A visual companion to proofs

Michael Shulman

Draft of February 12, 2025

Contents

0 Introduction **4**

1 Algebraic proofs **11**
1.1 The rules of algebra . . . 12
1.2 About examples . . . 14
1.3 On working forwards . . . 16
1.4 Chaining equalities . . . 17
1.5 Number systems . . . 20
1.6 A digression about counting . . . 31
1.7 Inequalities . . . 33
Exercises . . . 34

2 Propositional proofs **37**
2.1 The algebra of truth values . . . 37
2.2 What is a proof? . . . 38
2.3 And ($\wedge$) . . . 39
2.3.1 Introduction to Olorin . . . 41
Exercises . . . 42
2.4 $\wedge$ with algebra . . . 42
Exercises . . . 44
2.5 Or ($\vee$) . . . 45
Exercises . . . 48
2.6 $\vee$ with algebra . . . 49
Exercises . . . 55
2.7 If-then ($\Rightarrow$) . . . 56
Exercises . . . 60
2.8 $\Rightarrow$ with algebra . . . 61
2.9 If and only if ($\Leftrightarrow$) . . . 62
Exercises . . . 63
2.10 Truth value constants ($\top$ and $\bot$) . . . 64
Exercises . . . 66

3 Quantifier proofs **67**
3.1 Free and bound variables . . . 67
Exercises . . . 72
3.2 Σ and Π . . . 73
Exercises . . . 77
3.3 Quantifiers (∀ and ∃) . . . 78
Exercises . . . 80
3.4 For all (∀) . . . 81
Exercises . . . 83
3.5 ∀ with algebra . . . 84
Exercises . . . 86
3.6 There exists (∃) . . . 88
Exercises using ∃ . . . 89
Exercises combining ∀ and ∃ . . . 90
3.7 ∃ with algebra . . . 91
Exercises . . . 93
3.8 Unique existence . . . 96
3.9 Implicit universals and implications . . . 96
Exercises . . . 98

4 Negation and contradiction **99**
4.1 Not (¬) . . . 99
Exercises . . . 102
4.2 The algebra of negation . . . 103
4.3 ¬ with algebra . . . 108
Exercises . . . 110
4.4 Constructivity . . . 110
Exercises . . . 112

5 Recursion and induction **113**
5.1 Recursive definitions . . . 113
Exercises . . . 116
5.2 Inductive proofs . . . 116
Exercises . . . 123
5.3 More general induction . . . 125
Exercises . . . 131
5.4 Strong vs weak induction . . . 132

A Principles of English Proof **134**

B Proof Guidance **136**

Chapter 0

Introduction

This introduction is written primarily for instructors, but students will benefit from reading it as well, particularly the first couple of sections.

Formal proofs

Proof is arguably the most unique feature of mathematics, the characteristic that distinguishes it from all other human activities. Mathematics shares experimental and theoretical aspects with the sciences; it shares application and design principles with engineering; it shares beauty and aesthetic sensibilities with the arts; it shares rhetorical and narrative concerns with literature; it shares exploration and strategy with gaming; it shares precision and computation with computer science; but no other human pursuit involves *rigorous deductive proof*. When a mathematical theorem is proven, if the proof is correct, then the theorem is true — necessarily, unquestionably, undeniably true. In no other sphere is there such absolute certainty.

Given the centrality of proof to mathematics, it is not surprising that there are now many textbooks intended to teach students how to write proofs. What is surprising, however, is that almost none of these books tell students *what a proof is*.[1] In defiance of the mathematical principle that *a collection of examples is not a definition*, nearly all "introduction to proof" books simply supply a collection of "proof techniques" without explaining what exactly it is that these techniques produce. The student learns about "direct proof" and "contrapositive proof", about "proof by cases" and "proof by contradiction", about "proof by induction" and "proof by strong induction", and often comes away with an impression that a proof is just a "convincing argument" and mathematicians are always making up new proof techniques, so you can never be sure what you're allowed to do.

Nothing could be further from the truth. A proof is a formal object that is constructed according to a *fixed, finite* set of rules. Some of these rules coincide

[1] Notable exceptions include Velleman's *How To Prove It* [Vel19] and Newstead's *An Infinite Descent Into Pure Mathematics* [New], which have both been great inspirations to me.

with some of the "techniques" in the average introductory text, but others are rarely mentioned there at all, while a number of the common "techniques" are not primitive and can be derived from the basic rules. There are many ways to *represent*, *encode*, or *describe* a formal proof object, but at root there is a complete and unchanging specification of what a proof is, and once a proof is fully specified there is a definite method to determine whether it is correct.

Of course, this statement comes with many caveats. It is true that mathematicians wrote proofs for many centuries before anyone fully understood what a proof is; the formal notion of proof was only discovered in the late 19th century by George Boole, Augustus De Morgan, Charles S. Peirce, Gottlob Frege, and others. It is also true that to this day, most mathematicians do not think about, or probably even understand, the formal notion of proof; instead they have an intuitive understanding of what a proof is, developed from examples, which serves *them* sufficiently well. This is possible because mathematicians also rarely *work* directly with fully specified formal proofs (although this is becoming more common with the advent of computer proof assistants); usually they *describe* proofs in a natural human language such as English, and the level of precision and detail in such descriptions varies widely according to the intended audience. Finally, it's now known that there is not only *one* notion of formal proof, but a wide variety of "logics" suitable for different purposes.

All of those caveats notwithstanding, I contend that students being exposed to proof for the first time will benefit from learning exactly *what a proof is*. There are many advantages to this. It takes the guesswork out of verifying proofs: once you know the rules, you can determine for yourself whether your proof (or someone else's) is valid, simply by checking whether it follows the rules. It makes it easier to *remember* the rules, since when they are all written down precisely, the rules of proof are not haphazard: they are organized cleanly according to the logical operators (connectives and quantifiers), with some number of "rules to prove" and some number of "rules to use" associated to each operator. And, perhaps most importantly, it provides *guidance* when *writing* a proof: by inspecting the logical form of the givens and the goal, you immediately obtain a list of proof rules that can be applied.

Graphical proofs

To teach proofs in this way, we need to choose a specific representation of proofs. In these notes I will use a *graphical* representation that seems, to me, the most compelling and the easiest to understand for the most students compared to other notions of formal proof. It is a variant of the graphical representation used by The Incredible Proof Machine [Bre16], which is closely related to string diagrams [Pen71, JS91] and proof nets [Gir87] as well as to Gentzen's "natural deduction" proof trees.

I should emphasize right away that the *end goal* is still for the student to write and read proofs written in "mathematical English" rather than in the graphical notation. For example, when I teach proofs in this way, there are no

graphical proofs allowed on the final exam (except as scratchwork), only English proofs. The role of the graphical proofs is to help students understand what a proof *fundamentally is*, but in the end the usefulness of this understanding is "local", in terms of what rules can be applied in what situations and what their effects are. We move away from actually writing complete graphical proofs as quickly as possible: partly because we want to learn to communicate with other mathematicians, who uniformly write proofs in English; but also because fully formal proofs (in any representation) quickly become extremely large, making them impractical to use for all but the smallest exercises (except with the help of a computer proof assistant).

With this in mind, in these notes I will write English proofs alongside graphical ones, pointing out the conventions of how specific English words and phrases are used to describe certain rules of a formal proof. Feedback from past students suggests that what they find the most helpful is reading and watching proofs written with graphical and English versions in parallel, so I encourage instructors to do this in class as well, for as long as is feasible. The instructor can choose whether to ask students to start writing their own proofs in English right away (as well as graphically), or to have them stick with only graphical proofs for a short while.

For inspiration, here are some (slightly edited) quotes and advice from some of my previous students who learned proofs in this style:

> "Due to this class, I feel as though my ability to visualize proofs has improved, and I really enjoyed the graphical proofs and see them as the greatest tools that led to this."

> "The graphical proofs are helpful, especially for visual learners like myself — it may feel unique/different, but it translates well into the English proofs."

> "For future students, I would recommend really being able to understand graphical proofs, because if you can, then the English proofs should just come along almost just by the way."

> "I would tell future students to spend the time learning the graphical proofs, as if you know the graphical proofs, you will know inherently know how to do the English proofs."

Proof assistants

Another advantage of formal proofs is that they can be verified by a computer program. Since there is a fixed, finite, set of proof rules, when a proof is written out completely formally using these rules, the computer can simply check step-by-step whether all the rules are correctly used. Such a program usually provides additional help and support to the user in writing out formal proofs in the first place, and so it is known as a *proof assistant*.

Computer proof assistants have been around for decades, but they have recently enjoyed a boom in popularity among mathematicians for a number of reasons, one of which is improvements in usability. It's true that formalizing mathematics in a proof assistant is still significantly more work than writing in on paper, but the difference is shrinking, and the outcome of the formalization is more reliable since every step in the proof is verified by the computer. In addition, the formal languages of mathematics used by proof assistants are, in many cases, also programming languages, and so formalized proofs intertwine naturally with a computational approach to mathematics that is growing in importance. Indeed, in recent years several high-profile proofs, such as the proof of the Four-Color Theorem by Appel–Haken, and the proof of the Kepler Conjecture by Hales, depend on computer calculations too extensive for a human to check by hand, and were not completely accepted by the mathematical community until they were formalized using a proof assistant.

It is not the purpose of these notes to teach students the "industrial-strength" proof assistants that are used to formalize serious mathematics of this sort, although I hope and believe that the solid grounding in proofs I do intend to impart will stand them in good stead if and when they do decide to learn these tools. However, using a specially designed proof assistant for simple proof exercises can also be pedagogically helpful. In addition to providing students immediate feedback on their work, this allows proof-writing to be treated as a sort of video game, where students can learn "the rules of the world" by experimenting, and develop strategies by exploration. The computer only permits valid "moves", but gives the user complete freedom to apply those rules in ways that may or may not lead towards the goal. This can be very beneficial, helping students to learn and practice the rules organically on their own, and clarifying the distinction between the "science" of a proof *is* and the "art" of *constructing* a proof.

However, the use of a computer proof assistant has dangers as well. In particular, some students have a tendency to treat each problem in the proof assistant as an independent mystery, to be solved by sticking things together randomly until the light turns green, rather than developing a true understanding of the rules and using that to solve each problem systematically. This is one reason that, when I teach proofs in this way, I have the students use a proof assistant early on in the semester, but transition away from it even before we switch entirely to English proofs. Fortunately, graphical proofs are also easy to draw by hand on paper (or whiteboards, blackboards, or electronic tablets).

There are two available proof assistants for the graphical proofs used in these notes. The original one, which inspired our graphical proofs, is *The Incredible Proof Machine* (`https://incredible.pm/`) by Joachim Breitner. However, it doesn't support quantifiers (chapter 3) in the style that I prefer, so for that reason (and others) I am developing an alternative. Its working title is *Olorin*. Olorin also comes with three "difficulty settings" that attempt to transition the student gradually away from relying on the proof assistant as a crutch and towards independence in proof-writing. I recommend using Olorin for all the exercises that it can handle; please report all bugs you find, and let me know of

any suggestions you have to make it better! A more extensive introduction to Olorin can be found in section 2.3.1, as well as behind its own "About" button.

Drill exercises

Another principle on which these notes are based is that proof-based mathematics, like calculational mathematics, and like other pursuits such as playing a sport or a musical instrument, is based on fundamental *skills* that can be developed through *drill*. Textbooks for more calculational parts of mathematics, such as algebra and calculus, usually include a lot of drill exercises; but these are rare in proof textbooks. Perhaps it is more difficult to isolate the basic skills in the absence of the organizing principles of formal proof.

In any case, in these notes I have endeavored to provide a large number of drill exercises covering each of the basic proof skills. This does admittedly require some care in selecting a subject matter that can produce that many exercises. I have tried to keep the prerequisites minimal by staying as close to algebra as possible, while also trying to supply a bit of exposure to fun concepts that students may not have encountered before, like surreal numbers, modular arithmetic, and p-adic numbers. In most cases, the principle underlying the drill exercises should be clear enough that an instructor should be able to produce more if needed.

In my experience, after being "taught" each skill, students require several iterations of trying to do it themselves, receiving feedback, and trying again, before they finally master it. Each of these iterations could be in-class group work, homework, traditional quizzes, standards-based quizzes, or some other method; but *repeated* feedback does seem to be necessary, with enough exercises available to provide material for repeated attempts. It would be much easier on the instructor if this feedback could be automated somehow, but as of this writing in January 2025, my experiments show that even the most advanced Large Language Models cannot provide feedback on attempted proofs that is reliably both correct and helpful.

Strong typing

Another complaint I have with most introduction-to-proof textbooks is that they introduce quantifiers $\forall$ and $\exists$ without a set to bound the variable: $\forall x$ rather than $\forall x \in A$. This is odd, because the latter is what is used almost universally in mathematics. More generally, nearly all mathematics is *strongly typed*: every variable has a declared "type" (the set or collection of its possible values) and we never mix elements of distinct sets. I believe it is important for students to start thinking about mathematics in this way from the start, so in these notes I introduce quantifiers as $\forall x \in A$ and $\exists x \in A$ right away.

For the set-theoretically savvy, note I don't require that A be a *set* — it can be an arbitrary "collection" or "class". Thus, the point is not about "bounded"

versus "unbounded" quantifiers in the sense of axiomatic set theory; it's really just about type discipline.

Strong induction

My last complaint is that most textbooks teach "weak induction" first (from k to $k+1$) and only later introduce "strong induction". This seems unnecessarily complicated, given that so many important examples require strong induction, and weak induction is an easy special case of strong induction. I believe it is just as easy to motivate strong induction directly as to motivate weak induction, so in these notes "induction" means what is usually called "strong induction". (But I do inform students about the usual strong/weak distinction in the brief section 5.4, since they will encounter it elsewhere.)

Need for a supplement

The current version of these notes is not intended as a standalone textbook. In particular, it does not contain very much mathematical *content* beyond proofs. Individual instructors (or departments) usually have preferences or requirements about what additional material should be covered in their introduction to proofs course, and should choose an additional textbook that covers that material. In particular, sets, functions, and relations (particularly equivalence relations) are common requirements; but it's also a good idea to include something more intrinsically interesting.

This other material should also provide a context to expose students to a variety of proofs, which can be more interesting, more complicated, and more beautiful than the ones they can reasonably be expected to create themselves (which these notes focus on), but which clearly use the same rules and the same structure. I believe such "proof appreciation" is also very important, for students to get a sense of the scope and beauty of mathematics, and how the proof skills they are learning can be used to establish nontrivial truths.

In addition, an unrelenting focus on nothing but the "nuts and bolts" of proof writing can be boring and disheartening. I believe it's important to *motivate* the learning of proofs with interesting applications. Number theory is an especially good match for these notes, since divisibility and modular arithmetic already appear in examples and exercises; but any desired application can be used.

Finally, there are other activities that should arguably also be included in an introduction-to-proofs course, such as exploration and experimentation. These notes do not provide these, although some of the material I discuss naturally fits with some such activities.

Here are a few textbooks that I can tentatively recommend as supplements. Please don't be offended if your favorite doesn't appear!

- Daniel J. Velleman, *How To Prove It*.

- Clive Newstead, *An Infinite Descent Into Pure Mathematics*
- Jay Cummings, *Proofs: A Long-Form Mathematics Textbook*. If you are using Cummings' book, I recommend working through chapters 1, 2 and 3 of my notes first, along with his Appendix C which is an excellent collection of advice for writing English proofs. Then you can read his chapters 1, 2, and 3, being aware that at that point he is already secretly using the rules for quantifiers without having introduced them. His chapter 4 on induction is weak-first, so I recommend working through my chapter 5 first, but then read his for the nice examples. Skip his chapter 5 completely (it's replaced by my chapters 1–3). His chapters 6 and 7 are roughly equivalent to my chapter 4 and you can read them in either order, and his chapters 8 and 9 are then a good conclusion.
- Joel David Hamkins, *Proof and the Art of Mathematics*
- Edward B. Burger, *Extending the Frontiers of Mathematics: Inquiries into proof and argumentation*
- Sebastian M. Cioabă and Werner Linde, *A Bridge to Advanced Mathematics: From Natural to Complex Numbers*
- Daniel Solow, *How to Read and Do Proofs*. This book is focused mainly on proof-writing rather than mathematical content, but you may still find it interesting.
- George R. Exner, *An accompaniment to higher mathematics*. Also focused mainly on proof-writing rather than mathematical content, but you may still find it interesting.

The author was supported by the Air Force Office of Scientific Research under award number FA9550-21-1-0009.

Chapter 1

Algebraic proofs

What is a proof? Over the course of these notes, we will give a complete definition of what a proof is and how to construct one. That doesn't mean that at the end of this course you will be able to immediately prove any desired statement, but you will know *how* to go about *trying* to prove it, and how to tell whether you have succeeded. Indeed, many mathematicians spend their whole lives trying, and failing, to prove some desired statement! *Finding* proofs is an art; but an artist must train with their tools systematically before they can use them creatively. The goal of these notes is to train you systematically in the tools of proof.

We will start small, with simple proofs, and build up towards more complicated ones. In any proof, the object is to start from certain statements considered as known to be true — called *hypotheses*[1], *assumptions*, or *givens* — and show that another claimed statement is also true — called the *conclusion* or the *goal*. The *rules* of proof tell us what sort of steps we are allowed to take to get from here to there.

The statement that we prove, specifying all the hypotheses and the conclusion, is called a *theorem*. Sometimes we use other words like *lemma* (a theorem that we proved mainly because it will help us prove something else) or *corollary* (a theorem that follows very quickly from a more difficult theorem we just proved).[2] In general, to say that a theorem is true means that *whenever* the hypotheses are true, so is the conclusion (we will return to this in chapters 2 and 3).

[1] "Hypotheses" is a plural; the singular is "hypothesis".

[2] Some people also use the word *proposition* for a theorem that they don't think is important enough to dignify with the name "theorem". I prefer to avoid this because it conflicts with the use of the word *proposition* in logic to mean a statement that could be either true *or* false (whereas a *theorem*, dignified or not, is *always* true, since it has been proven).

1.1 The rules of algebra

In this chapter we consider a kind of proof that you are likely already sort of familiar with, which we call *algebraic* proof. In an algebraic proof, all the statements involved are *equations* (or, in some cases, *inequalities*) between expressions, often involving variables. And the rules we use are those that you are probably already familiar with from algebra, such as:

1. You can add or subtract the same thing to both sides of a true equation to produce another true equation. For instance, if $x = y$ is true, then $x + 1 = y + 1$ is also true.

2. You can multiply both sides of a true equation by the same thing to produce another true equation. For instance, if $x = y$ is true, then $x^2 = xy$ is also true.

3. You can divide both sides of a true equation by the same thing to produce another true equation, as long as the thing divided by is *nonzero*. (There is a caveat to this that we will come back to this in section 1.5.) For instance, if $x^2 = 2x$ is true and $x \neq 0$ in ordinary algebra, then $x = 2$ is also true.

4. You can apply any function to both sides of a true eqution to produce another true equation, as long as both sides are in the domain of the function. For instance, if $x = y + 1$ is true, then $\sin(x) = \sin(y + 1)$ is also true.

5. You can simplify either or both sides of a true equation using the rules of algebra and the equation will remain true. For instance, if $y = x + 3 - 3$ is true, then $x = y$ is also true; and if $yz - 1 = \frac{x^2+x}{x}$ is true, then $yz - 1 = x + 1$ is also true. (Of course, in the latter case, x had to be nonzero before $\frac{x^2+x}{x}$ could even be written.)

6. You can add, subtract, or multiply the corresponding sides of two true equations to obtain a new true equation. For instance, if $x - 1 = y + 1$ and $z = x + y$ are true, then $(x - 1)z = (y + 1)(x + y)$ is also true.

7. You can substitute one side of a true equation by the other side of that equation, anywhere in another true equation, to obtain a third true equation. For instance, if $x = 3$ and $xy = y + 1$ as true, then $3y = y + 1$ is also true. Often the side we substitute for is a simple variable, but it doesn't have to be: if $x + 1 = y$ and $3(x + 1) = 2 - z$ are true, then $3y = 2 - z$ is also true.

8. Any expression is equal to itself. For instance, $x + 2 = x + 2$ is true.

9. A true equation can be reversed and remain true. For instance, if $x^2 - 3 = y + 1$ is true, then $y + 1 = x^2 - 3$ is true.

10. Two true equations that "match in the middle" can be "chained" to produce a third true equation. For instance, if $x = y$ and $y = z$ are true, then $x = z$ is also true.

For example, here is a theorem that we can prove algebraically.

Theorem 1.1. *Suppose $x + 1 = y$ and $x - 1 = z$. Then $x^2 = yz + 1$.*

We call this "Theorem 1.1" because it is theorem number 1 in chapter 1. To prove this theorem, we can first multiply the two given equations together to get $(x + 1)(x - 1) = yz$. Then we can simplify the left-hand side to get $x^2 - 1 = yz$. After that we can add 1 to both sides to get $x^2 - 1 + 1 = yz + 1$, and finally simplify the left-hand side again to get $x^2 = yz + 1$, which is the desired conclusion.

The preceding paragraph is a perfectly fine way to describe a proof of this theorem. However, it's a bit wordier than necessary: when we are just going from one true equation to another one by simplifying, substituting, or operating on both sides, a reader familiar with the rules of algebra can usually figure out what we did just by seeing the two equations. When we proceed through a sequence of equations like this, we usually line up their equals signs vertically. We only need to add words if we are doing something less obvious, like adding or multiplying two equations. Thus, we would more usually write a proof of the above theorem like this:

Proof of 1.1. Multiplying the two given equations, we obtain

$$\begin{aligned} (x+1)(x-1) &= yz \\ x^2 - 1 &= yz \\ x^2 - 1 + 1 &= yz + 1 \\ x^2 &= yz + 1. \end{aligned}$$

□

The symbol "□" denotes the end of a proof. Some people use other symbols for this, or the abbreviation "Q.E.D." which stands for the Latin *quod erat demonstrandum*, "which was to be proved".

