound, infimum 184

An element of a poset $(P, \leq)$ is a greatest lower bound of a given subset of that poset if it is both, a lower bound and greater than or equal to every lower bound of the given set.

group 73

A set with an associative binary operation which has a unit and in which every element has an inverse.

identity function 34

The identity function on a set is a function from that set to itself which returns its input as the output.

identity relation 144

The identity relation on a set S relates every element of s to itself, and to nothing else.

if and only if 57

Connects two properties or statement, and it is expected that one holds precisely when the other holds.

image of a set, $f[\cdot]$ 35

The image of a set consists of the images of all its elements, and one writes $f[S]$ for the image of the set S under the function f.

image of an element 35

The image of an element under a function is the output of that function for the given element as the input.

imaginary part 43

Every complex number $a + bi$ has an imaginary part b.

implies 56

Connects two properties or statements, and if the first of these holds then the second is expected to also hold.

infinite 110

A set is infinite if and only if there is an injection from it to a proper subset.

injective 74

A function is injective if and only if the same output can only arise from having the same input. An injective function is called an injection.

integer 16

A whole number that may be positive or negative.

integer division, div 15, 17

The integer $n \operatorname{div} m$ is defined as the number of times m divides n (leaving a remainder).

intersection, $\cap$ 25

The intersection of two sets S and T is written as $S \cap T$, and it consists of all the elements of the underlying set that belong to both, S and T, The symbol $\bigcap$ is used for the intersection of a finite number of sets.

inverse 72

One element is the inverse for another with respect to a binary operation if and only if when using the two elements as inputs (in either order) to the operation the output is the unit.

inverse function 82

A function is the inverse of another if and only if the compose (either way round) to give an identity function.

least element, $\bot$ 182

An element which is less than or equal to every element of the given poset.

least upper bound, supremum 184

An element of a poset $(P, \leq)$ is a least upper bound of a given subset of that poset if it is both, a upper bound and less than or equal to every upper bound of the given set.

list over a set S 129

A list over a set S is a recursively defined concept consisting of an ordered tuple of elements of the given set.

lower bound 183

An element of a poset $(P, \leq)$ is a lower bound for a given subset of P if it is less than or equal to every element of that set.

maximal element 181

An element which does not have any elements above it.

minimal element 181

An element which does not have any elements below it.

monoid 71

A set with an associative binary operation which has a unit.

$\mathbb{N}$ 13, 119

The natural numbers as a set with a number of operations. This set and its operations are formally defined in Section 5.1.

natural number 13

One of the 'counting numbers', 0, 1, 2, 3,....

odd 17

An integer (or natural number) number is odd if it is not even or, equivalently, if it leaves a remainder of 1 when divided by 2.

opposite relation of R, R^{op} 144

The relation consisting of those pairs (t, s) for which (s, t) is in R.

or 55

Connects two properties or statements, at least one of which is expected to hold.

ordered binary tree with labels from a set S 135

Such a tree is ordered if the set S is ordered, and if for every node, all the nodes in the left subtree have a label below that of the current node, while all the nodes in the right subtree have a label above.

partial order 177

A binary relation on a set is a partial order provided it is reflexive, anti-symmetric and transitive.

polynomial equation 21

An equation of the form $\sum_{i=0}^{n} a_i x^i$.

polynomial function 37

A function from numbers to numbers whose instruction is of the form x is mapped to $\sum_{1i=1}^{n} a_i x^i$ (where the a_i are from the appropriate set of numbers).

poset 177

A set with a partial order, also known as a *partially ordered set*.

powerset, $\mathcal{P}$ 32

The powerset of a set S is the set of of all subsets of S.

prime 64

A natural number or an integer is prime if its dividing a product implies its dividing one of the factors.

product of two sets 31

A way of forming a new set by taking all the ordered pairs whose first element is from the first set, and whose second element is from the second set.

proper subset 24

A set S is a proper subset of the set T if and only if S is a subset of T and there is at least one element of T which is not in S.

$\mathbb{Q}$ 18

The set of all rational numbers together with a variety of operations, formally defined in Chapter 6.

$\mathbb{R}^+$ 20

The set of all real numbers greater than or equal to 0.

$\mathbb{R}$ 20

The set of all real numbers.

range of a function 35

The range of a function is the set of all elements which appear as the output for at least one of the inputs, that is, it is the collection of the images of all the possible inputs.

rational number 18

A number is rational if it can be written as the fraction of two integers. A formal definition is given on page 173 (and the preceding pages).

real number 20

We do not give a formal definition of the real numbers in this text.

real part 43

Every complex number $a + bi$ has a real part a.

reflexive 150

A binary relation on a set is reflexive if it relates each element of the set to itself.

reflexive closure 151

The reflexive closure of a binary relation on a set S is formed by adding all pairs of the form (s, s) to the relation.

relational composite 144

A generalization of composition for (partial) functions.

remainder for integer division, mod 15

The integer n mod n is defined to be the remainder left when dividing n by m in the integers.

set difference, $\setminus$ 26

The set difference $S \setminus T$ consists of all those elements of S which are not in T.

size of a set 109, 110

A set is smaller than another if there exists an injective function from the first to the second. They have the same size if they are both smaller than the other.

string over a set S 136

A formal word constructed by putting together symbols from S.

surjective 78

A function is surjective if and only if every element of the target appears as the output for at least one element of the input. This means that the image of the function is the whole target set. A surjective function is called a surjection.

symmetric closure 152

The symmetric closure of a binary relation on a set is formed by taking the union of the relation with its opposite.