It is sometimes permissible to combine multiple steps in one. For instance, very frequently we "move something from one side to another" by adding or subtracting it and then immediately canceling it from the side where it originally appeared, as with the $-1 + 1$ above. In this case we commonly omit the cancellation step, so the equations in the above proof could also be written

$$\begin{aligned} (x+1)(x-1) &= yz \\ x^2 - 1 &= yz \\ x^2 &= yz + 1. \end{aligned}$$

Here are some more examples.

Theorem 1.2. *Suppose $xy + 2 = 14 - y^2$ and $y = 3x$. Then $|x| = 1$.*

Proof of 1.2. Substituting $y = 3x$ into $xy + 2 = 14 - y^2$, we obtain

$$\begin{aligned} x(3x) + 2 &= 14 - (3x)^2 \\ 3x^2 + 2 &= 14 - 9x^2 \\ 12x^2 &= 12 \\ x^2 &= 1 \\ \sqrt{x^2} &= \sqrt{1} \\ |x| &= 1. \end{aligned}$$

□

There are several comments to make about this proof. First, the result of substituting $y = 3x$ in $14 - y^2$ is $14 - (3x)^2$, *not* $14 - 3x^2$. Even though there are no parentheses written in $y = 3x$, we have to *put parentheses* around the $3x$ when we substitute it for y to make sure that any operations applied to it act on the whole thing. That is, the expression $14 - y^2$ means to square y and then subtract the result from 14; so when $y = 3x$, that means we must square $3x$, and we have to write $(3x)^2$ to mean that we are squaring the whole expression $3x$. If in doubt, always put parentheses around an expression when substituting it for something.

Secondly, we have combined two "moving things to the other side" steps in going from $3x^2+2 = 14-9x^2$ to $12x^2 = 12$. Thirdly, we are allowed to apply the square-root function because both expressions x^2 and 1 are in its domain (which consists of numbers that are greater than or equal to zero). And finally, note that $\sqrt{x^2}$ simplifies to $|x|$, not to x: for example, $\sqrt{(-3)^2} = \sqrt{9} = 3 = |-3|$. (By contrast, $(\sqrt{x})^2$ does simplify to x, as long as $\sqrt{x}$ makes sense in the first place, that is as long as $x \geq 0$.)

Our final example involves a disequality hypothesis ($\neq$), which we use to show that we can divide by something.

Theorem 1.3. *Suppose $x^2 - y^2 = 2x - 2y$ and $x \neq y$. Then $x + y = 2$.*

Proof of 1.3.

$$\begin{aligned} x^2 - y^2 &= 2x - 2y \\ (x + y)(x - y) &= 2(x - y) \\ x + y &= 2 \end{aligned}$$

where in the last step we can divide by $x - y$, since $x \neq y$ implies $x - y \neq 0$. □

Always make sure when you divide by something that it's something you're allowed to divide by — which in ordinary arithmetic means that it's nonzero — and remark on that in the proof.

1.2 About examples

When a theorem statement involves variables, and doesn't specify otherwise, the intended meaning is that the conclusion is true *no matter what* the values of

those variables are, as long as the hypotheses are satisfied. Thus, such a theorem is actually claiming all at once that *infinitely many* statements are true, each of which is called an *instance* or *specialization* or *example* of the theorem.

For example, one instance of Theorem 1.1 is when $x = 5$, $y = 6$, and $z = 4$. These choices are an instance of the theorem because when substituted, they make the hypotheses true: $5 + 1 = 6$ and $5 - 1 = 4$. Therefore, in this instance the theorem tells us that $5^2 = 4 \cdot 6 + 1$. We can verify this directly, since both equal 25. The point of the theorem is that this will *always* be true, and we can't verify all those infinitely many cases directly.

By the same token, if I claim that some statement is true, but you can find *a single instance* in which that statement is not true, then my claim was false. For example, consider the following:

Non-Theorem 1.4. *Suppose $x^2 + xy = 2y^2$. Then $x = y$.*

One instance of this claim is when $x = 7$ and $y = 7$; this satisfies the hypothesis since $7^2 + 7 \cdot 7 = 2 \cdot 7^2$ (both equaling 98), and also the conclusion since $7 = 7$. However, another instance of this claim is $x = 4$ and $y = -2$; this satisfies the hypothesis since $4^2 + 4 \cdot (-2) = 2(-2)^2$ (both equaling 8), but *not* the conclusion since $4 \neq -2$. Therefore, Non-Theorem 1.4 is *not true*: even though some instances are true, at least one false instance is enough to break it. A false instance of a statement is known as a *counterexample*. Specifically, a counterexample to a statement consists of specific values for the variables in it such that all the hypotheses are true, but the conclusion is false.

Since Non-Theorem 1.4 is not true, it *cannot have a correct proof*. The whole point of a proof is that it guarantees, beyond a shadow of a doubt, that the statement being proven is true. Therefore, if a statement is *not* true, it is impossible for it to have a correct proof.

In particular, *an example, or a collection of examples, is not a proof*. For instance, we could *not* prove Non-Theorem 1.4 by saying "let $x = 7$ and $y = 7$; then $7^2 + 7 \cdot 7 = 2 \cdot 7^2$ and $7 = 7$, so the theorem is true". This would be verifying a single instance, but that is not enough to prove the statement — as it had better not be, since we have seen that Non-Theorem 1.4 is false, and a false statement cannot have a correct proof.

In a class or a textbook, you are frequently presented with a theorem and told to prove it. In this case, you can be fairly certain the theorem *is* true, and that your job is only to *find* a correct proof of it. However, outside of a classroom setting, there is no god of mathematics who descends from the clouds and tells us what the true theorems are that we have to prove. Instead, we have to *guess* what statements *might* be true, and then try to verify our guess by proving them. (*How* to make such guesses is a whole nother deep question, beyond the scope of these notes — but not, I hope, beyond the scope of your instructor and supplementary textbook.)

A statement that we think might be true and we are trying to prove is called a *conjecture*. However, when trying to prove a conjecture, you should always be open to the possibility that it is *not*, in fact, true, in which case what you'd like to find instead is a counterexample. Often it is useful to alternate between trying

to prove a conjecture and trying to find a counterexample to it: sometimes your failure to find a proof will point you in the direction of a counterexample, while other times your failure to find a counterexample will point you in the direction of a proof.

The distinction between proofs, examples, and counterexamples is extremely important. We will return to it in section 1.5, and place it in a more general context in chapter 3.

1.3 On working forwards

You may have noticed that in stating the rules of algebra I spoke about "true equations" rather than simply "equations". Grammatically, an equation represents a *sentence*,[3] where "=" represents the verb "equals".[4] For instance, $x+1=y-2$ represents the sentence "x plus one equals y minus two", in which "x plus one" is the subject and "y minus two" is the object. Being a sentence, an equation can be either true or false. (We will talk more about sentences that can be either true or false in chapter 2.) For instance, $2+2=4$ is true, while $2+2=5$ is false.

It is *absolutely essential* that when writing a proof, *everything we say is known to be true at that point.* At the beginning of the proof, the only statements known to be true are the hypotheses. We then obtain more true statements as we go through the proof, deducing them from the previously known statements, until we eventually show that the conclusion is true. This is the only way we can be sure that the theorem is true, i.e.[5] that the conclusion is necessarily true *whenever* the hypotheses are true.

I emphasize this because it often happens when "doing algebra" that we do things to an equation that is *not yet known to be true.* For example, suppose you are asked to *solve* the equation $x^2=2x-1$. This means to *find the values of x that make it true.* You might reason as follows:

$$\begin{aligned} x^2 &= 2x-1 \\ x^2-2x+1 &= 0 \\ (x-1)^2 &= 0 \\ x-1 &= 0 \\ x &= 1. \end{aligned}$$

[3] Or, more generally, a *clause*: a part of a sentence that could be a sentence on its own, with a subject and a verb.

[4] If it doesn't contain the symbol "=", then it is *not an equation*! If it contains the symbols $<$, $>$, $\leq$, or $\geq$, it is an *inequality*, and if it contains $\neq$ it is a *disequality* (although some people include disequalities as inequalities). Something containing none of them, like $4x^2+7$, can be called an *expression* or a *formula*.

[5] Remember that "i.e." stands for the Latin *id est*, which means "that is"; in contrast to "e.g.", which stands for the Latin *exempli gratia*, which means "for example". Don't confuse these two abbreviations.

This calculation bears a superficial resemblance to our algebraic proofs, but it is *not a proof*, because we *don't yet know that* $x^2 = 2x - 1$. Instead, we are doing "scratch work" to figure out what x must be in order to *make* that equation true.

After we find x in this way, we can then work forwards to show that this value of x does work: $1^2 = 1$ and $2(1) - 1 = 1$, so $1^2 = 2(1) - 1$. In elementary algebra this is sometimes called *checking your work*, but from our perspective *this is the proof.* What came before it was just preliminary.

I emphasize this because over and over again, when writing algebraic proofs, I see students try to start from the *goal* and "do algebra to it", ending with a tautology like "$x = x$", and think they have proven the theorem. *This is not a proof!* For example, consider the following:

Non-Theorem 1.5. *Suppose* $x^2 - x = xy$. *Then* $x = y + 1$.

We might try to prove this as follows:

$$\begin{aligned} x &= y + 1 \\ x^2 &= xy + x \\ x^2 - x &= xy \\ xy &= xy. \end{aligned}$$

The individual steps of algebra here are fine: first we multiply both sides by x, then we move x to the other side, and then we substitute $x^2 - x$ by xy. However, the overall result is *not a proof* because it *starts from the goal* $x = y + 1$, which we *do not yet know to be true.*

In fact, Non-Theorem 1.5 is *false*: for instance, if $x = 0$ and $y = 3$, the hypothesis $x^2 - x = xy$ is true, but the conclusion $x = y + 1$ is not. As discussed in section 1.2, this counterexample means that *it cannot have a correct proof.* In particular, the above non-proof is wrong; and the reason it is wrong is that it starts from the goal.

On the other hand, however, just because a statement is true does not mean that every attempted proof of it is correct! For example, suppose we tried to prove Theorem 1.1 like this:

$$\begin{aligned} x^2 &= yz + 1 \\ x^2 - 1 &= yz \\ (x - 1)(x + 1) &= yz \\ y(x + 1) &= yz && \text{(substituting } x - 1 = y\text{)} \\ yz &= yz && \text{(substituting } x + 1 = z\text{).} \end{aligned}$$

Even though Theorem 1.1 is true, this would still *not be a correct proof* of it, because it starts from the goal.

1.4 Chaining equalities

Consider the following.

Theorem 1.6. $(x-1)^2 + (x+1)^2 = 2(x^2+1)$.

At first sight, it may not be clear how we can possibly prove this. The only equation in sight is the *goal*, and I've just shouted from the mountaintops that you can't start from the goal: so where *can* we start? But fortunately, I included in the list of algebraic rules that "any expression is equal to itself", so that means we always have some place to start: we can start with, say $(x-1)^2 + (x+1)^2 = (x-1)^2 + (x+1)^2$. Now we can just successively simplify the right-hand side.

Proof of 1.6.

$$\begin{aligned}
(x-1)^2 + (x+1)^2 &= (x-1)^2 + (x+1)^2 \\
(x-1)^2 + (x+1)^2 &= (x^2 - 2x + 1) + (x^2 + 2x + 1) \\
(x-1)^2 + (x+1)^2 &= 2x^2 + 2 \\
(x-1)^2 + (x+1)^2 &= 2(x^2+1).
\end{aligned}$$

□

This pattern is so common that we usually omit the first equation and don't usually bother to re-write the left-hand side every time:

Better Proof of 1.6.

$$\begin{aligned}
(x-1)^2 + (x+1)^2 &= (x^2 - 2x + 1) + (x^2 + 2x + 1) \\
&= 2x^2 + 2 \\
&= 2(x^2+1).
\end{aligned}$$

□

This can also be thought of as putting together a sequence of equalities that "match in the middle": we have $(x-1)^2+(x+1)^2 = (x^2-2x+1)+(x^2+2x+1)$, and $(x^2 - 2x + 1) + (x^2 + 2x + 1) = 2x^2 + 2$, and $2x^2 + 2 = 2(x^2 + 1)$, so in conclusion $(x-1)^2 + (x+1)^2 = 2(x^2+1)$. For this reason, we call this method of proof *chaining* equalities. We can even put it on one line, if it fits:

$$(x-1)^2 + (x+1)^2 = (x^2 - 2x + 1) + (x^2 + 2x + 1) = 2x^2 + 2 = 2(x^2+1).$$

A variation on chaining equalities is to successively simplify both sides of the goal equation and show that we end up with the same thing.

Theorem 1.7. $(x-3)^2 + 6x = (x+3)^2 - 6x$.

Proof of 1.7. We simplify both sides separately:

$$\begin{aligned}
(x-3)^2 + 6x &= x^2 - 6x + 9 + 6x \\
&= x^2 + 9 \\
(x+3)^2 - 6x &= x^2 + 6x + 9 - 6x \\
&= x^2 + 9.
\end{aligned}$$

□

This is perhaps an appropriate place to mention that when simplifying or manipulating equations, some students have a tendency to write that the first equation is *equal* to the second, for instance

$$¿¿¿ \qquad x+1=y \;=\; x=y-1 \qquad ???^{6}$$

This is confusing for a number of reasons, one of which is that it looks very much like a "chaining equalities" claim that $x+1=y$ and $y=x$ and $x=y-1$ and therefore $x+1=y-1$, which is *not* at all what is meant. Like many of the world's problems, this could be somewhat mitigated by parentheses:

$$(x+1=y) \;=\; (x=y-1).$$

This is technically correct in that the two equations represent equal "truth values" (a concept we will discuss more in chapter 2), but it still looks confusing because the $=$ symbol is being used in two different ways in the same statement. It's better to use the symbol $\Leftrightarrow$; we will discuss this more in section 2.9, but for the moment I'll just say that it means two sentences are equivalent (have the same truth value):

$$(x+1=y) \;\Leftrightarrow\; (x=y-1).$$

However, even this has dangers, because not infrequently in an algebraic proof we operate on an equation to produce another equation that is not *equivalent* to it, but only *follows from it*. For example, starting from $x=y$ we can deduce $x^2=y^2$, but the two are not equivalent: $x^2=y^2$ is true sometimes even if $x=y$ is not, for instance if $x=2$ and $y=-2$. But if you're in the habit of connecting your algebraic steps with $=$ or even $\Leftrightarrow$, you might accidentally write

$$¿¿¿ \qquad (x=y) \;\Leftrightarrow\; (x^2=y^2) \qquad ???$$

which is not true. One way to try to fix this is by replacing the $\Leftrightarrow$ with the symbol $\Rightarrow$, which means that the statement on the right follows from the one on the left (we will discuss it more in section 2.7):

$$(x=y) \;\Rightarrow\; (x^2=y^2).$$

However, this is not really quite correct either: it asserts that *if* $x=y$ were true (hypothetically) then $x^2=y^2$ *would* be true, whereas what you want to say is that *because* we *already know* that $x=y$ is true we can *deduce* that $x^2=y^2$ is *also* true. You can describe that in words, as I did in the paragraph right after Theorem 1.1, but the standard convention is to do as I've done in all the example proofs and write each equation *below* the previous one, with the equals signs lined up, and *no* connecting symbols at all:

$$\begin{aligned} x &= y \\ x^2 &= y^2. \end{aligned}$$

This also extends more cleanly to sequences of many equations, and is easier to parse (with practice) when reading.

[6] When I am forced by the exigencies of narrative to display a statement or equation that is not just discouraged but *incorrect*, I will put these bracketing question-marks around it, in an attempt to ensure that no careless reader latches onto it as truth.

1.5 Number systems

The objects we work with when we do algebra are called *numbers*, but there is not just one notion of "number". In your education until now, you have probably encountered the following notions of "number", although you may not be familiar with the names and symbols that mathematicians use.

- The *natural numbers*, also called the *counting numbers*, are the non-negative whole numbers:
$$0, 1, 2, 3, 4, \ldots.$$
We write $\mathbb{N}$ for the collection of all natural numbers. (Regarding the inclusion of 0, see section 1.6.)

- The *integers* are the natural numbers together with their negatives:
$$\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots.$$
We write $\mathbb{Z}$ for the collection of all integers; this stands for the German word *zählen* ("number").

- The *rational numbers* are the fractions (proper or improper, positive or negative) that can be written as one integer divided by another. Thus, every integer is a rational number, as are $\frac{1}{3}$, $\frac{27}{5}$, $\frac{-2}{7}$, and so on. We write $\mathbb{Q}$ for the collection of all rational numbers; this stands for the word *quotient*.

- The *real numbers* are all the numbers that can be written as a finite or infinite decimal expansion. Every rational number is a real number; in fact they are precisely the real numbers whose decimal expansion is either finite or repeating, e.g. $\frac{1}{2} = 0.5$ and $\frac{1}{3} = 0.33333\cdots$ and $\frac{1}{7} = 0.142857142857\cdots$. Some familiar real numbers that are not rational include $\sqrt{2} = 1.414213\cdots$ and $\pi = 3.1415926\cdots$ (in some cases we will be able to prove their non-rationality in chapter 4). We write $\mathbb{R}$ for the collection of all real numbers.

Note that the names of number systems ($\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$) are usually written in a funny mathematical font called "blackboard bold" or "double-struck", where one or more of the lines are doubled.

A given number can have more than one *representation*, but these different representations always still denote the same *number*, and we write that they are *equal*.[7] Probably the most familiar case of this is unreduced fractions; for instance, $\frac{1}{2}$ and $\frac{2}{4}$ are two representations of the same rational number, and so

[7] The mathematician John Baez has a saying that "every interesting equation is a lie". For instance $2 + 2 = 4$ is a "lie" since "$2 + 2$" and "4" are evidently *different expressions*, whereas $1 = 1$ is evidently true but uninteresting. The point is that an interesting equation expresses the fact that the two sides, while not identical expressions, are *representations* of the same underlying number. A philosopher might say that they are *intensionally distinct but extensionally equal*, or have *different senses but the same reference*, analogously to how "the morning star" and "the evening star" are two different names for the planet Venus.

we write $\frac{1}{2} = \frac{2}{4}$. Every rational number has infinitely many representations as a fraction, although only one of them is in "lowest terms".

Other kinds of numbers also have multiple representations. Some real numbers have more than one decimal expansion; for instance, $1 = 0.99999\cdots$[8] and $0.5 = 0.499999\cdots$. This happens precisely when one of the decimal expansions is finite (or, equivalently, ends with an infinite string of zeros), in which case the other one ends with an infinite string of nines.

Even natural numbers have more than one representation. We usually write numbers in a *base ten* place value representation, so for instance "238" means two times ten squared, plus three times ten (that is, three times ten to the one), plus eight (that is, eight times ten to the zeroth). However, the same number can be written in other bases as well. In base n notation, the right-most digit is still the units, while the next digit to the left represents the n's, the next one the n^2's, and so on. Each digit is between 0 and $n-1$, so for instance in base two (*binary*) the only digits allowed are 0 and 1, and for instance the base two number 1011 represents eight plus two plus one, or eleven.[9] If $n > 10$, we use letters as additional "digits" after the ordinary ones $0, \ldots, 9$, so for instance B3 in base sixteen (*hexadecimal*) means eleven times sixteen, plus three, or 179 in base ten.[10]

We often add a subscript to denote the base, so for instance, 314_8 is in base eight and means three times eight squared, plus one times eight, plus four, which is the number that we would write as 204 in base ten, or 204_{10} for emphasis.[11] To emphasize it aganin, these are just two different ways to represent the *same* number (that is, $314_8 = 204_{10}$), and any natural number can be written in any base. So can integers and rational numbers in a straightforward way, and also real numbers: in base n, the digit to the right of the point is the $\frac{1}{n}$'s, the next is the $\frac{1}{n^2}$'s, and so on. (When $n \neq 10$ the point is not called a *decimal* point, since "decimal" refers to ten; sometimes it is called a *radix point*.) Whether a number has a finite or infinite radix expansion can depend on the base, for instance $0.33333\cdots_{10} = 0.1_3$, but the collection of real numbers does not depend on the

[8] If you have never encountered this fact before, one simple way to justify it to yourself is to start from $\frac{1}{3} = 0.33333\cdots$ and multiply both sides by 3. It's also the sum of an infinite geometric series; if you know what that means, you can probably work out how it applies.

[9] This means that technically there is no such thing as "base one", since the only digit allowed would be 0, and so 0 would be the only number that could be represented in base one. However, it is common to abuse language a bit and speak of representing a number in *unary* when we write it with "tally marks", for instance writing 5 as |||||, since at least the place values are right: five is one to the fourth, plus one to the third, plus one squared, plus one, plus one to the zeroth.

[10] If you are sufficiently masochistic to muck around with $n > 36$, you can make up any new "digits" for it that you want.

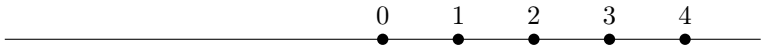
[11] The subscript itself is always written in base ten. For instance, in 314_8, the 8 is written in base ten; and in $\mathrm{B3}_{16}$, the 16 is written in base ten. Otherwise no one would ever have any clue what we were talking about. If the ten-centric-ness of this bothers you, you can use number words in the subscript, such as 318_{eight} or 204_{ten} or $\mathrm{B3}_{\text{sixteen}}$. Number words are independent of the base: the word "ten" always means the same *number*, whether written as 10_{10} or 12_8 or A_{16}; whereas 10_8 is the number eight, 10_2 is the number two, and 10_{16} is the number sixteen, and none of them are ever pronounced "ten". (Yes, the system of number words in English is also ten-centric. Deal with it.)

base: any real number can be written in any base.

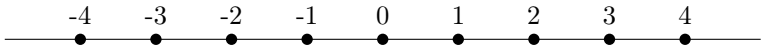
As you probably know, digital computers represent natural numbers internally in binary, while programmers often use hexadecimal due to its close connection with binary. Namely, since $16 = 2^4$, when rewriting a binary number in hexadecimal, every group of four binary digits becomes exactly one hexadecimal digit. So for instance $1011\,0011_2 = \mathrm{B3}_{16}$, since $1011_2 = \mathrm{B}_{16}$ and $0011_2 = 3_{16}$. Computers also work with things called *floating point* numbers, which are approximations of real numbers obtained by writing them in binary and then remembering only some number of the leftmost nonzero digits, along with how far from the radix point (and in which direction) those digits were. This sounds sensible at first (it's the base-two version of "scientific notation"), but the behavior of floating point numbers turns out to be *exceedingly weird*, so be very careful if using them for anything serious.

If we treat each collection[12] of numbers as a world unto itself, not all the familiar algebraic operations may exist in that world. For example, the world of natural numbers $\mathbb{N}$ has addition and multiplication, but not subtraction, because subtracting two natural numbers may not yield another natural number. Similarly, the world of integers $\mathbb{Z}$ has addition, subtraction, and multiplication, but not division, because dividing two integers may not yield another integer. The world of rational numbers $\mathbb{Q}$ has addition, subtraction, multiplication, and division (by nonzero numbers), but not operations such as square roots. And the world of real numbers $\mathbb{R}$ has addition, subtraction, multiplication, division by nonzero numbers, and square roots of non-negative numbers.[13]

These number systems $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ can all be drawn on a traditional "number line", and can be thought of intuitively as filling in more and more "gaps" in that line. We start with the natural numbers, $\mathbb{N}$:



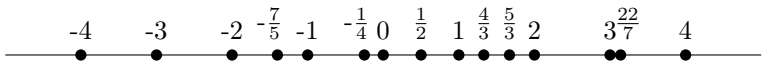
Then we fill in the "gap" stretching off to the left by adding the negative numbers to get the integers, $\mathbb{Z}$:



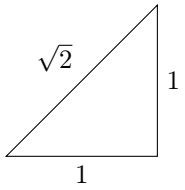
[12] Throughout these notes I use the word "collection" where most mathematicians would use the word "set". Partly this is because it's outside the scope of these notes to talk about what a "set" is (although your supplementary textbook will probably discuss it), whereas "collection" is nicely vague. And partly it's because some of the collections we will consider, such as the surreal numbers and ordinal numbers, are *not* actually "sets" in the usual sense, being "too large". However, for the most part, whenever I say "collection" you can read "set" if you prefer. Programmers are free to say "type" instead of either word.

[13] Mathematicians have special words for different kinds of number systems depending on what operations exist in that world. For instance, a number system with addition, subtraction, multiplication, and division is called a *field*, and one with just addition, subtraction, and multiplication is called a *ring*. Thus $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ are rings, while $\mathbb{Q}$ and $\mathbb{R}$ are fields. If you're a programmer at heart, then you can think of each of these words as defining an *interface*, and each number system as an *implementation* of some interface. You can learn more about these ideas in a subject called *abstract algebra*.

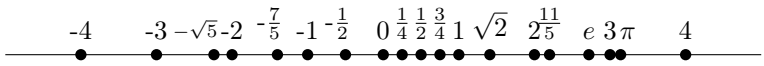
And then we start filling in the gaps in between each integer with rational numbers, $\mathbb{Q}$:



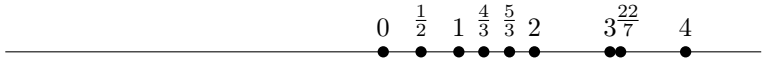
At this point it may not be obvious, geometrically, that there are any gaps left. Indeed, at first the ancient Greeks believed there weren't any! But then they discovered, using the Pythagorean theorem, that (for example) the length of the hypoteneuse of an isoceles right triangle is not any rational multiple of the lengths of its sides:



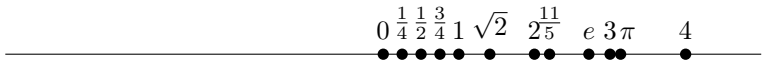
(We'll come back to this in section 4.3.) This corresponds to a gap in the rational numbers: on the left there are those whose square is < 2, and on the right are those whose square is > 2. If we add in numbers labeling all the gaps of this sort, we get the real numbers, $\mathbb{R}$:



It's worth observing that we didn't have to start filling in gaps in this order. We could have started with $\mathbb{N}$ and filled in the gaps *between* natural numbers first, obtaining the *non-negative rational numbers*, which I will denote $\mathbb{Q}_{\geq 0}$:



and then the *non-negative real numbers*, which I will denote $\mathbb{R}_{\geq 0}$:



before adding any negative numbers. In $\mathbb{Q}_{\geq 0}$ and $\mathbb{R}_{\geq 0}$, we can add and multiply and divide (by anything nonzero), but not subtract (in general).

There are also many other number systems beyond the familiar ones listed above. Here are a few:

- The *complex numbers* are the numbers that can be written as $x + iy$, where x and y are real numbers and i is a new "imaginary" number with the property that $i^2 = -1$. We write $\mathbb{C}$ for the collection of all complex numbers, which includes all the real numbers since $x = x + i \cdot 0$. Like real numbers, complex numbers can be added and subtracted and multiplied

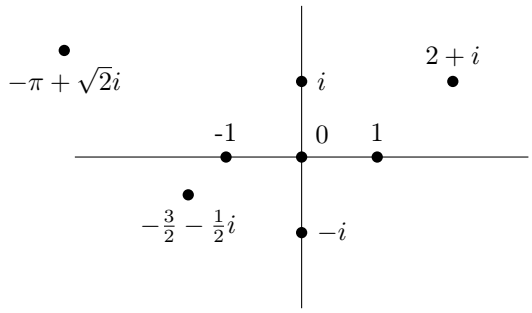


Figure 1.1: The complex plane

and divided (except by zero). However, unlike our previous examples, the complex numbers are *not ordered*, so inequalities such as $<$, $>$, $\leq$, and $\geq$ do not make sense: for instance, we cannot ask whether $i > 0$ or $i < 0$. While the real numbers can be thought of as points on a line, as discussed above, the complex numbers can be thought of as points in a two-dimensional *plane*, with the number $x + iy$ corresponding to the point with coordinates (x, y), as shown in Figure 1.1.

- The *surreal numbers* expand the real numbers in a different way, retaining an order but adding new "infinite" numbers that are larger than all real numbers. For example, there is a surreal number called ω (the lowercase Greek letter "omega") such that $\omega > x$ for all real numbers x. Although ω is "infinite" in a certain sense, it is not "infinity", because we can treat it as a number in ordinary ways. For instance, we can add one to it to get another surreal number $\omega + 1$ such that $\omega < \omega + 1$, and subtract one from it to get another surreal number $\omega - 1$ such that $\omega - 1 < \omega$ (but $\omega - 1$ is still greater than all real numbers). Or we can negate it to get $-\omega$, which is less than all real numbers. Or double it to get 2ω, which is greater than $\omega + x$ for any real number x.

  We can also take its reciprocal $\frac{1}{\omega}$, which is a "positive infinitesimal", meaning that it is positive but smaller than every positive real number. Similarly, $-\frac{1}{\omega}$ is a negative infinitesimal, while $2 + \frac{1}{\omega^2}$ is "infinitely close" to 2, and so on. There are also many more infinite surreal numbers, many of them *much* larger than ω and not even "definable" in terms of ω, although I will not try to define them all precisely here.

  I will write $\mathbb{S}$ for the collection of all surreal numbers, which includes all the real numbers as well as the new ones like ω, $\omega + 1$, and $\frac{1}{\omega}$. Surreal numbers can be added and subtracted and multiplied and divided (except by zero), and have square roots of non-negative numbers (for instance, $\sqrt{\omega}$ is smaller than ω but still greater than all real numbers). Geometrically, we can think of the surreal numbers as filling in additional "gaps" in the real number line, such as the "gap" that comes *after* all the real numbers (which we

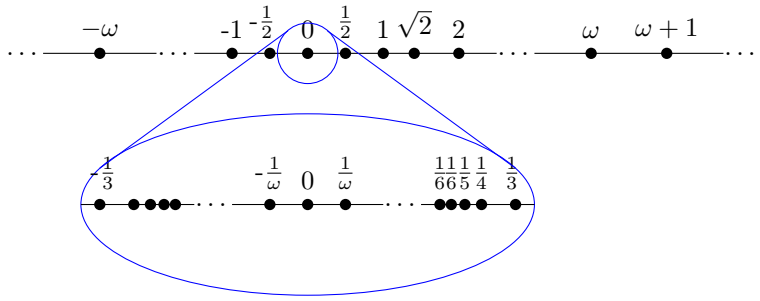


Figure 1.2: The surreal line

can only see by "zooming out" with an "infinitely powerful telescope"), or the gap in between 0 and all the positive real numbers (which we can only see by "zooming in" with an "infinitely powerful microscope"), as shown in Figure 1.2. In fact, there's a precise sense in which the surreal numbers are, by definition, the result of filling in *all possible* gaps.

- For an integer n, the *integers modulo* n are like the integers but "wrap around" after n steps, so that for instance $n+1$ is regarded as the same as 1. This is also called "clock arithmetic" with n hours on the clock. We write $[k]_n$ for the integer k considered modulo n, so that for instance we can say $[13]_{12} = [1]_{12}$. The general rule is that we can add or subtract any number of copies of n without changing a number modulo n, for instance

$$[3]_5 = [8]_5 = [13]_5 = [18]_5 = [23]_5 = [-2]_5 = [-7]_5 = [-12]_5 = \cdots .$$

We write $\mathbb{Z}_{/n}$ for the collection of integers modulo n. Unlike all our other number systems, there are only *finitely many* integers modulo n (although each has many different *representations*, as with $[3]_5$ above); in fact there are exactly n of them. For instance, the integers modulo 5 are $[0]_5$, $[1]_5$, $[2]_5$, $[3]_5$, and $[4]_5$.

Integers modulo n can be added and subtracted and multiplied just like ordinary integers but with the brackets "coming along for the ride", for instance

$$[2]_5 + [3]_5 = [2+3]_5 = [5]_5$$

which, of course, also equals $[0]_5$. You can think of this as working with integers expressed in a base n representation, as above, but remembering only the units digit. However, it is *not* in general possible to *divide* two integers modulo n, and they are not ordered. Geometrically, we can think of the numbers modulo n as existing on a circle, such as an analog clock when $n = 12$, as shown in Figure 1.3. We will discuss $\mathbb{Z}_{/n}$ more in section 3.7.

It is also possible to "fill in the gaps" on a circle, or equivalently allow rational or real numbers inside the "modulo brackets". For instance $[\frac{5}{2}]_{12}$

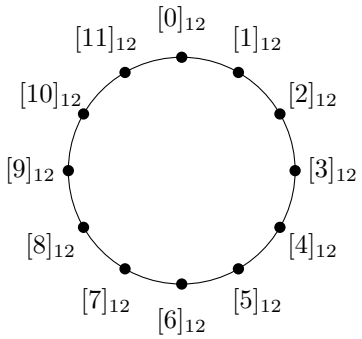


Figure 1.3: Numbers modulo 12

is halfway between $[2]_{12}$ and $[3]_{12}$; on a clock it would be the time 2:30. Thus we get number systems $\mathbb{Q}_{/n}$ and $\mathbb{R}_{/n}$, in which we can still add and subtract in them, but we cannot consistently multiply. For instance, $[\frac{1}{3}]_3 \cdot [3]_3 = [\frac{1}{3}]_3 \cdot [0]_3 = [0]_3$, whereas $[\frac{1}{3} \cdot 3]_3 = [1]_3$.

- The decimal expansions that represent real numbers can be infinite on the right side of the decimal point, but are finite on the left side of the decimal point. By contrast, the *10-adic numbers* are defined by "decimal expansions" that can be infinite on the *left* side of the decimal point, but must be finite on the right. (You might try to allow expansions that are infinite on both sides, but it turns out to be impossible to multiply these.) They can be added, subtracted, and multiplied with the usual "carrying and borrowing" rules of arithmetic, continuing forever off to the left. This has the nice consequence that negative numbers can be represented without a minus sign: for instance,

$$\begin{array}{rccccccc} & \cdots & 9^1 & 9^1 & 9^1 & 9^1 & 9 \\ + & \cdots & 0 & 0 & 0 & 0 & 1 \\ \hline & \cdots & 0 & 0 & 0 & 0 & 0 \end{array}$$

  so the 10-adic number $\cdots 99999$ is equal to -1, since when we add it to 1 we get 0. But the 10-adic numbers are not ordered, and only some of them can be divided. We write $\mathbb{Q}_{10}$ for the collection of 10-adic numbers.

  More generally, we can consider *n-adic numbers* $\mathbb{Q}_n$ that are written in base n instead of base 10, and can be infinite to the left of the radix point but finite to the right. Unlike the infinite-to-the-right case of real numbers, we do actually get *different* number systems here by choosing different values of n. For instance, it turns out that if n is prime, then we *can* divide by anything nonzero in $\mathbb{Q}_n$.

  By the *n-adic integers* we mean those n-adic numbers with nothing to the right of the radix point. As mentioned above, these include both negative and positive ordinary integers, without needing a minus sign.

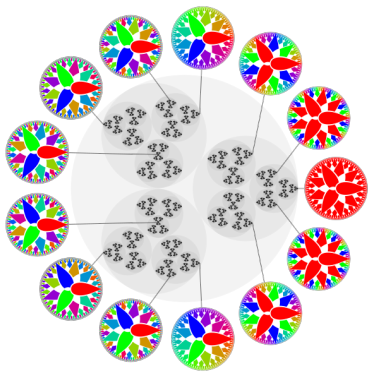


Image by Melchoir – Own work, CC BY-SA 3.0, Link

Figure 1.4: The 3-adic integers

This lies behind the *two's complement* representation of signed integers in binary on a computer, which essentially means using 2-adic integers but truncating them some fixed distance to the left of the radix point.

It is possible to draw (beautiful!) geometric pictures of the n-adic numbers, such as shown in Figure 1.4; but explaining exactly what a picture like this means is beyond the scope of these notes.

There is a standard notation to say that a variable belongs to a given number system. Instead of "x is a real number" we write $x \in \mathbb{R}$, and similarly for the other number systems; the symbol $\in$ is a stylized version of the Greek letter epsilon and means "is an element of". (Don't confuse it with the *actual* Greek letter epsilon, which looks like ϵ or ε; that is used in mathematics as a letter, not a symbol.) We also write "x and y are real numbers" as $x, y \in \mathbb{R}$, and so on.

You must learn to both read and write the symbol $\in$ without confusing it with ε (for example, in the subject of *real analysis* one commonly uses ε a variable, with $\varepsilon \in \mathbb{R}$), and you must learn the meanings of the symbols $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$; these are utterly standard and ubiquitous in mathematics. The other numbers we have mentioned are less common: although $\mathbb{Z}_{/n}$ is still quite common, $\mathbb{Q}_n$ is less so, and $\mathbb{S}$ is rather esoteric. The notations for these other number systems are also less standardized: $\mathbb{Q}_n$ is pretty standard, but the surreals are more often denoted **No**; and $\mathbb{Z}_{/n}$ has more names than Odin, being variously called $\mathbb{Z}_n$[14] or $\mathbb{Z}/n\mathbb{Z}$ or C_n or Z_n or (when n is prime) $\mathbb{F}_n$.

Because different number systems have different properties, it is important to specify what sort of numbers we are working with when we state and prove a theorem. For instance, here is a more precise statement of Theorem 1.1:

[14] Unfortunately, the notation "$\mathbb{Z}_n$" is ambiguous. Some mathematicians use it to mean what I am calling $\mathbb{Z}_{/n}$, the integers mod n. But other mathematicians use it to mean the n-adic integers (those n-adic numbers with nothing to the right of the radix point). In these notes I will avoid the notation $\mathbb{Z}_n$.

Theorem 1.1 (better)**.** *Suppose $x, y, z \in \mathbb{R}$, and that $x + 1 = y$ and $x - 1 = z$. Then $x^2 = yz + 1$.*

Recall that in section 1.2 I said that a theorem statement involving variables makes a claim that its conclusion is true *no matter what* those variables are, as long as the hypotheses are true. Now we can be a bit more careful about this: it only makes this claim as long as the value of the variables belong to the correct number systems. Thus, Theorem 1.1 only makes a claim about *real number* values for x, y, and z.

In this case, we could instead have said that $x, y, z \in \mathbb{Z}$, and that would have given us a *different theorem* which is *also true* (with an identical-looking proof). Similarly, we could have said that they are rational numbers, or complex numbers, etc. However, the truth of other theorems may depend on what kind of numbers we are talking about. Here are some examples.

Theorem 1.8. *Suppose $x, y \in \mathbb{R}$ and $x^2 + y^2 = 0$. Then $x = 0$ and $y = 0$.*

This is true, but it would be false if we supposed instead that $x, y \in \mathbb{C}$, since then a counterexample would be $x = 1$ and $y = i$.

Theorem 1.9. *Suppose $x \in \mathbb{R}$. Then there is an $n \in \mathbb{N}$ such that $n > x$.*

This is true, but it would be false if we supposed instead that $x \in \mathbb{S}$, since then a counterexample would be $x = \omega$, which is greater than all natural numbers.

Theorem 1.10. *Suppose $x, y \in \mathbb{Z}$ and $xy = 0$. Then either $x = 0$ or $y = 0$.*

This is true, but it would be false if we supposed instead that $x, y \in \mathbb{Z}_{/6}$. In that case, a counterexample would be $x = [2]_6$ and $y = [3]_6$, which are nonzero, but $xy = [6]_6 = [0]_6$.

Theorem 1.11. *Suppose $x \in \mathbb{R}$ and $x^2 = x$. Then either $x = 0$ or $x = 1$.*

This is true, but it would be false if we supposed instead that $x \in \mathbb{Q}_{10}$. If we start with 5 and repeatedly square it:

$$\begin{aligned}
&&&5\\
5^2 &= &&25\\
25^2 &= &&625\\
625^2 &= &&390625\\
390625^2 &= &&152587890625\\
152587890625^2 &= &&\cdots 386962890625\\
\cdots 386962890625^2 &= &&\cdots 855712890625\\
\cdots 855712890625^2 &= &&\cdots 793212890625
\end{aligned}$$

then the rightmost sequence of digits, colored blue above, "stabilize" or "converge". Thus, there is a nonzero 10-adic number $x = \cdots 12890625$ such that $x^2 = x$. (Note that this also yields a counterexample to Theorem 1.10 in $\mathbb{Q}_{10}$, since $x^2 = x$ implies $x(x - 1) = 0$. In the other direction, $\mathbb{Z}_{/6}$ also contains a counterexample to Theorem 1.11, since $([4]_6)^2 = [4^2]_6 = [16]_6 = [10]_6 = [4]_6$.)

If a statement refers to elements or operations that *don't even exist* on some number system, then it is not just false but *meaningless* in that number system. For instance:

Non-Theorem 1.12. *Suppose* $x, y, z \in \mathbb{Z}_{/10}$ *and* $xy = z$. *Then* $x = \dfrac{z}{y}$.

This would be true in any number system where division exists, but in $\mathbb{Z}_{/10}$ where division doesn't exist, it doesn't even make sense: the symbol $\frac{z}{y}$ *doesn't mean anything* and thus the statement is *neither true nor false*, but meaningless.

Technically speaking, we should similarly complain about the analogous:

Theorem 1.13. *Suppose* $x, y, z \in \mathbb{Z}$ *and* $xy = z$. *Then* $x = \dfrac{z}{y}$.

However, since every integer is also a rational number (or a real number, or even a complex number or surreal number), and division makes sense in those number systems, we conventionally give a meaning to statements like this by assuming we are in one of those larger worlds. The problem with Non-Theorem 1.12 is that there is no "larger world" where division exists and in which we can regard the integers modulo 10 as sitting.

Moving to a larger world doesn't always make things better, though: sometimes it can make things worse. For example, this is true:

Theorem 1.14. *Suppose* x *and* y *are positive real numbers. Then we have* $\sqrt{xy} = \sqrt{x}\sqrt{y}$.

But if we allow x and y to be not-necessarily positive real numbers, the statement becomes *not true or false but meaningless*, since the $\sqrt{}$ operation is only defined on non-negative real numbers (or, more generally, non-negative surreal numbers).

You might think we could fix this particular example by working with complex numbers, but no dice. Every complex number does have *a* square root... but in fact it has *two* of them (unless it's zero), and there's no consistent way to single out one of them to call "the" square root. If x is a positive real number, we can define $\sqrt{x}$ to be *the positive* real number whose square is x, but since the complex numbers don't have a notion of "positive", that method isn't available. In particular, it is *wrong* to write i as "$\sqrt{-1}$", since $-i$ is also a square root of -1, and neither of them is "positive" or "negative".[15] It *never* makes sense to talk about "*the* square root" of a negative number.

Another thing that's meaningless is to combine numbers in two different systems, or compare them for equality (or inequality, see section 1.7). For example:

[15] Paul Sally, a famous and famously cantankerous mathematician, used to inveigh against the common pronunciation of "$-x$" as "negative x", because if x happens to be a negative number then $-x$ is a *positive* number, and if x is a complex number then neither x nor $-x$ might be "negative". He always insisted that "$-x$" be pronounced as "minus x". I've mostly given up that battle as a lost cause (although a little something still dies inside me when I hear "negative x"), but it's still important not to fall into the trap of thinking that a number is a *negative number* just because it has a $-$ sign in front of it.

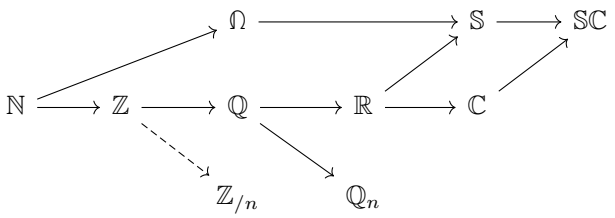
Non-Theorem 1.15. *Suppose* $x \in \mathbb{Q}_{10}$ *and* $y \in \mathbb{S}$*, and* $x = y + 1$*. Then* $x - 1 = y$*.*

This is *not true or false but meaningless*: we can't even ask whether a 10-adic number equals a surreal number. Again, we allow ourselves a bit of leeway here to implicitly consider numbers as living in larger worlds, for instance:

Theorem 1.16. *Suppose* $x \in \mathbb{Z}$ *and* $y \in \mathbb{Q}$*, and* $x = 2y$*. Then* $y = \dfrac{x}{2}$*.*

Technically we should object to this, but we allow it since every integer is also a rational number.