there exists 59

Expresses the fact that a statement or property holds for at least one of the entities specified.

total order 178

A total order is a partial order in which every two elements are comparable.

transitive 153

A binary relation on a set S is transitive provided that (s, s') and (s', s'') being in the relation implies that (s, s'') is in the relation.

transitive closure 154

The transitive closure of a binary relation on a set is formed by adding all pairs of elements (s_1, s_n) for which there is a list of elements s_1, s_2 to s_n in S such that (s_i, s_{i+1}) is in the relation.

union, $\cup$ 25

The union of two sets S and T is written as $S \cup T$. It consists of all elements of the underlying set that belong to at least one of S and T. The symbol $\bigcup$ is used for the union of a finite number of sets.

uncountable 111

A set is uncountable if it is not countable.

unique existence 60

A more complicated statement requiring the existence of an entity, and the fact that this entities is unique with the properties specified.

unit 70

An element of a set is a unit for a binary operation on that set if and only if applying the operation to that, plus any of the other elements, returns that other element.

upper bound 183

An element of a poset $(P, \leq)$ is an upper bound for a given subset of P if it is greater than or equal to every element of that set.

$\mathbb{Z}$ 16, 170

The integers with various operations, formally defined on page 170.

COMP11120, Semester 1

Exercise Sheet 0

For examples classes in Week 1

Core Exercises marked this week

Exercise 6

Exercise 7

Exercise 8

Extensional Exercises marked this week

Exercise 5

Exercise 9

Remember that

- the **deadline** is the beginning of the examples class, and that you have to be able to promptly answer questions by the TA, referring to your rough work as needed;
- you may only use concepts which are defined in these notes (Chapter 0 establishes concepts for numbers), and for every concept you do use you should find the definition in the notes and work with that;
- you should justify each step in your proofs;
- if you are stuck on an exercise move on to the next one after ten minutes, but write down why you got stuck so that you can explain that to the TA in the examples class. If you couldn't get started then note down all the relevant definitions (use the Glossary to find these),

You should make sure this week that you understand the content and in particular the notation used in Chapter 0

COMP11120, Semester 1

Exercise Sheet 1

For examples classes in Week 2

Core Exercises marked this week

Exercise 11

Exercise 15

Exercise 20 Carry out your proof in the style of that given on page 45 as far as you can.

Extensional Exercises marked this week

Exercise 17 Carry out your proof in the style of that given on page 45 as far as you can.

Exercise 18

Remember that

- the **deadline** is the beginning of the examples class, and that you have to be able to promptly answer questions by the TA, referring to your rough work as needed;
- you may only use concepts which are defined in these notes (Chapter 0 establishes concepts for numbers), and for every concept you do use you should find the definition in the notes and work with that;
- you should justify each step in your proofs;
- if you are stuck on an exercise move on to the next one after ten minutes, but write down why you got stuck so that you can explain that to the TA in the examples class. If you couldn't get started then note down all the relevant definitions (use the Glossary to find these),

Exercises you could potentially do this week are all those in Chapter 1.

COMP11120, Semester 1

Exercise Sheet 2

For examples classes in Week 3

Core Exercises marked this week

Exercise 21. Do three of the parts, one from (a)–(c) and two from (d)–(g).

Exercise 24. Do three of the parts, one from (a)–(d), one from (e)–(f) and one from (g)–(i).

Exercise 25. Do two of the parts, one from (a)–(d) and one from (e)–(g).

Extensional Exercises marked this week

Exercise 26. Do two of the parts, one from (a)–(b) and one from (c)–(e). If you prefer you may do **Exercise 28** instead.

Exercise 30.

Remember that

- the **deadline** is the beginning of the examples class, and that you have to be able to promptly answer questions by the TA, referring to your rough work as needed;
- you may only use concepts which are defined in these notes (Chapter 0 establishes concepts for numbers), and for every concept you do use you should find the definition in the notes and work with that;
- you should justify each step in your proofs;
- if you are stuck on an exercise move on to the next one after ten minutes, but write down why you got stuck so that you can explain that to the TA in the examples class. If you couldn't get started then note down all the relevant definitions (use the Glossary to find these),

Exercises you could do this week or those in Sections 2.1 to 2.3.2.

COMP11120, Semester 1

Exercise Sheet 3

For examples classes in Week 4

Core Exercises marked this week

Exercise 32. Do three of the parts, one from (a)–(c), one from (d)–(f). and one from (g)–(i). *Hint: If you find this hard then try to do the previous exercise first, where yu know what the answer is in each case.*

Exercise 36. Do three of the parts, one from (a)–(d), one from (e)–(f), and one from (g)–(h). *Hint: If you find this hard then try to do the previous exercise first, where yu know what the answer is in each case.*

Exercise 38. Do two of the parts, one from (a)–(c) and one from (d)–(f).

Extensional Exercises marked this week

Exercise 33. Do any three parts.

Exercise 43.

Remember that

- the **deadline** is the beginning of the examples class, and that you have to be able to promptly answer questions by the TA, referring to your rough work as needed;
- you may only use concepts which are defined in these notes (Chapter 0 establishes concepts for numbers), and for every concept you do use you should find the definition in the notes and work with that;
- you should justify each step in your proofs;
- if you are stuck on an exercise move on to the next one after ten minutes, but write down why you got stuck so that you can explain that to the TA in the examples class. If you couldn't get started then note down all the relevant definitions (use the Glossary to find these),

Exercises you could do this week are those in Section 2.3.3.