The "larger words" that are generally allowed can be drawn like this:



Here Ω denotes the "ordinal numbers", which I'll introduce in section 1.6; while $\mathbb{SC}$ denotes the "sur-complex numbers" obtained by adding i to the surreals, such as $\omega^2 + \frac{i}{\omega}$. In general, the rule is that we can implicitly move a number to the right along one of these arrows, for instance regarding $\frac{1}{3}$ as a real number, or π as a complex number. It's not immediately obvious that every rational number can be regarded as an n-adic number so that we have $\mathbb{Q} \to \mathbb{Q}_n$, but it's true; see Exercise 1.24. And the dashed arrow $\mathbb{Z} \dashrightarrow \mathbb{Z}_{/n}$ must be used with caution: every integer can be regarded as an integer modulo n, but two different integers can become the same in $\mathbb{Z}_{/n}$, such as 6 and 2 modulo 4. This is why we write brackets around the integers modulo n, so we can say $[6]_4 = [2]_4$ even though $6 \neq 2$.

By the way, the fact that $\mathbb{Z}$ sits inside the larger world $\mathbb{Q}$ where division is allowed means that even though we can't in general divide two integers and get another integer, we can *cancel* nonzero integers that are multiplied by both sides of an equality. For instance, if $a, x, y \in \mathbb{Z}$ and $ax = ay$, and $a \neq 0$, we can divide by a in $\mathbb{Q}$ and get $x = y$ in $\mathbb{Q}$, hence also in $\mathbb{Z}$. Similarly, since $\mathbb{N}$ sits inside $\mathbb{Z}$ where subtraction is allowed, we can cancel natural numbers that are added to both sides of an equality: if $u, v, w \in \mathbb{N}$ and $u + v = u + w$, then $v = w$. However, $\mathbb{Z}_{/n}$ and $\mathbb{Q}_n$ do *not* sit inside any larger worlds where division is allowed,[16] and so we cannot cancel a from an equation like $ax = ay$ in those worlds even if $a \neq 0$ (indeed, recall that Theorem 1.10 fails in those worlds).

In general, if a statement is meaningless because it tries to do things that don't make sense, we say that it *doesn't typecheck*. This should be familiar to programmers; for instance, if you ask Python to evaluate `3 + "hello"`, it complains

[16] Unless n is prime, in which case division already exists in $\mathbb{Z}_{/n}$ and $\mathbb{Q}_n$.

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Python is called a *dynamically typed* language, because this sort of error is only reported at run-time. Mathematics is more akin to a *statically typed* programming language, like C/C++ or Java, in which type errors are reported at compile-time,[17] but the principle is the same.

1.6 A digression about counting

Some mathematicians do not include 0 in the natural numbers. This choice has the weight of history on its side, since 0 was only discovered about 2000 years ago (with a variation of some ±500 years depending on which ancient civilization you look at), but little else to recommend it.

The simplest argument for the inclusion of 0 is that the "counting numbers" should be what we use to count things. But if I try to count the number of purple dragons on this page, then (unless someone has been doodling in the margins) I've got to count 0 of them.

If you reply that by "counting numbers" you mean the numbers that you *say* when you *point* at the things being counted (rather than the possible *answers* you get from counting them), and those numbers start at 1, then you have a different problem. Namely, if you say numbers starting at 1 when you point at things, then how do you decide how many things there are when you've finished pointing? The usual rule is that the number of things is the *last number you said*. But if you try to count the number of purple dragons on this page, well, it doesn't take very long to point at all of them and say numbers, but then *there is no last number that you said*!

A better way to count things (which I learned from John H. Conway) is to say natural numbers starting at 0 when you point at them. Then when you're all done, the number of things is the *smallest natural number you didn't say*. So if I want to count the number of stars here:

⋆ ⋆ ⋆ ⋆

I would point at them and say "zero, one, two, three" and then the number of stars is the smallest natural number I didn't say, namely 4. With this method, I have no trouble counting the number of purple dragons on this page: I'm done pointing at them pretty quickly, and then the smallest natural number I didn't say is the smallest natural number of all, namely 0.

This is an example of something that happens a lot in mathematics, which I call "the power of triviality". Quite often it happens that we consider a general notion which has one or more "trivial cases", often pertaining to the number 0.

[17] Traditionally, the "formal foundation" for mathematics has been a form of "pure set theory" such as "Zermelo–Fraenkel set theory", which is actually completely untyped — more like assembly language than either Java or Python. However, like assembly language, this is better regarded as a *compilation target* rather than a language in which we actually write. Moreover, nowadays there are alternative formal foundations for mathematics that are natively statically typed, such as *Martin-Löf type theory* or *homotopy type theory*.

It is a tempting trap (which even many professional mathematicians fall into) to exclude the trivial cases by definition (such as by defining the natural numbers to not include 0), or to treat them separately as a special case — we might call this the "trap of triviality". However, in nearly all situations, if the general definitions are formulated correctly and carefully, they will *include* any trivial cases with the correct behavior automatically, such as with Conway's method of counting.

By the way, Conway's method also has a nice extension to counting *infinite* things. Suppose I want to count the positive multiples of three:

$$3,\ 6,\ 9,\ 12,\ \ldots$$

If I point at them one by one and say numbers starting with 1, then it takes a while to finish, but when I'm done there is once again no last number that I said. But if I instead count them with numbers starting at 0, then it takes the same amount of time to finish, and when I'm done there isn't a smallest *natural* number that I didn't say (since I used them all) — but there are more general *numbers* I didn't say, like ω.

Indeed, there is a subclass of the surreal numbers called the *ordinal numbers* that can be used for this purpose, and ω is the smallest ordinal number that's greater than all the natural numbers. (There are *surreal* numbers smaller than ω but still greater than all the natural numbers, like $\omega - 1$ and $\frac{\omega}{2}$ and $\sqrt{\omega}$, but none of them are ordinal numbers.) More generally, like natural numbers, ordinal numbers can be added and multiplied but not (in general) subtracted or divided; thus $\omega + 1$, 2ω, $\omega^2 + 3\omega + 2$, and so on, are all ordinal numbers. There is also an operation of raising ω to a surreal power, satisfying the usual laws like $\omega^{x+y} = \omega^x \omega^y$, and $\omega^x < \omega^y$ if $x < y$, and such that ω^x is an ordinal number if x is. Thus ω^ω, ω^{ω^ω}, and so on, are also ordinal numbers. So natural numbers are for counting *finite* collections, while ordinal numbers can be used in the same way to count *infinite* collections. (In fact, Conway himself *discovered*[18] the surreal numbers, although the ordinal numbers had been around for a long time before that.) I'll write Ω (the capital Greek letter omega, in double-struck font) for the collection of ordinal numbers; we'll meet them again in section 5.3.

[18] There is a long-standing debate about whether we should say that mathematics is *discovered* or *invented*, which quickly gets into deep philosophical waters. My official rule of thumb is that mathematical *objects* are discovered, but mathematical *techniques* are invented. But this is actually a lie, kind of like saying that the difference between fantasy and science fiction is that science fiction is about what's possible while fantasy is about what's impossible; since then on a strict reading, time-travel and faster-than-light spaceships should be considered fantasy. The real difference is that in science fiction you make impossible things happen by pushing a button rather than chanting a spell. Similarly, the real difference in my head is that "natural-seeming" or "inevitable" parts of mathematics are discovered, while "artificial-seeming" parts of mathematics are invented. And yes, that's totally subjective.

In any case, I stand by my statement that Conway *discovered* the surreal numbers, although given what I've said about them so far it's not at all clear why they are inevitable. If you want to know more, check out the book *Surreal Numbers: How Two Ex-Students Turned On to Pure Mathematics and Found Total Happiness* by Donald Knuth, or Conway's book *On Numbers and Games*.

There are also more advanced reasons for the inclusion of 0 in the natural numbers. For instance, the natural numbers with 0 are "the free monoid on one generator" and "the initial unitary semiring", while those without 0 have no such nice properties. Properties like that are studied in the mathematical subjects of *abstract algebra* and *category theory*.

1.7 Inequalities

As you probably remember from algebra, many of the same rules for working with *equalities* also apply to *inequalities* such as $<$, $>$, $\leq$, and $\geq$ (when working in an ordered number system, unlike $\mathbb{C}$ or $\mathbb{Z}_{/n}$). Specifically, you can always *add or subtract* things from both sides of an inequality, but you can only multiply or divide by *positive* numbers if you want to preserve the direction of the inequality. If you multiply or divide by a negative number, it reverses the direction of the inequality. You can always simplify either side of an inequality, or substitute an *equality* into it, and inequalities of the same sort can be chained. Here is an example.

Theorem 1.17. *Suppose $x \in \mathbb{R}$, and $x > 3$ and $x < 4$. Then $x^2 + 12 < 7x$.*

Proof of 1.17. Subtracting 3 from $x > 3$, we get $x - 3 > 0$. Subtracting 4 from $x < 4$, we get $x - 4 < 0$. Multiplying these two inequalities we get

$$\begin{aligned} (x-3)(x-4) &< 0 \\ x^2 - 7x + 12 &< 0 \\ x^2 + 12 &< 7x. \end{aligned}$$

□

In this case, we were able to work forwards directly from the given inequalities and end up with the goal. However, in other cases we may need to modify the givens first.

Theorem 1.18. *Suppose $x \in \mathbb{R}$ and $x < \frac{1}{3}$. Then $3x - 2 < x - 1$.*

It is not immediately obvious what algebra we can do to $x < \frac{1}{3}$ to end up with $3x-2 < x-1$. To figure this out, we can do *scratch work*, which is not part of the proof, and therefore does not need to follow the rules of always working forwards. In our scratch work, we can start with the goal $3x - 2 < x - 1$ and try to simplify it:

$$\begin{aligned} 3x - 2 &< x - 1 \\ 2x - 2 &< -1 \\ 2x &< 1 \\ x &< \tfrac{1}{2}. \end{aligned}$$

Now we can expect to be able to reverse this algebra in the proof to *prove* the goal, and for that we will need to start with $x < \frac{1}{2}$. This is different than our given $x < \frac{1}{3}$, but fortunately it follows from it.

Proof of 1.18. Since $x < \frac{1}{3}$ and $\frac{1}{3} < \frac{1}{2}$, we have $x < \frac{1}{2}$. Therefore:

$$\begin{aligned} x &< \tfrac{1}{2} \\ 2x &< 1 \\ 2x - 2 &< -1 \\ 3x - 2 &< x - 1. \end{aligned}$$

□

Importantly, *functions* cannot in general be applied to inequalities. For instance, if $x < y$, we cannot conclude that $x^2 < y^2$; it might be the case that $x = -3$ and $y = 2$. In this case, it is true if we additionally assume x and y are positive: if $0 < x < y$ then $0 < x^2 < y^2$, and in fact also $0 < \sqrt{x} < \sqrt{y}$; this holds for real numbers and even also surreal numbers. But in general, preserving inequalities is a nontrivial fact about a function.[19]

Reasoning about *disequalities* such as $x \neq y$ has similar issues. We can add and subtract from both sides of a disequality (we did this already in proving Theorem 1.3), but we can only multiply or divide by nonzero numbers (assuming we're in a number system with division), and we can't apply functions.[20] We will learn a better way to reason about disequalities in chapter 4.

Exercises

Exercise 1.1. Suppose $x, y \in \mathbb{Q}$ and $x = 3y$ and $1 - x = 4y$. Prove $y = \frac{1}{7}$.

Exercise 1.2. Suppose $x, y \in \mathbb{R}$ and $x + y = xy$ and $x \neq 1$. Prove $y = \frac{x}{x-1}$.

Exercise 1.3. Suppose $x, y \in \mathbb{Z}$ and $x = y$. Prove $3x^2 - xy = y^2 + xy$.

Exercise 1.4. Suppose $x, y, z \in \mathbb{R}$ and $3x = 2y$ and $3y = 2z$. Prove $y^2 = xz$.

Exercise 1.5. Suppose $x, y \in \mathbb{Q}$ and $x = y - 1$ and $x^2 - y^2 = 3$. Prove $x = -2$.

Exercise 1.6. Suppose $u, v \in \mathbb{C}$ and $3u - v = \pi$ and $v + u^2 = 2u$. Prove $u^2 = \pi - u$.

Exercise 1.7. Suppose $a, b, c \in \mathbb{Z}$ and $a + b + c = 13$ and $a - b + c = 11$ and $c - a = 12$. Prove $a = 0$.

Exercise 1.8. Suppose $a, b, c, d \in \mathbb{R}$ and $a = b^2 + 2d$ and $c = d^2 - 2b$. Prove $ab + cd = b^3 + d^3$.

Exercise 1.9. Suppose $x, y \in \mathbb{Z}_{/2}$. Prove $(x + y)^2 = x^2 + y^2$.

Exercise 1.10. Suppose $x, y \in \mathbb{Z}_{/3}$. Prove $(x + y)^3 = x^3 + y^3$.

Exercise 1.11. Suppose $x, y \in \mathbb{Z}_{/4}$. Prove $(x + y)^4 = x^4 + x^2y^2 + x^2y^2 + y^4$.

[19] This fact has a name, which you may remember from calculus: a function f is *increasing* if whenever $x < y$ we have $f(x) < f(y)$.

[20] There is also a name for functions that can be applied to disequalities, as you will probably learn in supplementary material: they are called *injective* or *one-to-one*.

Exercise 1.12. Suppose $x \in \mathbb{C}$ and $x^2 + 1 = x - i$ and $x \neq i$. Prove $x = 1 - i$.

Exercise 1.13. Suppose $x, y \in \mathbb{S}$ and $xy = \omega - x$ and $x + y = \omega + x$. Prove $x^2 + x^2y + xy^2 = \omega^2$.

Exercise 1.14. Suppose $x, y \in \mathbb{Z}_{/15}$ and $x + y = [5]_{15}$ and $x - y = [3]_{15}$. Prove $x^2 = y^2$.

Exercise 1.15. Suppose $x, y \in \mathbb{Q}_{10}$ and $xy = 0$. Prove $(x + y)^2 = x^2 + y^2$.

Exercise 1.16. Suppose $x, y \in \mathbb{C}$ and $x^2 = -5$ and $y^2 = -5$ and $x \neq y$. Prove $y = -x$.

Exercise 1.17. Suppose $x, y \in \mathbb{R}$ and $xy = 1$ and $x + y = 3$ and $x > y$. Prove $x - y = \sqrt{5}$.

Exercise 1.18. Suppose $x \in \mathbb{S}$ and $y \in \mathbb{R}$, and $0 < x < \frac{1}{\omega}$ and $0 < y$. Prove $\sqrt{x} < y$. *(Hint: remember $\sqrt{\omega}$ is greater than all real numbers.)*

Exercise 1.19. Find a counterexample to the following conjecture:
Conjecture. Suppose $x \in \mathbb{R}$ and $x^2 = 2x + 3$. Then $x = -1$.

Exercise 1.20. Find a counterexample to the following conjecture:
Conjecture. Suppose $x, y \in \mathbb{R}$ and $x^2y - x = y - xy^2$. Then $xy = 1$.

Exercise 1.21. Find a counterexample to the following conjecture:
Conjecture. Suppose $x \in \mathbb{C}$ and $x^2 + 1 = 0$. Then $x = i$.

Exercise 1.22. Find a counterexample to the following conjecture:
Conjecture. Suppose $x \in \mathbb{Z}_{/16}$ and $x^2 = [9]_{16}$. Then either $x = [3]_{16}$ or $x = [-3]_{16}$.
(Note that although $([13]_{16})^2 = [169]_{16} = [9]_{16}$, for instance, $x = [13]_{16}$ is not a counterexample because $[13]_{16} = [-3]_{16}$.)

Exercise 1.23. What is wrong with the following non-proof?

Non-Theorem. Suppose $x, y \in \mathbb{R}$ and $x = y$. Then $2 = 1$.

Non-Proof.

$$\begin{aligned} x &= y \\ x^2 &= xy \\ x^2 - y^2 &= xy - y^2 \\ (x - y)(x + y) &= (x - y)y \\ x + y &= y \\ 2y &= y \qquad \text{(substituting } x = y\text{)} \\ 2 &= 1. \end{aligned}$$

Exercise 1.24. Remember that 10-adic numbers can be infinite to the *left* of the decimal point, but must be finite to the *right*. So the usual infinite repeating decimal expansions of fractions like $\frac{1}{3} = 0.33333\cdots$ and $\frac{1}{7} = 0.142857142857\cdots$ are not allowed.

(a) Multiply the 10-adic number $\cdots 666667$ by 3 and show that you get 1. Therefore, $\cdots 666667 = \frac{1}{3}$ in $\mathbb{Q}_{10}$, in the same sense that $\cdots 99999 = -1$.

(b) Find a 10-adic representation of $\frac{1}{7}$.

(c) Find a 10-adic representation of $\frac{1}{2}$. *(Hint: don't overthink it!)*

(d) Find a 10-adic representation of $\frac{1}{6}$.

In fact, although not *everything* can be divided by in $\mathbb{Q}_{10}$, we can divide by any nonzero *ordinary* integer in $\mathbb{Q}_{10}$.

Chapter 2

Propositional proofs

In logical parlance, a *proposition* means a statement that has a truth value (either true or false).[1] In mathematics, we are concerned with *mathematical* propositions that make claims about numbers or other mathematical objects, and a proof is a way to establish that a given mathematical proposition is true.

The equations and inequalities we worked with in chapter 1 were examples of propositions. However, to state and prove anything more complicated than simple algebraic statements, we need *compound propositions*, which are put together from basic propositions like equalities using *logical operators* or *connectives* such as "and", "or", and "not". In this chapter we will learn these operators and how to deal with them in proofs.

2.1 The algebra of truth values

A *truth value* is a possible answer when we ask whether a statement is true. Although philosophers and logicians investigate worlds with many possible truth values, for the purposes of these notes, there are exactly *two* truth values:

- *true*, which we denote $\top$ (a stylized "T" for "true"), and
- *false*, which we denote $\bot$ (an upside-down $\top$).

Therefore, any statement that is not true must be false, and any statement that is not false must be true. Programmers and electrical engineers sometimes use 1 and 0 in place of $\top$ and $\bot$, respectively.[2]

[1] Although you should be aware, as noted in footnote 2 on page 11, that "proposition" is also commonly used as a label for a "less important theorem".

[2] The word "respectively" after a relation stated between two lists means that each element of the first list stands in that relation to the corresponding element of the second list. For example, in this case, "1 and 0 in place of $\top$ and $\bot$, respectively" means "1 in place of $\top$, and 0 in place of $\bot$". Another way of saying the same thing is "1 (respectively 0) in place of $\top$ (respectively $\bot$); in this usage "respectively" is sometimes shortened to "resp." However, overuse of this idiom can be more confusing than enlightening.

We denote the collection of truth values by $\mathbb{B}$, for "Boolean" (after George Boole). It is possible to regard $\mathbb{B}$ as a sort of "number system" analogous to $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$, etc., where instead of addition, subtraction, and so on, the operations on $\mathbb{B}$ are logical operations. There are four basic logical operators: "and" (denoted $\wedge$), "or" (denoted $\vee$), "if-then" (denoted $\Rightarrow$), and "not" (denoted $\neg$). We will spend the rest of this chapter defining and studying the first three; the last one we postpone to chapter 4.

From this algebraic point of view, it may seem as if we could simply "compute" the truth value of any proposition, the same way we can compute the value of an arithmetic expression like $3 \cdot (2 + 4)^2 - 17$. So why do we need proofs? At the moment, our answer to this is the same reason that we do algebra rather than just arithmetic: we are interested in propositions that involve *variables*, and their truth value may depend on the values of those variables. For example, a statement like "either $x < -2$ or $x > 2$" is true if $x = 3$, but false if $x = 1$. But a compound statement involving such might *always* be true, even if its constituent pieces are not; for instance, "if either $x < -2$ or $x > 2$, then $x^2 > 4$" is always true. Proofs allow us to establish the truth of such general statements even though we cannot "compute" a definite truth value for the intermediate pieces. (In chapter 3 we will have a more general answer to this question.)

2.2 What is a proof?

A proof is an argument or process that leads from hypotheses to a conclusion following valid *proof rules*, which are set up in such a way that having a correct proof guarantees that whenever the hypotheses are true, the conclusion is also true. Proof rules are not something you or anyone else can "make up": there is a finite and unchanging collection of such rules. In chapter 1 we mentioned the proof rules that pertain to algebra.

In this chapter we will learn the proof rules that pertain to the logical operators $\wedge, \vee, \Rightarrow$ (and the truth values $\top, \bot$). These rules are not haphazard; they are organized according to the operators, with each operator being associated to some number of *rules to prove it* and *rules to use it*. Moreover, each rule should make sense according to the intended meaning of that operator. Therefore, although you must learn all the proof rules, this is not arbitrary memorization.

Thus defined, a proof is an abstract object, which can be represented in many ways. We will use two such representations: a *graphical* representation and an *English* description. The graphical representation is precise, visually evocative, and can be created interactively using a computer; but it is not standard in mathematics and becomes very verbose and space-intensive for significant proofs. The English description is more concise and standard in mathematics. Thus our eventual goal will be to learn to write all proofs in English, using the graphical representation to help us learn the proof rules.

In the remaining sections of this chapter, we'll gradually develop all the proof rules associated to the operators $\wedge$, $\vee$, and $\Rightarrow$ (we save $\neg$ for chapter 4). Each

section ends with some exercises. If you are reading these notes on your own, ***DO NOT*** skip these exercises! It is essential for understanding that you solve the exercises in one section before moving on to the next section. If you have an instructor, he or she should assign at least some of the exercises as homework or classwork.

2.3 And (∧)

The simplest logical operator is "and", also known as *conjunction*. If P and Q are statements, we write $P \wedge Q$ for the statement "P and Q".[3] Since "P and Q" asserts that both P and Q are true, the truth value of $P \wedge Q$ is $\top$ if the truth values of P and Q are both $\top$, and otherwise it is $\bot$. Since there are only two possible values for each of P and Q, there are only four possible combinations; thus we can completely define $\wedge$ by a so-called *truth table*:

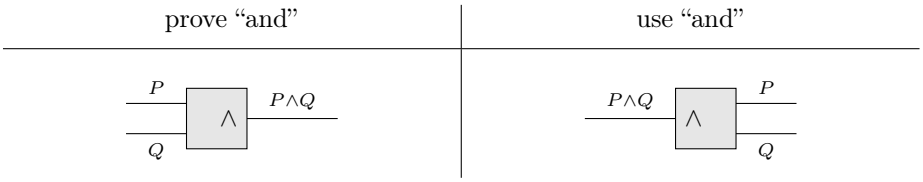
P	Q	$P \wedge Q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$

When writing in English, we sometimes say "both P and Q" instead of "P and Q" for emphasis. The English connective "but" also has the same logical meaning: "P but Q" also means $P \wedge Q$; the only difference is a connotation that Q is somehow surprising given P.

The proof rules for $\wedge$ are very simple and express the fact that $P \wedge Q$ asserts that P and Q are both true.

- The rule to *prove "and"* says that in order to prove $P \wedge Q$, what we have to do is to prove P and separately prove Q.
- The rule to *use "and"* says that if we know (or have assumed) that $P \wedge Q$ is true, we can deduce that P is true and separately that Q is true.

The graphical representations of these rules are as follows:



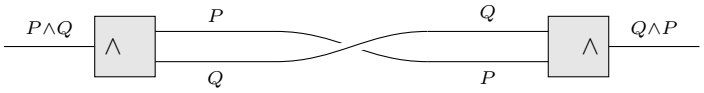
In our graphical proofs, truth "flows" from left to right. We start with wires on the left labeled by our hypotheses, and want to get out a wire on the right labeled by our conclusion. Thus, the left-hand rule above represents the *prove "and"* rule, and the right-hand one represents the *use "and"* rule.

[3] Many programming languages write "and" as `&` or `&&`.

As an example, we can put these together to prove that $\wedge$, like addition and multiplication, is *commutative*. This is an "abstract" proof where P and Q represent arbitrary statements, as in the discussion above.

Theorem 2.1 (Olorin[4] 1-2-5)**.** *Suppose $P \wedge Q$. Then $Q \wedge P$.*

Graphical Proof of 2.1.



□

That is, we start with our hypothesis that $P \wedge Q$ is true, we use the *use "and"* rule to break it up into knowing that P is true and separately that Q is true, then we use the *prove "and"* rule to put those together in the other order to deduce that $Q \wedge P$ is true, which was our goal.

A proof like this is so simple that it's almost difficult to describe in English. In fact, normally when writing proofs in English we don't even bother to mention when we use the *prove "and"* and *use "and"* rules as they are considered so obvious. However, for the present, we can write these proofs in English using the following general principle:[5]

Principle of English Proof 1 (Since)**.** In an English proof, we write "Since P and Q, we have R" to mean that we already know P and Q, and we are deducing R from them by a rule that the reader should be able to guess.

There are many possible variations. Instead of two facts P and Q we can have one, three, or any number. In place of "since" we can use an equivalent word like "because". If one of the facts (or the only one) was just mentioned in the sentence immediately before, we don't need to restate it; in this case we generally use a word like "hence", "thus", or "therefore". (Some people use the symbol $\therefore$ to mean "therefore", but I find this very hard to notice on the page, so I do not recommend it.) The filler phrase "we have" can be omitted; it is mainly used when Q ends with a variable or symbol and R begins with a variable or symbol, according to a general readability principle that distinct mathematical formulae should be separate by words and not just punctuation. Finally, to indicate what rule is being used we can end the sentence with "by ⟨rule⟩".

Using this principle, therefore, we can write the above proof in English:

English Proof of 2.1. Since $P \wedge Q$, we have P and Q. And since Q and P, we have $Q \wedge P$. □

Now it's your turn. As you work on the exercises below, you may find yourself feeling lost or stumped as to what to do. Whenever this is the case (in

[4] See section 2.3.1.

[5] We will encounter a number of Principles of English Proof throughout the notes. They are all repeated in appendix A for ease of reference.

any proof, not just these exercises), I recommend the first thing that you do is the following.[6]

Proof Guidance 1 (Follow the structure)**.** Follow the logical structure of the givens (wires currently coming from the left) and goal (unattached wire currently coming from the right). For instance, if you have a given that is a ∧ statement, try connecting it to a *use "and"* block; and if you have a goal that is a ∧ statement, try connecting it to a *prove "and"* block.

This guidance won't solve every proof for you, but very often it will get you past the feeling of being lost at the beginning of a proof, and lead you to a place where you can figure out how to make progress.

2.3.1 Introduction to Olorin

As mentioned in the introduction, the in-development graphical proof assistant *Olorin*. All of the "abstract" theorems and exercises in these notes (those that don't refer to specific number systems or algebraic operations — specifically, this is those in sections 2.3, 2.5, 2.7, 2.9, 2.10, 3.4, 3.6 and 4.1) can be done in Olorin.

Olorin is structured like a game: there are four "worlds", each divided into a number of "stages" having several "levels" each. Each level thus has a number like "1-2-3", meaning the third level in the second stage in the first world. All the theorems and exercises that can be done in Olorin are labeled with their world-stage-level number in parentheses, so you can find them more easily.[7] You may have already noticed such a label on Theorem 2.1.

If you are using Olorin (which I highly recommend), I suggest you just work your way through the appropriate worlds and stages in order, rather than jumping around to stick with the exercises as they are listed here. I've listed the most important exercises here so that instructors can refer to them and assign them (although I encourage instructors to simply assign certain stages of Olorin as well), and so that they are still available to readers who are unable or unwilling to use Olorin.

Olorin also has three *difficulty settings*: Novice, Adept, and Master. On the Novice setting (★☆☆), Olorin will automatically label all the wires, inputs, and outputs for you, so that all you have to do is choose the proof rules and connect them together correctly. Incorrect connections are highlighted in red, and the conclusion and background turn green when the proof is correct.

On the Adept setting (★★☆), you have to label all the wires yourself, but Olorin will still let you know when the connections or labels are incorrect. If the proof is correct — which, on this setting, requires all the labels to be correct as well — the conclusion and background will turn blue.

[6] We will encounter a number of Proof Guidances throughout the notes. They are all repeated in appendix B for ease of reference.

[7] Although at present, these numbers are synchronized with Olorin manually, so some of them are likely to be incorrect. Please let me know when you find incorrect numbers.

Finally, on the Master setting (★★★), you still have to label all the wires yourself, but Olorin will not highlight *which* connections or labels are incorrect. The only feedback you get is that the conclusion won't turn purple until the proof is correct. If this seems harsh, consider that it's still better than what you get when doing a homework exercise on paper, where you don't have *any* immediate feedback about whether it's correct.

Since the ultimate goal is to be able to write proof entirely on your own without help from a proof assistant, you should aim to reach Master on all the Olorin levels. However, it won't do you much good if you solve a given level on Novice and then immediately go back and do the same level on Adept or Master, since the proof you wrote in the Novice version with copious help will still be in your short-term memory. Thus, I recommend you work through an entire world on one difficulty setting first, then at least partway through the next world on the same difficulty setting, before returning to the first world at a higher difficulty setting.

In addition, for each of the exercises, you should *not only* be creating a graphical proof (with or without Olorin's help), but *also* an English proof. At this point, feel free to write the graphical proof first and then translate it into English. Later on, you'll eventually want to start writing an English proof directly, using the graphs only in your head or as partial scratch work.

Exercises

This section pertains to Stage 1-2 of Olorin, the first substantive stage in "Proposition World". You'll probably want to start out on the Novice difficulty setting.

Exercise 2.3.1 (Olorin 1-2-2)**.** Suppose P. Prove $P \wedge P$.

Exercise 2.3.2 (Olorin 1-2-3)**.** Suppose $P \wedge Q$. Prove P.

Exercise 2.3.3 (Olorin 1-2-6)**.** Suppose $(P \wedge Q) \wedge R$. Prove $P \wedge R$.

Exercise 2.3.4 (Olorin 1-2-7)**.** Suppose $(P \wedge Q) \wedge R$. Prove $P \wedge (Q \wedge R)$.

Exercise 2.3.5 (Olorin 1-2-8)**.** Suppose $P \wedge (Q \wedge R)$. Prove $(P \wedge Q) \wedge R$.

The preceding two exercises say that $\wedge$, like addition and multiplication, is "associative". Because of this, we often simply write $P \wedge Q \wedge R$.

Exercise 2.3.6 (Olorin 1-2-9)**.** Suppose $(P \wedge Q) \wedge (R \wedge S)$. Prove $R \wedge (S \wedge Q)$.

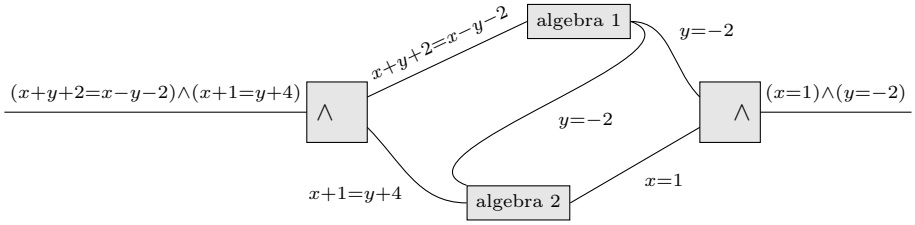
2.4 $\wedge$ with algebra

The proofs in section 2.3 are abstract, involving letters P, Q that represent arbitrary statements. But in practice, we are usually more interested in "concrete" proofs, involving specific statements such as equations and inequalities. The proof rules work for these too, but we have to combine them with the rules of algebra from chapter 1. When we draw a graphical proof involving algebra, we

generally just use a single block labeled "algebra" and include the algebra separately below the proof. (It is technically possible to represent an algebraic proof graphically, but it would be very tedious and inefficient.) Here is an example.

Theorem 2.2. *Suppose* $x, y \in \mathbb{R}$ *and* $(x + y + 2 = x - y - 2) \wedge (x + 1 = y + 4)$. *Then* $(x = 1) \wedge (y = -2)$.

Graphical Proof of 2.2.



Algebra 1:

$$\begin{aligned} x + y + 2 &= x - y - 2 \\ y + 2 &= -y - 2 \\ 2y &= -4 \\ y &= -2 \end{aligned}$$

Algebra 2:

$$\begin{aligned} x + 1 &= y + 4 \\ x &= y + 3 \\ x &= (-2) + 3 = 1. \end{aligned}$$

□

This proof merits a remark or two about how we draw graphs. If there is not enough horizontal space on the page, we can feel free to draw wires that curve around from a block somewhere towards the right back to a block somewhere more towards the left. However, to make the left-to-right "flow of logic" clear, all the *inputs* of a block should always go in on *its left*, and all the *outputs* of a block should always come out on *its right*. Thus, the wire from "algebra 1" to "algebra 2" can go "backwards" from right to left, but we don't draw it as a straight line; instead we make it curved, so it can come out on the right of "algebra 1" and go in the left of "algebra 2". (In Olorin, wires like this bend with right angles rather than curvily; you can draw whichever you like in your proofs.)

English Proof of 2.2. Since $(x + y + 2 = x - y - 2) \wedge (x + 1 = y + 4)$, we have

$x + y + 2 = x - y - 2$ and $x + 1 = y + 4$. From the first we have

$$\begin{aligned} x + y + 2 &= x - y - 2 \\ y + 2 &= -y - 2 \\ 2y &= -4 \\ y &= -2 \end{aligned}$$

Now from the second we have

$$\begin{aligned} x + 1 &= y + 4 \\ x &= y + 3 \\ x &= (-2) + 3 = 1. \end{aligned}$$

Therefore, $(x = 1) \wedge (y = -2)$. □

In fact, the rules for $\wedge$ are so tautological that, like Molière's *bourgeois gentilhomme* who has been speaking in prose all his life without realizing it, you have been using the $\wedge$ rules all your life (or at least since chapter 1). For instance, look back at our very first theorem:

Theorem 1.1. *Suppose $x, y, z \in \mathbb{R}$, and that $x + 1 = y$ and $x - 1 = z$. Then $x^2 = yz + 1$.*

An equivalent way to say this would be

Theorem 1.1 (with $\wedge$)**.** *Suppose $x, y, z \in \mathbb{R}$, and that $(x+1 = y) \wedge (x-1 = z)$. Then $x^2 = yz + 1$.*

The proof of this version would be exactly the same as that of the original, except that it would start with a *use "and"* to break $(x + 1 = y) \wedge (x - 1 = z)$ up into $x + 1 = y$ and separately $x - 1 = z$. For the most part, when writing mathematics in English, we omit symbols like $\wedge$ in favor of words like "and", and never mention use of the $\wedge$ rules. However, at least for this section, please be explicit about using them, for practice.

We will return to this theme again in sections 2.8 and 3.9.

Exercises

Continue to give both a graphical proof and an English proof for each exercise. But Olorin doesn't (yet) do algebra, so you'll need to write your graphical proofs on paper (or virtual paper), with one or more blocks labeled "algebra" and the algebra done below the graphical proof, as we did for Theorem 2.2. If using physical "letter" or A4 size paper, you may want to turn it sideways ("landscape") to give yourself more horizontal space for graphical proofs.

Exercise 2.4.1. Suppose $x, y \in \mathbb{Q}$ and $(x = 3y) \wedge (1 - x = 4y)$. Prove $(x = \frac{3}{7}) \wedge (y = \frac{1}{7})$.

Exercise 2.4.2. Suppose $x, y \in \mathbb{R}$ and $(x - y = 3x^2) \wedge (y = x - x^2)$. Prove $(x = 0) \wedge (y = 0)$.

Exercise 2.4.3. Suppose $x, y \in \mathbb{S}$ and $(x + y = \omega) \wedge (x - y = \frac{1}{\omega})$. Prove $(x^2 = y^2 + 1) \wedge (x = \frac{\omega^2+1}{2\omega})$.

2.5 Or (∨)

Remark 2.5.1. *There is room for debate about which logical operator to introduce next: ∨ or ⇒. In these notes I have chosen ∨, because I believe its* meaning *is easier to understand, and more evidently useful in concrete algebraic proofs at this point. However, the* rules *of ⇒ are a little simpler, and so Olorin introduces it first. You (or your instructor) are free to follow Olorin and skip ahead to section 2.7 before coming back to this section and the next.*

The operator $\vee$ is called *disjunction*. If P and Q are statements, $P \vee Q$ denotes the statement "P or Q".[8] However, the truth value of this is not immediately obvious. If P and Q are both false, then certainly $P \vee Q$ should be false. And if one of P and Q is true and the other is false, then certainly $P \vee Q$ should be true. The question is what should happen when both P and Q are true.

In everyday[9] language, we sometimes use the word "or" in a way suggesting that it is false if both of its sub-phrases are true, such as "eat your vegetables or you'll get no dessert" — the child being addressed here would certainly be incensed if the vegetables were eaten *and* no dessert were forthcoming. This interpretation is called the *exclusive or*. However, in mathematics the *inclusive or*, which allows "P or Q" to be true if both P and Q are true, is much more useful, and this is what we mean by $\vee$. Thus, its truth table is

P	Q	$P \vee Q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$

When writing in English, we sometimes write "either P or Q" instead of "P or Q", but they mean the same thing: both are an *inclusive* or.

It can be hard to remember which of "and" and "or" is $\wedge$ and which is $\vee$. The best mnemonics I've heard are that the symbol $\wedge$ looks vaguely like both the capital letter "A" and the lowercase letter "n", both of which appear in "And" but not in "Or".

The rules to prove an "or" are quite simple: since $P \vee Q$ says that one of P or Q is true, to prove it, we must choose one of them and show that it is true.

[8] Many programming languages write "or" as `|` or `||`.

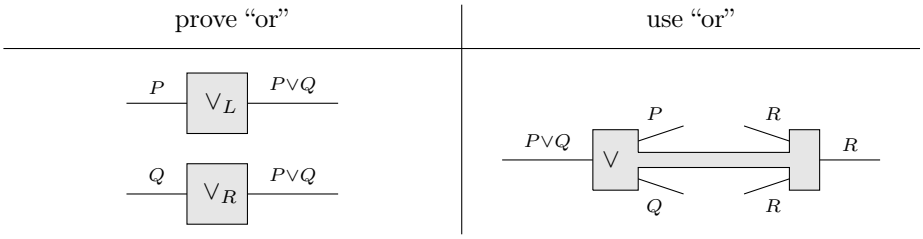
[9] The adjective "everyday", with no space, means "common, daily". The adverbial phrase "every day", with a space, means "on each day". Don't confuse them.

- The first rule to *prove "or"* says that to prove $P \vee Q$, we can prove P.
- The second rule to *prove "or"* says that to prove $P \vee Q$, we can prove Q.

The rule to use an "or" is more complicated. If we know $P \vee Q$, we know that one of P or Q is true, but we don't know which one, so we can't *deduce* either P or Q. However, if we can show that some other consequence follows *both* from P and also from Q, then that consequence must definitely be true, since one of those two cases most hold. Thus:

- The rule to *use "or"* says that if we know (or have assumed) $P \vee Q$, and we are trying to prove some other proposition R, then we can divide the proof into cases. In the first case we assume, hypothetically, that P is true, and we must prove R is true under that assumption; while in the second case we assume, hypothetically, that Q is true, and we must also prove R under *that* assumption.

The graphical representations of these rules are as follows:



The graphical "prove" rules on the left are straightforward. The notations $\vee_L$ and $\vee_R$ are intended to distinguish the two, based on whether the input statement appears on the *Left* or the *Right* of the "or" statement in the output. However, when writing graphical proofs by hand on paper, you can generally just write $\vee$ for either one.

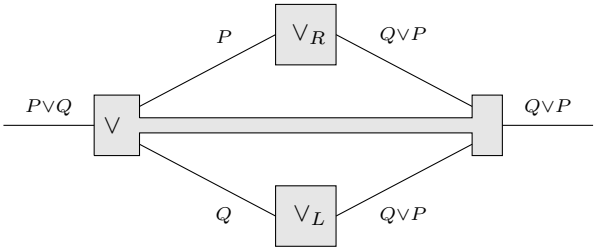
The "use" rule on the right is more complicated. Recall that truth flows from left to right: so this rule applies, as stated above, if we know $P \vee Q$ and we are trying to prove some arbitrary other statement R. The "bar" running down the middle of the "barbell" shape separates the two *cases* or *sub-proofs*. On the top we get to assume P hypothetically, so we have a new wire coming from the left labeled P; and similarly on the bottom we get a new wire labeled Q. In both cases we must prove R, but the two proofs may be different; so we need two wires coming out the right labeled R, one on the top and one on the bottom. The gaps on top and bottom are to be filled with the "bodies" of the two cases.

It is *absolutely essential* that the assumption P on the top can *only be used* to prove the goal R *on the top*: it cannot "cross over" the bar to be used on the bottom, nor can it "escape" to be used somewhere in the proof after the *use "or"* rule is finished. Trying to connect the wires in that way makes the proof invalid, and Olorin will reject it. However, givens that we already have *before* we started the *use "or"* can come "inside" and be used in either or both cases: thus the situation is asymmetrical.

Here is an example, showing that ∨, like addition, multiplication, and ∧, is also commutative. Like Theorem 2.1, this is an "abstract" proof with letters P, Q that stand for arbitrary statements.

Theorem 2.3 (Olorin 1-5-5)**.** *Suppose* $P \vee Q$*. Then* $Q \vee P$*.*

Graphical Proof of 2.3.



□

That is, we start with our hypothesis that $P \vee Q$ is true, and to prove our goal of $Q \vee P$, we break into cases with the *use "or"* rule. In the first case we assume P, and deduce $Q \vee P$ by using the second *prove "or"* rule; while in the second case we assume Q, and deduce $Q \vee P$ by using the first *prove "or"* rule.

Note that in this proof I have stretched the "barbell" block for the *use "or"* rule so that both sub-proofs "fit inside" its left-to-right span. I recommend that you do this yourself whenever possible, as it makes the structure of the proof clearer visually, and helps remind you of which wires can be used where. In Olorin, you can do this by hovering over the right- or left-hand side of the barbell until your cursor turns into a "resizing" double-headed-arrow, and clicking and dragging it to make the barbell bigger. This can also help you arrange the boxes on the screen to minimize wire-crossings. However, extending the barbell like this is not strictly necessary, and if you're writing on paper and don't foresee in advance how long the barbell will have to be, you may want to avoid erasing and re-drawing it.

To write this proof in English, we can use the following general principles.

Principle of English Proof 2 (Cases)**.** When dividing a proof into cases, label each case clearly with markers such as "Case 1:" and "Case 2:". If desired for clarity, you can introduce the cases with a phrase like "Since . . . we can divide into cases", and/or conclude them with a phrase like "Since R is true in both cases, we must have R."

Principle of English Proof 3 (Assume)**.** When introducing a *hypothetical assumption*, which is represented graphically by a new wire coming from the left that can only be used inside a "bracket" of some kind (like the two sides of a *prove "or"* rule), indicate it with a word like "suppose" or "assume".

This is the *only* situation in which we use these words. In particular, *do not* say "assume" or "suppose" when you are *deducing* something from previously known facts; for that use something like "since" (PEP 1).

Using these principles (and PEP 1 from section 2.3), here is an English version of the above proof.

English Proof of 2.3. Since $P \vee Q$, we can divide into two cases.
Case 1: Assume P. Therefore, we have $Q \vee P$.
Case 2: Assume Q. Therefore, we have $Q \vee P$.
Since $Q \vee P$ is true in both cases, we must have $Q \vee P$. □

Now it's your turn. If you feel lost at the beginning of a proof, remember Proof Guidance 1 from section 2.3. However, Proof Guidance 1 doesn't tell you what *order* in which to follow it: do you break down the givens first or build up the goals? There is no universal answer to that question; you can develop intuition by experience, but in general you may have to try something, find that it doesn't work, and back up and try something else. Learn to be aware when you've made a choice in constructing your proof that could have been made differently, so that if you end up "painted into a corner" with an impossible task, you can back up and make that choice differently.

In addition, the following guidance is often helpful when using cases.

Proof Guidance 2 (The goal of a case split)**.** Very often, when breaking a proof into cases with the *use "or"* rule, the goal proposition R of that rule should be the *overall goal*: either the desired conclusion of the entire theorem, or the current goal (the wire on the right you're trying to connect to) at the moment when the "or" statement becomes available. In particular, "prove" rules that involve a choice (such as the choice of which *prove "or"* rule to use) usually shouldn't be *outside* the *use "or"* rule to the right: often you'll need to make *different choices in the two cases*, so the choice rules must be used inside the two branches separately.

Exercises

As before, for each of these exercises you should give *both* a graphical proof *and* an English proof. This section pertains to Stages 1-3 and 1-4 of Olorin.

Exercise 2.5.1 (Olorin 1-5-3)**.** Suppose P. Prove $P \vee P$.

Exercise 2.5.2 (Olorin 1-6-1)**.** Suppose $P \wedge Q$. Prove $P \vee Q$.

Exercise 2.5.3 (Olorin 1-5-4)**.** Suppose $P \vee P$. Prove P.

Exercise 2.5.4 (Olorin 1-5-6)**.** Suppose $P \vee (Q \vee R)$. Prove $(P \vee Q) \vee R$.

Exercise 2.5.5 (Olorin 1-5-7)**.** Suppose $(P \vee Q) \vee R$. Prove $P \vee (Q \vee R)$.

The previous two exercises show that $\vee$, like addition, multiplication, and $\wedge$, is *associative*. This justifies our writing simply $P \vee Q \vee R$.

Exercise 2.5.6 (Olorin 1-5-8)**.** Suppose $R \vee (S \vee Q)$. Prove $(P \vee Q) \vee (R \vee S)$.

Exercise 2.5.7 (Olorin 1-6-2)**.** Suppose $P \vee (Q \wedge P)$. Prove P.

Exercise 2.5.8 (Olorin 1-6-3)**.** Suppose $P \wedge (Q \vee R)$. Prove $(P \wedge Q) \vee R$.

Exercise 2.5.9 (Olorin 1-6-4)**.** Suppose $P \wedge (Q \vee R)$. Prove $(P \wedge Q) \vee (P \wedge R)$.

Exercise 2.5.10 (Olorin 1-6-5)**.** Suppose $(P \wedge Q) \vee (P \wedge R)$. Prove $P \wedge (Q \vee R)$.

Exercise 2.5.11 (Olorin 1-6-6)**.** Suppose $P \vee (Q \wedge R)$. Prove $(P \vee Q) \wedge (P \vee R)$.

Exercise 2.5.12 (Olorin 1-6-7)**.** Suppose $(P \vee Q) \wedge (P \vee R)$. Prove $P \vee (Q \wedge R)$.

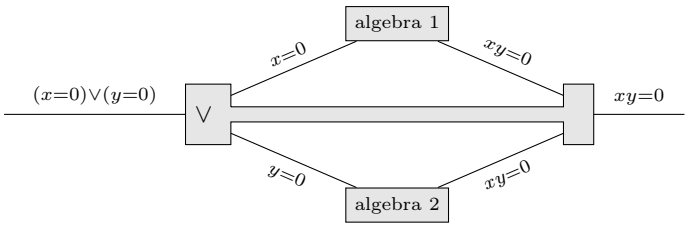
The previous four exercises show that, like addition and multiplication, $\wedge$ and $\vee$ together satisfy a *distributive law* — and not just one distributive law but *both*, with $\wedge$ and $\vee$ each playing the role of both addition and multiplication.

2.6 ∨ with algebra

As in section 2.4, we can combine the rules of section 2.5 with the algebraic rules of chapter 1 to prove theorems involving both $\vee$ and concrete statements like equations and inequalities. Here is an example.

Theorem 2.4. *Suppose* $x, y \in \mathbb{R}$ *and* $(x = 0) \vee (y = 0)$*. Then* $xy = 0$*.*

Graphical Proof of 2.4.



Algebra 1: $xy = 0 \cdot y = 0$.
Algebra 2: $xy = x \cdot 0 = 0$. □

English Proof of 2.4. Since $(x = 0) \vee (y = 0)$, we have two cases.
Case 1: Assume $x = 0$. Then $xy = 0 \cdot y = 0$.
Case 2: Assume $y = 0$. Then $xy = x \cdot 0 = 0$. □

In this proof, we divided into cases based on a hypothesis that was an "or" statement. However, sometimes we may need to divide into cases even if there is no "or" statement in the hypotheses. Instead, we can use a *known fact* or *previously proven theorem* that supplies a true "or" statement for us to apply the *use "or"* rule to.

Proof Guidance 3 (Use previous facts)**.** If it seems like you don't have enough information to complete a proof, ask yourself what general facts or previously proven theorems you know of that could be helpful.

In the case of algebraic proofs, some "or" facts that are often useful include:

- For any x, y in any ordered number system (such as $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{S}$), we have $(x < y) \vee (x \geq y)$, and also $(x < y) \vee (x = y) \vee (x > y)$. This is called the *trichotomy* principle.[10]
- For any $x \in \mathbb{N}$, we have $(x = 0) \vee (x \geq 1)$.
- For any x, y in any number system we have $(x = y) \vee (x \neq y)$.
- For any x, y in $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$, or $\mathbb{S}$ with $xy = 0$, we have $(x = 0) \vee (y = 0)$. (But remember from section 1.5 that this can fail in $\mathbb{Z}_{/n}$ and $\mathbb{Q}_n$.)
- For any $x \in \mathbb{Z}$, either x is even or x is odd.

When using one of these facts in a proof, the variables such as x, y in the facts can be substituted by *any expression* involving the variables appearing in the current proof. It could be one of those variables, or it could be a constant like $\frac{3}{17}$ or 2π, or it could combine variables with constants in an expression like $3ab+2$. (We will explain this as an instance of a more general principle in chapter 3.) Of course, this immediately raises the question[11] of how to choose *what* expressions to use. The answer is:

Proof Guidance 4 (Do scratch work)**.** When you need to specify an expression in the course of a proof, don't just pick something at random. Set the proof aside, pick up another sheet of paper, and do some *scratch work* to figure out what the best choice is. Often, your scratch work will consist of "working backwards" from the goal or some other desired equation or inequality — the sort of algebra that is *not* valid in a proof.

This is easiest to explain with an example.

Theorem 2.5. *Suppose* $x \in \mathbb{R}$. *Then* $|3x - 2| > x - 1$.

To start with, how can we even guess, looking at a theorem like this, that dividing into cases will be useful?

Proof Guidance 5 (Follow function definitions)**.** When the statement of a theorem involves a function or operation that is defined in a "piecewise" way, often it will be useful to divide the proof into cases in an analogous way (using an arithmetic "or" fact such as trichotomy) so that in each case only one "piece" of the function applies.

In particular, note that the absolute value is defined piecewise:

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0. \end{cases}$$

[10] Although you should be aware that there are other contexts in which mathematicians use the symbols "$\leq$" and "$<$" in which trichotomy fails. These more general contexts are called *partial orders*; those that satisfy trichotomy are called *total orders*.

[11] Not "begs the question". Begging the question refers to a fallacy of reasoning that assumes the desired conclusion rather than proving it.

Thus, looking at Theorem 2.5, we can guess that it will be useful to divide into cases so that in each case one of these two pieces applies to the expression $|3x-2|$. In other words, our cases should be $3x-2 \geq 0$ and $3x-2<0$. Doing some algebra to these inequalities in our scratch work, we obtain $x \geq \frac{2}{3}$ and $x<\frac{2}{3}$. Thus, we will use the fact that $(x \geq \frac{2}{3}) \vee (x<\frac{2}{3})$ in our proof.

We are not done with scratch work yet, however. Each of the cases of this proof is similar to Theorem 1.18, and hence needs its own scratch work to figure out where to start from. In fact, the case when $x \geq \frac{2}{3}$ is exactly Theorem 1.18 with the inequality reversed:

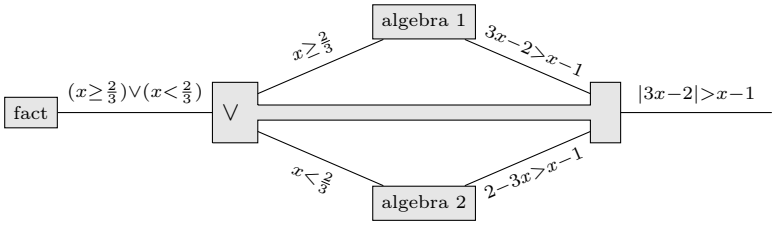
$$\begin{aligned} 3x-2 &> x-1 \\ 2x-2 &> -1 \\ 2x &> 1 \\ x &> \tfrac{1}{2}. \end{aligned}$$

The other case is similar:

$$\begin{aligned} 2-3x &> x-1 \\ 3-3x &> x \\ 3 &> 4x \\ \tfrac{3}{4} &> x. \end{aligned}$$

At last we're ready to write the proof.

Graphical Proof of 2.5.



Algebra 1: Since $x \geq \frac{2}{3}$ and $\frac{2}{3}>\frac{1}{2}$, we have $x>\frac{1}{2}$. Therefore,

$$\begin{aligned} x &> \tfrac{1}{2} \\ 2x &> 1 \\ 2x-2 &> -1 \\ 3x-2 &> x-1. \end{aligned}$$

Algebra 2: Since $x<\frac{2}{3}$ and $\frac{2}{3}<\frac{3}{4}$, we have $x<\frac{3}{4}$. Therefore,

$$\begin{aligned} \tfrac{3}{4} &> x \\ 3 &> 4x \\ 3-3x &> x \\ 2-3x &> x-1 \end{aligned}$$

□

The main remark to make about this proof is that, in seeming contradiction to the general form of the *use "or"* rule, the goal appears to have *changed* in the two branches: instead of the overall goal of $|3x-2| > x-1$, on the top branch it is $3x-2 > x-1$ and on the bottom branch it is $2-3x > x-1$. However, this is only apparent, since in fact $|3x-2| > x-1$ *is the same as* $3x-2 > x-1$ *under the assumption* that $x \geq \frac{2}{3}$ that obtains in the top branch, and similarly on the bottom. This is an important general principle: the goal doesn't *change* in the branches of a proof by cases, but it can get *simplified* based on the different assumptions in the branches.

There are no surprises in the English version.

English Proof of 2.5. Since $(x \geq \frac{2}{3}) \vee (x < \frac{2}{3})$, we have two cases.
Case 1: Assume $x \geq \frac{2}{3}$. Then since also $\frac{2}{3} > \frac{1}{2}$, we have $x > \frac{1}{2}$. Therefore,

$$\begin{aligned} x &> \tfrac{1}{2} \\ 2x &> 1 \\ 2x - 2 &> -1 \\ 3x - 2 &> x - 1. \end{aligned}$$

Case 2: Assume $x < \frac{2}{3}$. Then since also $\frac{2}{3} < \frac{3}{4}$, we have $x < \frac{3}{4}$. Therefore,

$$\begin{aligned} \tfrac{3}{4} &> x \\ 3 &> 4x \\ 3 - 3x &> x \\ 2 - 3x &> x - 1 \end{aligned}$$

□

Here's a theorem that exemplifies a different class of proofs.

Theorem 2.6. *Suppose $n \in \mathbb{Z}$. Then $n^2 + n$ is even.*

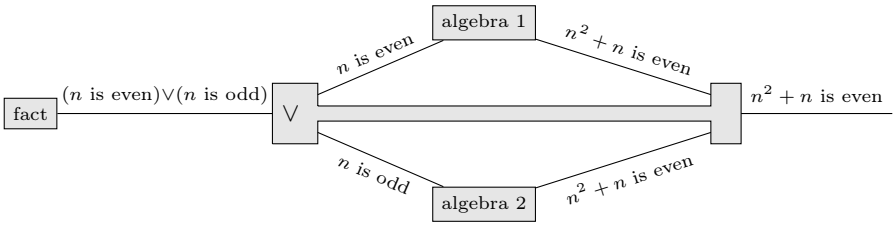
How might we guess, looking at a theorem like this, that dividing into cases will be useful?

Proof Guidance 6 (Make extra assumptions)**.** If it seems like you don't have enough information, ask yourself what additional hypothesis would be useful, and try proving the theorem with that extra hypothesis. If you can do that, there are multiple ways to proceed. First, try modifying your proof to eliminate or weaken the extra hypothesis. Second, look for a general fact or previously proven theorem saying that either your extra hypothesis is true or something else is true, and then try proving the theorem using the something else; if that works, you've completed a proof by cases. Third, if none of that works, you can go ahead and submit or publish the theorem you've proven with the extra hypothesis: it may still be interesting and nontrivial, and maybe someone else will be able to improve it further.

In this case, what extra assumption might be useful to prove that $n^2 + n$ is even? One idea that might occur to us is to assume that n is even, since in

that case n^2 is also even, and therefore so is $n^2 + n$. There's no obvious way to weaken this hypothesis, but we do know a general fact about evenness: every integer is either even or odd! So what happens if n is odd? Well, then n^2 is also odd, so $n^2 + n$ is the sum of two odd numbers and therefore even!

Graphical Proof of 2.6.



Algebra 1: Since n is even, n^2 is even. And since n^2 is even and n is even, n^2+n is even.
Algebra 2: Since n is odd, n^2 is odd. And since n^2 is odd and n is odd, $n^2 + n$ is even. □

It's admittedly a bit of a stretch to call these odd-even calculations "algebra". In fact, we haven't even defined what "even" and "odd" really mean! We'll come back to this in section 3.7. For now, we'll just assume the following basic facts that you are probably familiar with:

- If a and b are both even, or both odd, then $a + b$ and $a - b$ are both even.
- If one of a and b is even and the other is odd, then $a + b$ and $a - b$ are both odd.
- If at least one of a and b is even, then ab is even.
- If a and b are both odd, then ab is odd.

Each of these facts can be applied, like a rule of algebra for manipulating equations, to deduce new even-odd facts from previously known ones.

The English proof of Theorem 2.6 is, again, unsurprising.

English Proof of 2.6. Since n is either even or odd, there are two cases.
Case 1: Assume n is even. Therefore, n^2 is even. And since n^2 is even and n is even, $n^2 + n$ is even.
Case 2: Assume n is odd. Therefore, n^2 is odd. And since n^2 is odd and n is odd, $n^2 + n$ is even. □

Another way we can introduce a proof by cases is if we have a variable that belongs to a collection we know is finite. In that case, we can list all the elements of the collection, and it must be the case that that variable is equal to one of them. For instance, since $\mathbb{Z}_{/3}$ has only three elements $[0]_3$, $[1]_3$, and $[2]_3$, if we have a variable $x \in \mathbb{Z}_{/3}$ we know $(x = [0]_3) \vee (x = [1]_3) \vee (x = [2]_3)$, and then we can *use "or"* on that. This is known as *proof by exhaustion*; here's an example.

Theorem 2.7. *Suppose* $x \in \mathbb{Z}_{/3}$. *Then* $x^3 + [2]_3 \cdot x = [0]_3$.

(Most mathematicians would write this conclusion as $x^3 + 2x = 0$: the assumption $x \in \mathbb{Z}_{/3}$ has made it clear we're working in $\mathbb{Z}_{/3}$, so there should be no danger of confusion. However, I'll stick to the more explicit version.)

An "or" statement with three parts (called "disjuncts") can be used just like one with two, resulting in three cases instead of two. (See Exercises 2.5.4 and 2.5.5 and the comment afterwards). But this gets tedious to represent graphically, so I'll just give the English proof.

English Proof of 2.7. Since $x \in \mathbb{Z}_{/3}$ there are three cases.
Case 1: Assume $x = [0]_3$. Then

$$\begin{aligned} x^3 + [2]_3 \cdot x &= ([0]_3)^3 + [2]_3 \cdot [0]_3 \\ &= [0^3]_3 + [2 \cdot 0]_3 \\ &= [0]_3 + [0]_3 \\ &= [0]_3. \end{aligned}$$

Case 2: Assume $x = [1]_3$. Then

$$\begin{aligned} x^3 + [2]_3 \cdot x &= ([1]_3)^3 + [2]_3 \cdot [1]_3 \\ &= [1^3]_3 + [2 \cdot 1]_3 \\ &= [1]_3 + [2]_3 \\ &= [3]_3 \\ &= [0]_3. \end{aligned}$$

Case 3: Assume $x = [2]_3$. Then

$$\begin{aligned} x^3 + [2]_3 \cdot x &= ([2]_3)^3 + [2]_3 \cdot [2]_3 \\ &= [2^3]_3 + [2 \cdot 2]_3 \\ &= [8]_3 + [4]_3 \\ &= [12]_3 \\ &= [0]_3. \end{aligned}$$

□

A proof by exhaustion is the one situation in which, contra the discussion in section 1.2, a collection of examples *can* be a proof: if there are only finitely many *possible* examples, and you check them all.

Proofs by cases, including proofs by exhaustion, have gotten much more powerful with the advent of computers that can be programmed to check immense numbers of cases. An early example of this is that in 1976, Kenneth Appel and Wolfgang Haken used a computer to settle a famous conjecture known as the *Four Color Theorem* (every map can be colored using only four colors so that no two adjacent regions share a color) by checking 1,834 different cases. This sparked a lively debate about whether this was a valid proof, which was only fully resolved in 2005 when Benjamin Werner and Georges Gonthier used a

computer proof assistant to fully verify it. The program they used, called Coq, is related to Olorin in roughly the way that Python is related to Scratch. (In fact, a different program like Coq, called Narya, is used by Olorin "under the hood" to check the correctness of your proofs.)

Now it's your turn. We had a lot of new proof guidance in this section (Proof Guidances 3, 4, 5 and 6), so make sure you remember and apply it all.

Exercises

Exercise 2.6.1. Suppose $x, y \in \mathbb{R}$ and $(x^2 = y^2) \vee (x^3 = y^3)$. Prove $|x| = |y|$.

Exercise 2.6.2. Suppose $x, y \in \mathbb{R}$ and $(x = y + 1) \vee (2x = 3 - y)$. Prove $2x^2 - xy - y^2 = 5x - 2y - 3$.

Exercise 2.6.3. Suppose $x, y, z \in \mathbb{R}$ and $(x = z) \vee (y = z)$. Prove $z^2 + xy = xz + yz$.

Exercise 2.6.4. Suppose $x \in \mathbb{Z}$ and $x^2 = x$. Prove $(x = 0) \vee (x = 1)$.

Exercise 2.6.5. Suppose $x \in \mathbb{C}$ and $x^2 + 1 = 0$. Prove $(x = i) \vee (x = -i)$.

Exercise 2.6.6. Suppose $x \in \mathbb{R}$ and either $x^2 - x - 2 = 0$ or $x^2 - 1 = 0$. Prove either $x = 2$ or $x = -1$ or $x = 1$.

Exercise 2.6.7. Suppose $x \in \mathbb{C}$ and $x^2 + 2i = 0$. Prove $(x = 1+i) \vee (x = -1-i)$.

Note that the absolute value can be defined for any ordered number system with the same piecewise formula:

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0. \end{cases}$$

In particular, this applies to the surreal numbers $\mathbb{S}$. The absolute value of a surreal number is then another surreal number, e.g. $|-\omega| = \omega$.

Exercise 2.6.8. Suppose $x, y \in \mathbb{S}$. Prove $|xy| = |x| \cdot |y|$.

Exercise 2.6.9. Suppose $x, y \in \mathbb{S}$. Prove $|x + y| \leq |x| + |y|$.

Exercise 2.6.10. Suppose $x, y \in \mathbb{S}$. Prove $|x - y| \geq |x| - |y|$.

Exercise 2.6.11. Suppose $n \in \mathbb{Z}$. Prove $n^2 + 3n + 1$ is odd.

Exercise 2.6.12. Suppose $n \in \mathbb{Z}$. Prove $n^3 - 3n^2 + 2n$ is even.

Exercise 2.6.13. Suppose $n \in \mathbb{Z}$. Prove $n^3 + 2n^2 - n + 1$ is odd.

Exercise 2.6.14. Suppose $n \in \mathbb{Z}$. Prove $n^3 - 3n$ is even.

Exercise 2.6.15. Suppose $x \in \mathbb{Z}_{/3}$. Prove $x^4 - x^3 + [2]_3 \cdot x^2 + x = [0]_3$.

Exercise 2.6.16. Suppose $x \in \mathbb{Z}_{/4}$. Prove $[2]_4 \cdot x^2 = [2]_4 \cdot x$.

Exercise 2.6.17. Suppose $x \in \mathbb{Z}_{/6}$. Prove $x^3 - [3]_6 \cdot x^2 + [2]_6 \cdot x = 0$.

Exercise 2.6.18. Suppose $x \in \mathbb{Z}_{/6}$. Prove $x^3 = x$.

Exercise 2.6.19. Suppose $x \in \mathbb{Z}_{/6}$. Prove $x^4 + [2]_6 \cdot x^2 = x^3 + [2]_6 \cdot x$.

2.7 If-then ($\Rightarrow$)

If P and Q are statements, we write $P \Rightarrow Q$ for the truth value of the statement "if P, then Q".[12] The operator $\Rightarrow$ is called *implication*. The statement $P \Rightarrow Q$ can also be pronounced as "P implies Q", but *not* (contrary to the instincts of many students) simply "P then Q". To establish the truth table for this, I like to consider the following puzzle.

> Connor the Con Man has a deck of cards, each of which has a letter on one side and a digit on the other. He puts four cards from this deck on the table:
>
> 
>
>
> "If any of these four cards has a vowel on one side, then it has an even number on the other side," says Connor. Which of the cards do you need to turn over to check whether he's telling the truth?

In answer to some common questions:

- A con man is "one who cheats or tricks someone by gaining their trust and persuading them to believe something that is not true". The point is, you have no particular reason to trust what Connor says.
- You have inspected the entire deck beforehand and observed Connor closely, and you're certain that it is true that each card on the table has a letter on one side and a digit on the other. There's no room for doubt on that score.
- Another way to phrase the question is: for which of the cards, if you turned it over, would you have a chance of catching Connor in a lie?
- It's a separate question for each card: do you have to turn this card over? There might be one card that you have to turn over, or two, or three, or four, or none.

Please stop at this point and decide what you think the answer is.

[12] Besides $P \Rightarrow Q$, another common notation for implication is $P \to Q$. I've avoided that in these notes because of the potential for confusion with the notation $f : A \to B$ that means f is a function from A to B (although, formally speaking, the two are actually instances of the same thing; look up the "Curry-Howard correspondence").

A less common notation, used more by philosophers, is $P \supset Q$. I eschew that because it looks like the reverse of the "subset" relation $\subset$, whereas in fact implication is more like the subset relation itself than like its reverse.

No, really, I mean it. Please stop and decide what you think the answer is before you read on.

When I do this activity in class, I ask the students to vote, for each card, whether it needs to be turned over or not. Then I try to have them have pair up with someone who disagrees with them and discuss, then vote again. If your instructor wants to do the same thing without having you "spoiled", they may have deleted the answer from these notes until after the relevant class period.

Assuming we're past that, generally the results are:

- Nearly everyone agrees that we need to turn over the $\boxed{\text{E}}$, since if it has an odd number on the other side, Connor is lying about that card. This is correct.
- Most people usually agree, especially after discussion, that we don't need to turn over the $\boxed{\text{K}}$, since Connor isn't making any claim about that card. He only asserted that *if* a card has a vowel on one side it has an even number on the other side; he *didn't say anything* about cards with a consonant on their letter-side, so by turning over a card of that sort we can't catch him in a lie.
- There's more disagreement about the $\boxed{7}$, but usually someone points out that if we turn over the $\boxed{7}$ and find a vowel, then we've discovered that Connor *was* making a claim about that card, *and* the claim is false. Thus, we do need to turn over that card, because we have a chance of catching him in a lie.
- Finally, the $\boxed{4}$ card is usually the trickiest. It's hard to escape the sense of needing to turn it over, probably because "even numbers" appeared in Connor's claim. But consider: on one hand, if we turn over the $\boxed{4}$ and find a consonant, then Connor wasn't making any claim about that card. But on the other hand, if we turn over the $\boxed{4}$ and find a vowel, then we've discovered that Connor was making a claim about that card, *and the claim is true*. In neither case do we have a chance of catching him in a lie; so we *don't* need to turn over the $\boxed{4}$.

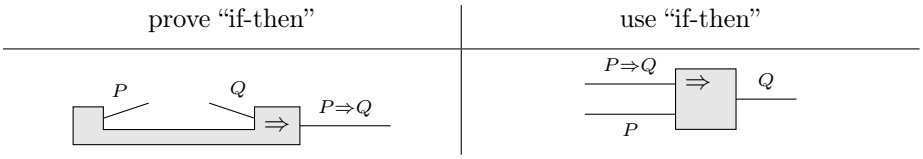
This activity helps us fill out the truth table for $\Rightarrow$. Just as the only way we can catch Connor in a lie is if we find a card that has a vowel on one side *and* an odd number on the other side, the only way $P \Rightarrow Q$ can be false is if the "if" part, P, is true, *and* the "then" part, Q, is false. So the truth table is:

P	Q	$P \Rightarrow Q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$

In particular, if P is false then $P \Rightarrow Q$ is true no matter what Q is, and similarly if Q is true then $P \Rightarrow Q$ is true no matter what P is. Put differently, $P \Rightarrow Q$ only asserts that Q is true *in the case when P is true*; in the case when P is false it makes no claim and hence is true (we call this being *vacuously true*: true because it makes no falsifiable claim). This justifies the proof rules for $\Rightarrow$:

- The rule to *prove "if-then"* says that to prove $P \Rightarrow Q$, we can assume, hypothetically, that P is true, and prove that Q is true under that assumption. This rule is sometimes known as *direct proof.*[13]
- The rule to *use "if-then"* says that if we know $P \Rightarrow Q$, and we also know (or can prove) P, we can deduce Q. This rule is so important it has a fancy Latin name: *modus ponens.*

The graphical representations of these rules are as follows:

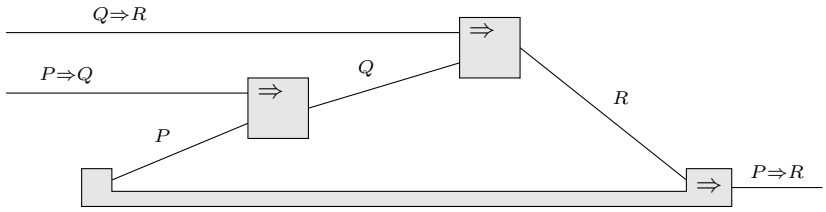


Note that the *prove "if-then"* block has a "bracket" like the two in the *use "or"* block. It has a similar meaning: the hypothetical assumption wire P can only be used to derive the subgoal Q; it cannot "escape" to be used for anything to the right of the *prove "if-then"* block. But, as before, givens that we already have before starting the *prove "if-then"* can "come inside".

Here is an example, showing that implication, like equality and inequality, is *transitive*:

Theorem 2.8 (Olorin 1-3-4)**.** *Suppose $P \Rightarrow Q$ and $Q \Rightarrow R$. Then $P \Rightarrow R$.*

Graphical Proof of 2.8.



□

That is, to prove $P \Rightarrow R$ we assume, hypothetically, that P, and must prove R under that assumption. From P and the given $P \Rightarrow Q$, we deduce Q; then from Q and the given $Q \Rightarrow R$ we deduce R; which was what we wanted in our sub-proof. Note that the givens $P \Rightarrow Q$ and $Q \Rightarrow R$ "come inside" the if-then bracket from the left.

To write this in English, it suffices to recall PEP 3 from section 2.5 for making assumptions. We may also add a concluding sentence at the end, although this is not strictly necessary.

English Proof of 2.8. Assume P. Since $P \Rightarrow Q$ and P, we have Q. And since $Q \Rightarrow R$ and R, we have R. Therefore, $P \Rightarrow R$. □

Now it's your turn. Don't forget Proof Guidances 1 and 2.

[13] Although some textbooks use that term so broadly as to make it almost meaningless, for instance including in it a silent use of the rules for "there exists" that we will discuss in chapter 3.

Exercises

As before, for each of these exercises you should give *both* a graphical proof *and* an English proof. This section pertains to Stages 1-5, 1-6, 1-7, and 2-1 of Olorin.

Exercise 2.7.1 (Olorin 1-3-2)**.** Prove $P \Rightarrow P$.

Exercise 2.7.2 (Olorin 1-3-3)**.** Prove $P \Rightarrow (Q \Rightarrow P)$. (This is known as *verum sequitur ad quodlibet*.)

Exercise 2.7.3 (Olorin 1-3-5)**.** Suppose $P \Rightarrow (P \Rightarrow Q)$. Prove $P \Rightarrow Q$.

Exercise 2.7.4 (Olorin 1-3-6)**.** Suppose $P \Rightarrow Q$ and $P \Rightarrow R$ and $Q \Rightarrow S$ and $R \Rightarrow S$. Prove $P \Rightarrow S$.

Exercise 2.7.5 (Olorin 1-3-7)**.** Suppose $P \Rightarrow (Q \Rightarrow R)$. Prove $Q \Rightarrow (P \Rightarrow R)$.

Exercise 2.7.6 (Olorin 1-3-8)**.** Suppose $P \Rightarrow Q$. Prove $(Q \Rightarrow R) \Rightarrow (P \Rightarrow R)$.

Exercise 2.7.7 (Olorin 1-3-9)**.** Suppose $P \Rightarrow Q$. Prove $(R \Rightarrow P) \Rightarrow (R \Rightarrow Q)$.

Exercise 2.7.8 (Olorin 1-4-1)**.** Suppose $P \Rightarrow R$ and $Q \Rightarrow R$. Prove $(P \wedge Q) \Rightarrow R$.

Exercise 2.7.9 (Olorin 1-4-2)**.** Suppose $(P \Rightarrow Q) \wedge (P \Rightarrow R)$.
Prove $P \Rightarrow (Q \wedge R)$.

Exercise 2.7.10 (Olorin 1-4-3)**.** Suppose $P \Rightarrow (Q \wedge R)$.
Prove $(P \Rightarrow Q) \wedge (P \Rightarrow R)$.

Exercise 2.7.11 (Olorin 1-4-4)**.** Suppose $P \Rightarrow (Q \Rightarrow R)$. Prove $(P \wedge Q) \Rightarrow R$.

Exercise 2.7.12 (Olorin 1-4-5)**.** Suppose $(P \wedge Q) \Rightarrow R$. Prove $P \Rightarrow (Q \Rightarrow R)$.

Exercise 2.7.13 (Olorin 1-4-6)**.** Suppose $P \Rightarrow Q$ and $R \Rightarrow S$.
Prove $(P \wedge R) \Rightarrow (Q \wedge S)$.

Exercise 2.7.14 (Olorin 1-7-1)**.** Suppose $(P \Rightarrow R) \wedge (Q \Rightarrow R)$.
Prove $(P \vee Q) \Rightarrow R$.

Exercise 2.7.15 (Olorin 1-7-2)**.** Suppose $(P \vee Q) \Rightarrow R$.
Prove $(P \Rightarrow R) \wedge (Q \Rightarrow R)$.

Exercise 2.7.16 (Olorin 1-7-3)**.** Suppose $(P \Rightarrow Q) \vee (P \Rightarrow R)$.
Prove $(P \wedge Q) \Rightarrow R$.

Exercise 2.7.17 (Olorin 1-7-4)**.** Suppose $(P \Rightarrow Q) \vee (P \Rightarrow R)$.
Prove $P \Rightarrow (Q \vee R)$.

Exercise 2.7.18 (Olorin 1-7-5)**.** Suppose $P \Rightarrow Q$ and $R \Rightarrow S$.
Prove $(P \vee R) \Rightarrow (Q \vee S)$.

Congratulations! You're done with Olorin's "Proposition World". On to "Advanced Proposition World".

Exercise 2.7.19 (Olorin 2-1-1)**.** Suppose $P \Rightarrow Q$ and $P \Rightarrow (Q \Rightarrow R)$. Prove $P \Rightarrow R$.

Exercise 2.7.20 (Olorin 2-1-2)**.** Suppose $P \Rightarrow (P \Rightarrow Q)$ and $(P \Rightarrow Q) \Rightarrow P$. Prove Q.

Exercise 2.7.21 (Olorin 2-1-3)**.** Suppose $(P \Rightarrow Q) \Rightarrow R$ and $(Q \Rightarrow R) \Rightarrow P$ and $(R \Rightarrow P) \Rightarrow Q$. Prove $P \wedge (Q \wedge R)$.

Exercise 2.7.22 (Olorin 2-1-4)**.** Suppose $P \Rightarrow (Q \wedge R)$ and $(P \wedge Q) \Rightarrow S$ and $(P \Rightarrow S) \Rightarrow (Q \Rightarrow P)$. Prove $Q \Rightarrow (R \wedge S)$.

Exercise 2.7.23 (Olorin 2-1-5)**.** Prove $((P \vee (P \Rightarrow Q)) \Rightarrow Q) \Rightarrow Q$.

2.8 ⇒ with algebra

Nope, sorry, I got nothin'.

Seriously: the only content of this section is to observe that you are a *bourgeois gentilhomme* again. Our rewritten Theorem 1.1 from section 2.4:

Theorem 1.1. *Suppose $x, y, z \in \mathbb{R}$, and that $(x + 1 = y) \wedge (x - 1 = z)$. Then $x^2 = yz + 1$.*

can be rewritten yet again to use ⇒:

Theorem 1.1 (with ⇒)**.** *Suppose $x, y, z \in \mathbb{R}$. Then*

$$\big((x + 1 = y) \wedge (x - 1 = z)\big) \Rightarrow (x^2 = yz + 1).$$

The proof of this would be exactly like the proof of the previous version, except that it would start with a *prove "if-then"* rule, assuming $(x + 1 = y) \wedge (x - 1 = z)$ and using it to prove $x^2 = yz + 1$. Since all theorems have some list of hypotheses[14] and a conclusion, they can all be rewritten as statements of the form $(P_1 \wedge \cdots \wedge P_n) \Rightarrow Q$. In practice, we don't usually mention the outer $\wedge$ and $\Rightarrow$ rules that destructure a theorem statement of this form, but they are officially there nonetheless.

Now I hear you ask, what about the "Suppose $x, y, z \in \mathbb{R}$"? Put a pin in that question until chapter 3, specifically section 3.9.

[14] By the power of triviality, the empty list is still a list, so this statement is true even if there are no hypotheses. But what is $P_1 \wedge \cdots \wedge P_n$ in the case $n = 0$? Come back to this footnote after you read section 2.10 and see whether you can guess the answer.

2.9 If and only if ($\Leftrightarrow$)

It is *extremely important* that unlike addition, multiplication, $\wedge$, and $\vee$, the operator $\Rightarrow$ is *not commutative*: $P \Rightarrow Q$ is different from $Q \Rightarrow P$. I find it helpful to remember this using an everyday example. One very traditional example is to let P be "it rained last night" and let Q be "the grass is wet". We can suppose that $P \Rightarrow Q$ is true: rain makes the grass wet. But $Q \Rightarrow P$ claims that "if the grass is wet, then it rained last night", which is generally not true: perhaps the sprinklers ran this morning.

The statement $Q \Rightarrow P$ is called the *converse* of the statement $P \Rightarrow Q$. The principle to remember is that *the converse of a true statement might also be true, but it might instead be false; there is no general rule.* We will discuss the converse again, along with two other related statements called the "inverse" and the "contrapositive", in section 4.2.

However, it does often happen that we encounter true implications whose converse *is* also true. This is common enough that we introduce a special notation[15] and terminology for it:

$$(P \Leftrightarrow Q) \;=\; (P \Rightarrow Q) \wedge (Q \Rightarrow P).$$

The operator $\Leftrightarrow$ is thus a *derived* operator, defined in terms of $\Rightarrow$ and $\wedge$. We can deduce from this definition a truth table for it:

P	Q	$P \Leftrightarrow Q$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\top$

Thus $P \Leftrightarrow Q$ is true precisely when P and Q have the same truth value (both true or both false).

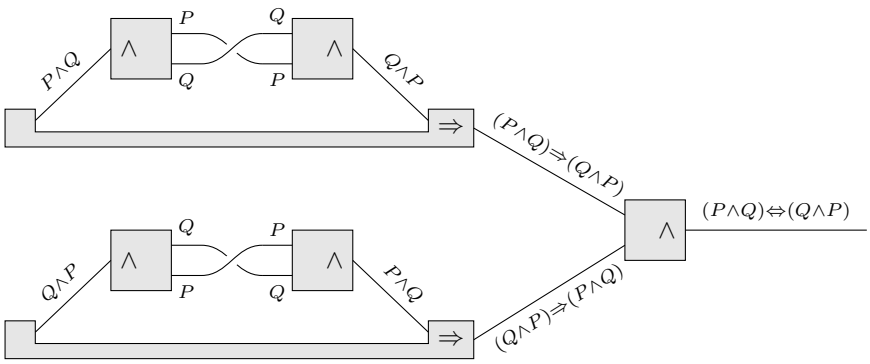
We pronounce $P \Leftrightarrow Q$ as "P if and only if Q". In this phrase, the "if" refers to the $Q \Rightarrow P$ direction and the "only if" refers to the $P \Rightarrow Q$ direction, but I always have to stop and think to remember that, so don't worry about memorizing it. Another equivalent phrase is "P is necessary and sufficient for Q"; here "necessary" refers to $Q \Rightarrow P$ and "sufficient" to $P \Rightarrow Q$, but again there's no real need to remember which is which.

Since $\Leftrightarrow$ is a derived operator, it does not really have proof rules of its own: to use it in proofs, we simply combine the appropriate proof rules for $\Rightarrow$ and $\wedge$. For example, here is an "internalized" version of the commutativity of $\wedge$:

Theorem 2.9 (Olorin 2-2-5)**.** $(P \wedge Q) \Leftrightarrow (Q \wedge P)$.

[15] People who write $\Rightarrow$ as $\rightarrow$ generally write $\Leftrightarrow$ as $\leftrightarrow$. I have no idea what most people who write $\Rightarrow$ as $\supset$ do, although Wikipedia informs me that at least one person has written it as $\supset\subset$ and at least one person has written it as $\subset\supset$.

Graphical Proof of 2.9.



□

When proving an ⇔ in English, it is customary to label the two proofs necessitated by the ∧ with "⇒" and "⇐". (Some authors label them with "if" and "only if", but since I have trouble remembering which of those is which, I generally eschew that.) Thus, for example:

English Proof of 2.9.
[⇒] Suppose $P \wedge Q$. Therefore, we have P and we also have Q. But since Q and P, we have $Q \wedge P$.
[⇐] Suppose $Q \wedge P$. Therefore, we have Q and we also have P. But since P and Q, we have $P \wedge Q$. □

At this point you might want to go back and re-read the discussion about simplifying and manipulating equations at the end of section 1.4.

Exercises

As always, give both a graphical proof and an English proof. This section pertains to Stages 2-2 and 2-3 of Olorin.

Exercise 2.9.1 (Olorin 2-2-1)**.** Prove $P \Leftrightarrow P$.

Exercise 2.9.2 (Olorin 2-2-2)**.** Prove $P \Leftrightarrow (P \wedge P)$.

Exercise 2.9.3 (Olorin 2-2-3)**.** Suppose $P \Leftrightarrow Q$. Prove $Q \Leftrightarrow P$.

Exercise 2.9.4 (Olorin 2-2-4)**.** Suppose $P \Leftrightarrow Q$ and $Q \Leftrightarrow R$. Prove $P \Leftrightarrow R$.

Exercise 2.9.5 (Olorin 2-2-6)**.** Suppose $P \Leftrightarrow Q$. Prove $(P \wedge R) \Leftrightarrow (Q \wedge R)$.

Exercise 2.9.6 (Olorin 2-3-1)**.** Prove $P \Leftrightarrow (P \vee P)$.

Exercise 2.9.7 (Olorin 2-3-2)**.** Prove $(P \vee Q) \Leftrightarrow (Q \vee P)$.

Exercise 2.9.8 (Olorin 2-3-3)**.** Suppose $P \Leftrightarrow Q$. Prove $(P \vee R) \Leftrightarrow (Q \vee R)$.

Exercise 2.9.9 (Olorin 2-3-4)**.** Suppose $P \Leftrightarrow Q$. Prove $(P \Rightarrow R) \Leftrightarrow (Q \Rightarrow R)$.

Exercise 2.9.10 (Olorin 2-3-5)**.** Suppose $P \Leftrightarrow Q$. Prove $(R \Rightarrow P) \Leftrightarrow (R \Rightarrow Q)$.

Now would be a good time to start going back to solve the Proposition World exercises from sections 2.3 and 2.5 on Olorin's Adept difficulty setting.

2.10 Truth value constants ($\top$ and $\bot$)

Continuing our analogy between addition/multiplication and logical and/or, we can ask whether $\wedge$ and $\vee$ have *identity elements*, like 0 for addition and 1 for multiplication:

$$0 + x = x \qquad 1 \cdot x = x$$

or *absorbing elements*, like 0 for multiplication:

$$0 \cdot x = 0.$$

Inspecting their definitions with truth tables, we see that $\wedge$ has $\top$ as an identity element and $\bot$ as an absorbing element:

$$\top \wedge P = P \qquad \bot \wedge P = \bot$$

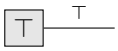
while $\vee$ has $\bot$ as an identity element and $\top$ as an absorbing element:

$$\bot \vee P = P \qquad \top \vee P = \top.$$

We can also *prove* these equivalences if we give *proof rules* for the truth value constants $\top$ and $\bot$. Here are the rules for $\top$:

- The *prove "true"* rule says that we can always prove $\top$ without doing any work.
- There is no *use "true"* rule (since $\top$ is always true, knowing that it is true doesn't give us any information).

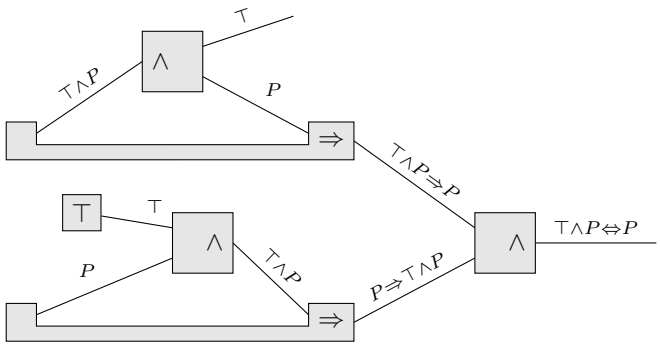
Graphically, the *prove "true"* rule is very simple:



With this, we can prove that $\top$ is an identity element for $\wedge$:

Theorem 2.10 (Olorin 2-4-3)**.** $(\top \wedge P) \Leftrightarrow P$.

Graphical Proof of 2.10.



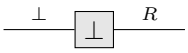
□

English Proof of 2.10.
⇒: Suppose $\top \wedge P$. Therefore, P.
⇐: Suppose P. We also have $\top$. Since $\top$ and P, we have $\top \wedge P$. □

The rules for $\bot$ are a little more complicated.

- There is no *prove "false"* rule (since $\bot$ is never true, there is no way to prove it).
- The *use "false"* rule says that if we have assumed $\bot$, we can deduce *any* other statement R.

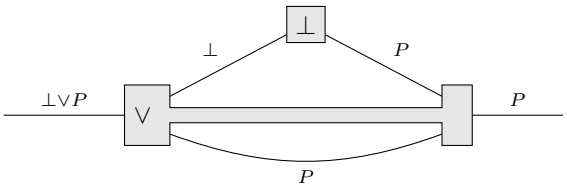
Graphically, the *use "false"* rule looks like this:



The idea behind this rule is that since $\bot$ is *never* true, an assumption of $\bot$ is impossible. Therefore, when we divide into cases, if one of the cases assumes $\bot$, then that case can't actually happen, so we shouldn't have to give a proof in that case. But formally, for a proof by cases to be valid, both cases must yield the same result; so in the *use "false"* rule we artificially stipulate that we can "deduce" any necessary goal. This is easiest to explain by an example.

Theorem 2.11. *Suppose $\bot \vee P$. Then P.*

Graphical Proof of 2.11.



□

English Proof of 2.11. Since $\bot \vee P$, we can divide into cases.
Case 1: Assume $\bot$. This is impossible.
Case 2: Assume P. Therefore, P. □

Exercises

As always, give both a graphical proof and an English proof. This section pertains to Stages 2-4 and 2-5 of Olorin.

Exercise 2.10.1 (Olorin 2-4-2)**.** Prove $(P \Rightarrow P) \Leftrightarrow \top$.

Exercise 2.10.2 (Olorin 2-4-4)**.** Prove $(\top \vee P) \Leftrightarrow \top$.

Exercise 2.10.3 (Olorin 2-4-5)**.** Prove $(\top \Rightarrow P) \Leftrightarrow P$.

Exercise 2.10.4 (Olorin 2-4-6)**.** Prove $(P \Rightarrow \top) \Leftrightarrow \top$.

Exercise 2.10.5 (Olorin 2-4-7)**.** Prove $(\top \Leftrightarrow P) \Leftrightarrow P$.

Exercise 2.10.6 (Olorin 2-5-2)**.** Prove $(\bot \wedge P) \Leftrightarrow \bot$.

Exercise 2.10.7 (Olorin 2-5-3)**.** Prove $(\bot \vee P) \Leftrightarrow P$.

Exercise 2.10.8 (Olorin 2-5-4)**.** Prove $(\bot \Rightarrow P) \Leftrightarrow \top$.

Congratulations! You're done with Olorin's "Advanced Proposition World". On to "Quantifier World"! Now would also be a good time to make sure you can solve all the "Proposition World" levels on the Adept difficulty setting.

How much to write in a proof? Whether and when to use symbols? Which proof rules to not mention?

Chapter 3

Quantifier proofs

So far, in all our proofs, the relevant variables have been fixed at the beginning in the statement of the theorem and didn't change over the course of the proof. But this restriction is very limiting: we want to be able to prove and use statements asserting that objects with certain properties *exist*, and describing how the values of some variables depend on others. In this chapter we remove that restriction by introducing *quantifiers* and their proof rules. First, however, we need a detour to discuss variables.

3.1 Free and bound variables

After a course in algebra, you may think you know what a variable is. But variables are really one of the most subtle things in mathematics. We won't delve into all their nooks and crannies here, but we do need to recognize that there are actually *two kinds* of variables.

The ordinary variables that we have been using in these notes so far are called *free* variables. They stand for some *unknown* or *unspecified* object (such as a number), and therefore they can be substituted by any object without damaging the correctness of any calculation or proof. For example, the expression $x^2 - 3$ does not evaluate to a specific number until we choose a value for x; but once we choose, say, $x = 7$, then $x^2 - 3 = 7^2 - 3 = 46$. Similarly, we have the algebraic identity $(x + 1)^2 = x^2 + 2x + 1$, and this remains true when we substitute some concrete number like 5 for x: we have $(5+1)^2 = 6^2 = 36$, while $5^2 + 2 \cdot 5 + 1 = 25 + 10 + 1 = 36$ also.

We can also substitute another expression for a free variable, perhaps an expression containing other free variables. As I reminded you in section 1.1, in this case it's important to put parentheses around the expression being substituted, so that it *as a whole* is treated like the variable. For instance, if we substitute $y + 1$ for the free variable x in $x^2 - 3$, we get $(y + 1)^2 - 3$, not $y + 1^2 - 3$.

In principle, we can even substitute an expression containing the *same* free variable. For instance, substituting $2x$ for the free variable x in $x^2 - 3$ yields

$(2x)^2 - 3$. However, if you ever find yourself doing this, it's usually a sign that you made some ill-advised choices of variable names; usually it's less confusing to ensure that variable names occurring in different contexts are distinct.

What we *cannot* do, ever, is substitute for *some* occurrences of a free variable but not *all* of them. For instance, we cannot substitute 3 only for the first x in $x^2 + 2x + 1$ to get $3^2 + 2x + 1$. Whenever you substitute a value or expression for a variable, make sure you substitute for *every* occurrence of that variable in the expression.

It should also be emphasized that the value of a free variable is not *completely* unspecified: every variable comes along with a *type*, which is the collection of objects that it could represent. In ordinary algebra, the type of most variables is $\mathbb{R}$, the real numbers. However, when doing more advanced mathematics, we need to be more careful to specify the type of each variable, as discussed for theorem statements in section 1.5. Only objects belonging to the correct type can be substituted for a variable.

The other kind of variable is called a *bound* variable. It is perhaps easiest to explain this with an example from calculus. Consider an integral[1]

$$\int_1^3 x^2 \, dx$$

The variable x appearing here is *not* a free variable like the x in $x^2 - 3$. For one thing, the value of the integral (namely $\frac{26}{3}$) does not depend on "the value x takes".[2] More importantly, we cannot substitute a value for x in this expression and get something meaningful:

¿¿¿ $\int_1^3 5^2 \, d5$???

Even if we ignore the highly suspicious-looking "$d5$", now 5^2 is a *constant* 25, so its integral from 1 to 3 is just $25(3-1) = 50$, quite different from the original value $\frac{26}{3}$.

[1] By which I mean, of course, a *definite* integral. The terminology and notation of the so-called "indefinite integral" is an atrocious abomination that ought to be taken out and shot. Only consider (after reading the rest of this section): we cannot substitute a value for x in $\int x^2 \, dx = \frac{1}{3}x^3 + C$ to get a meaningful statement like "$\int 5^2 d5 = \frac{1}{3}5^3 + C$", so x is not a free variable therein; but neither is x a bound variable, because $\int y^2 \, dy = \frac{1}{3}y^3 + C$ is *different* from $\int x^2 \, dx = \frac{1}{3}x^3 + C$, since x and y are certainly free in the expressions $\frac{1}{3}x^3 + C$ and $\frac{1}{3}y^3 + C$, which are therefore different. And don't get me started on the ubiquitous "$+C$", which sweeps all sorts of mendacity under the rug: if the "indefinite integral" is supposed to represent *all* the possible antiderivatives of a function, as we are often told, then the equation $\int \frac{1}{x} \, dx = \ln|x| + C$ gleefully parroted by calculus books and the Internet is a barefaced lie, since $F(x) = \begin{cases} \ln(x) + 3 & \text{if } x > 0 \\ \ln(-x) + 4 & \text{if } x < 0 \end{cases}$ is another perfectly good antiderivative. And the general antiderivative of $\tan(x)$ involves *infinitely* many arbitrary constants.

[2] This is *not* entirely dispositive, however, since the value of $x - x$, namely 0, also does not depend on x, but x is still a free variable in it. Whether a variable is free is a condition on "syntax", i.e. the way an expression is written on the page, rather than the "value" of that expression. But if the value of an expression *does* depend on the value of some variable, then that variable must be free.

Instead, the variable x in the above integral is what's called a *bound* variable. A bound variable cannot be substituted for; it is not a "parameter" of an expression or statement. Instead, it takes on *multiple values* in the *process* of evaluating that expression or statement; how exactly this happens is determined by the *binding operator* that controls it. The definite integral is one such binding operator, and its definition indeed evaluates the function being integrated at many different values, creating many different Riemann sums and then taking their limit.

Speaking of limits, the limit is another binding operator. In

$$\lim_{x \to 3} \frac{x^2 - 9}{x - 3}$$

we cannot meaningfully substitute a value for x: there is no sense to "the limit as 4 approaches 3." Instead, the definition of limit evaluates the function at many different values, getting closer and closer to 3 but never reaching it.

An important property of a bound variable is that it can be *renamed* without changing the meaning of the expression it appears in, as long as the new name does not conflict with any other existing variable (free or bound) in that expression. For example,

$$\int_1^3 x^2 \, dx = \int_1^3 y^2 \, dy = \int_1^3 t^2 \, dt = \int_1^3 u^2 \, du$$

and so on; the name of the variable doesn't matter. The non-conflict condition is important, though: if we consider an expression with a parameter (free variable) such as

$$\int_1^3 x^a \, dx$$

we can't rename x to a, since

$$\int_1^3 a^a \, da$$

would mean something quite different.

A bound variable always has a *scope*: the region of the expression in which it is bound. In the case of the integral, the scope of the bound variable is the region between the $\int$ and the d (but *not* including the upper and lower limits of integration). The occurrence of the variable *after* the d is not in this scope, but still refers to the bound variable: it is a sort of syntactic "label" that indicates the name of the variable being bound.[3]

In the case of the limit, the scope of the bound variable is the function written after the lim (but not including the value that the variable is approaching). Here the "label" is the occurrence of the variable in the subscript such as $x \to 3$.

[3]However, in the mathematical subject called *differential geometry*, there is a reintepretation of the integral according to which the whole expression "$f(x)\,dx$" is an object, called a *differential 1-form*, that is what gets integrated. If you've taken multivariable calculus and wondered whether there is something deeper underlying the similarities between Green's theorem, Stokes' theorem, the divergence theorem, and the fundamental theorem of calculus, you need to learn about differential forms.

Occurrences of a variable outside its bound scope are not bound by the same binder. They might be bound by another binder; for instance it is common to write

$$\int_a^b (f(x) + g(x))\,dx = \int_a^b f(x)\,dx + \int_a^b g(x)\,dx$$

in which the three integrals each bind a variable x, one with scope $f(x) + g(x)$, one with scope $f(x)$, and one with scope $g(x)$. We could make these distinct scopes clearer by using different variables in the three cases:

$$\int_a^b (f(x) + g(x))\,dx = \int_a^b f(u)\,du + \int_a^b g(v)\,dv$$

but this would probably be more confusing than helpful.

It is also, officially, technically, possible for a bound variable to appear elsewhere in the same expression as a *free* variable. For instance,

$$x + \int_1^3 x^2\,dx$$

is a meaningful expression that equals $x + \frac{26}{3}$, with the first x being free. However, this is generally considered bad practice, due to the potential for confusion. For example, half of the Fundamental Theorem of Calculus (no one can agree on whether this is the "first" half or the "second" half) states that if f is continuous and we define a new function F by

$$F(x) = \int_a^x f(t)\,dt$$

then F is an antiderivative of f, that is $F' = f$. Since the scope of the bound variable in an integral doesn't include the upper and lower limits of integration, the definition of F could, officially, technically, also be written as

$$F(x) = \int_a^x f(x)\,dx$$

as the x in the upper limit of integration is outside the scope bound by the integral. But most people find it less confusing to use a different letter for the bound variable (recalling that a bound variable can be renamed as desired).

Speaking of function definitions, the variable x used in a definition like

$$f(x) = x^2$$

is also bound.[4] Substituting a value for it would produce $f(3) = 3^2$, which is a true *consequence* of the original function definition, but is not itself any longer

[4] This means that in the *entire* definition $F(x) = \int_a^x f(t)\,dt$, the variable x is also bound. But in the sub-expression $\int_a^x f(t)\,dt$, it is free. More generally, a bound variable behaves as a free variable in the subexpression consisting of its scope.

a function definition. And we can change the name of the variable and still be defining the same function:

$$f(t) = t^2$$

namely, *the function that squares its argument*. In a function definition, the occurrence of x on the left-hand side "$f(x)$" is the "label" naming the variable, while the bound scope is the right-hand side of the equality.

And, by the way, the name of the function is f, not $f(x)$. There's no time like the present to permanently rid yourself of the execrable habit of saying "the function $f(x)$" perpetuated by many algebra books. The name of the function is f, while $f(x)$ is the *result* of applying that function to the argument x.

The habit of calling the function "$f(x)$" is probably exacerbated by the fact that we *define* the function f by an equation such as $f(x) = x^2$ in which the left-hand side is "$f(x)$". The reason this makes sense is that a function is determined by its values, and so we can completely determine the behavior of the function f by specifying its value at a *variable* such as x, since we could substitute anything for x to find the value of f at that thing.

However, this is admittedly confusing because if f has *already* been defined, then $f(x)$ is an expression in which x is *free*, representing the value of the function f at the input x, and with the above function definition $f(x) = x^2$ would then be a true *statement* in which x is free. (This is no different than other true statements such as $f(5) = 5^2$ and $f(y-1) = (y-1)^2$; it just looks confusing because the expression the function is being applied to is a single variable.) In order to distinguish these cases, sometimes people use a different version of the $=$ sign when making a *definition* (as opposed to a *statement* about an existing definition), such as one of the following:

$$f(x) \coloneqq x^2 \qquad f(x) \stackrel{\text{def}}{=} x^2 \qquad f(x) \triangleq x^2.$$

An even less confusing way to define this same function is

$$f = \lambda x.x^2.$$

The notation "$\lambda x.$" is a binding operator called a *lambda abstraction* (λ is the lowercase Greek letter lambda), and it means "the function defined to send x to…", with x a bound variable therein. Thus, the above equation says that f (the name of the function!) is the function defined to send x to x^2. This is the same thing that $f(x) = x^2$ means, but perhaps less confusing because it clearly states that we are defining f to be something (namely, a specific function).

The lambda notation also allows us to talk about a *function* (such as $\lambda x.x^2$) without giving it a *name* (such as f). For this reason, it has made its way into many modern programming languages. For instance, in Python, $\lambda x.x^2$ is written as `lambda x: x**2`. You can then define a variable to equal it:

```
f = lambda x: x**2
```

is the same as

```
def f(x):
    return x**2
```

But you can also pass it as an argument to another function *without* giving it a name:

```
numbers = [4, -6, 2, -1]
numbers.sort(key = lambda x: x**2)
```

This will result in `numbers` being `[-1, 2, 4, -6]`, with its elements sorted in the order of their squares.

Another equivalent notation to $\lambda x.x^2$ that's sometimes used is $x \mapsto x^2$ (note the little bar at the beginning of the arrow), which has programming counterparts such as `(x) => x**2` in JavaScript. For clarity, both $\lambda x.x^2$ and $x \mapsto x^2$ should be surrounded by parentheses when they appear in more complicated expressions.

So far in this section we have considered mainly variables in *expressions* that represent numbers, but we also use variables in *statements* (propositions). For example, $n^2 \leq 5$ is a proposition that contains a free variable n, and once we substitute a particular value for this variable, like $n = 3$, we obtain the statement $3^2 \leq 5$ that has a truth value (in this case, "false").[5]

Exercises

Exercise 3.1.1. What are the bound and free variables in $\displaystyle\int_a^{3a} (x^2 + 3ay)\,dx$?

Exercise 3.1.2. What are the bound and free variables in $\displaystyle\int_x^y u^x\,du$?

Exercise 3.1.3. What are the bound and free variables in $\displaystyle\lim_{n\to\infty} \int_1^\infty x^{-n}\,dx$?

Exercise 3.1.4. Rewrite the following expression in an equivalent way so that no bound variable is ever re-used in another place as a bound or free variable.

$$\sin(x) + \int_x^{x^2} x^n\,dx - x \cdot \lim_{x\to n} \frac{x-n}{x+n}$$

Exercise 3.1.5. If we define the function f by $f(x) = x^n + 3x^2 - 2n$, what are the bound and free variables in this definition?

[5] Remember that *substituting* for a variable means *replacing* that variable everywhere with some other value or expression. It does *not* mean simply affixing an assumption of equality. In particular, the result of substituting 3 for n in $n^2 \leq 5$ is *not* the statement "if $n = 3$ then $n^2 \leq 5$". The latter statement still contains n as a free variable. If we *actually* substitute 3 for n in this latter statement, we get "if $3 = 3$ then $3^2 \leq 5$", which is false since it is an if-then statement whose "if" part $3 = 3$ is true and whose "then" part $3^2 \leq 5$ is false. But we could instead substitute 4 for n in this latter statement, obtaining "if $4 = 3$ then $4^2 \leq 5$", which is *true* since it is an if-then statement whose "if" part is false.

Exercise 3.1.6. Give four different ways to define g to be the function that cubes its argument and subtracts two.

Exercise 3.1.7. Which of the following are equal to $\int_1^a \int_1^b (2x^2 + y^2)\, dx\, dy$?

(a) $\int_1^a \int_1^b (2u^2 + v^2)\, du\, dv$

(b) $\int_1^c \int_1^d (2x^2 + y^2)\, dx\, dy$

(c) $\int_1^b \int_1^a (2x^2 + y^2)\, dx\, dy$

(d) $\int_1^a \int_1^b (2y^2 + x^2)\, dy\, dx$

(e) $\int_1^b \int_1^a (2y^2 + x^2)\, dy\, dx$

3.2 Σ and Π

A simple sort of binding operator (simpler than integrals and limits, anyway) is one that just evaluates an expression at some values and combines them all in some way. When the objects being combined are numbers, there are two natural ways to combine them: by addition or by multiplication. The resulting binding operators are traditionally denoted with Σ (a capital "sigma", the Greek letter "s" for "sum") and Π (a capital "pi", the Greek letter "p" for "product"). For example,

$$\sum_{k=1}^{5} k^2$$

means to let k take on all the integer values from 1 to 5 inclusive (that is, 1, 2, 3, 4, and 5), evaluate the expression k^2 for each of those values of k, and add them up. Thus its value is

$$\sum_{k=1}^{5} k^2 = 1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55.$$

Here the variable k is bound, with scope the expression k^2 but not the upper and lower limits 1 and 5, and the "label" is the "$k =$" on the bottom. This is called an *indexed sum* or a *summation*.

Similarly, the *indexed product*

$$\prod_{k=1}^{5} \frac{k}{k+1}$$

means to let k take on all the integer values from 1 to 5 inclusive (that is, 1, 2, 3, 4, and 5), evaluate the expression $\frac{k}{k+1}$ for each of those values of k, and multiply them together. Thus its value is

$$\begin{aligned}\prod_{k=1}^{5} \frac{k}{k+1} &= \frac{1}{1+1} \cdot \frac{2}{2+1} \cdot \frac{3}{3+1} \cdot \frac{4}{4+1} \cdot \frac{5}{5+1} \\ &= \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{3}{4} \cdot \frac{4}{5} \cdot \frac{5}{6} \\ &= \frac{1}{6}.\end{aligned}$$

Unlike for an integral (but like for a limit), there is no "delimiter" on the right marking the end of the bound scope; thus an expression like

$$\sum_{k=1}^{5} k + 1$$

is a bit ambiguous. Some mathematicians have conventions about what this means, but I think it's better to use parentheses to indicate what you mean unambiguously:

$$\begin{aligned}\sum_{k=1}^{5} (k+1) &= 2+3+4+5+6 = 20 \\ \left(\sum_{k=1}^{5} k\right) + 1 &= (1+2+3+4+5)+1 = 16.\end{aligned}$$

These operators have various properties that you can figure out if you think about what they mean, and what you know about addition and multiplication. For instance, since it doesn't matter what order you add things up in, we have

$$\begin{aligned}\sum_{k=1}^{5} (k^2 + k^3) &= (1^2+1^3) + (2^2+2^3) + (3^2+3^3) + (4^2+4^3) + (5^2+5^3) \\ &= (1^2+2^2+3^2+4^2+5^2) + (1^3+2^3+3^3+4^3+5^3) \\ &= \left(\sum_{k=1}^{5} k^2\right) + \left(\sum_{k=1}^{5} k^3\right).\end{aligned}$$

Moreover, if we add up two sequences of terms and then add the results together, we might as well have added them all up to start with:

$$\begin{aligned}\sum_{k=1}^{5} k^2 &= 1^2+2^2+3^2+4^2+5^2 \\ &= (1^2+2^2+3^2) + (4^2+5^2) \\ &= \left(\sum_{k=1}^{3} k^2\right) + \left(\sum_{k=4}^{5} k^2\right).\end{aligned}$$

More abstractly, this rule can be written as:

$$\sum_{k=1}^{n} k^2 = \left(\sum_{k=1}^{m} k^2\right) + \left(\sum_{k=m+1}^{n} k^2\right). \qquad (*)$$

where in the above example, $n = 5$ and $m = 3$. And since multiplication distributes over addition, we can say

$$\begin{aligned} 3 \cdot \sum_{k=1}^{5} k^2 &= 3 \cdot (1^2 + 2^2 + 3^2 + 4^2 + 5^2) \\ &= 3 \cdot 1^2 + 3 \cdot 2^2 + 3 \cdot 3^2 + 3 \cdot 4^2 + 3 \cdot 5^2 \\ &= \sum_{k=1}^{5} 3k^2. \end{aligned}$$

The "power of triviality" suggests we should also consider carefully the meaning of $\sum_{k=1}^{n}$ when $n \leq 1$. When $n = 1$, the sum has exactly one term, so it should be equal to that term:

$$\sum_{k=1}^{1} 4k^2 = 4(1)^2 = 4.$$

When $n = 0$ there are *no* terms. What is the sum of no things? Well, if I have no baskets of apples, then the total number of apples I have in all my baskets is, guess what, zero. So a sum with no terms should be zero.

$$\sum_{k=1}^{0} 4k^2 = 0.$$

Another way to deduce this is to use the equation $(*)$ above: if this is to also be valid when $m = 0$ (which the power of triviality suggests that it should be), we must have

$$\left(\sum_{k=1}^{0} k^2\right) + \left(\sum_{k=1}^{n} k^2\right) = \sum_{k=1}^{n} k^2 \qquad \text{and hence, subtracting } \sum_{k=1}^{n} k^2,$$

$$\sum_{k=1}^{0} k^2 = 0.$$

This argument also works to deduce the value of the product of no things:

$$\left(\prod_{k=1}^{0} k^2\right) \cdot \left(\prod_{k=1}^{n} k^2\right) = \prod_{k=1}^{n} k^2 \qquad \text{and hence, canceling } \prod_{k=1}^{n} k^2,$$

$$\prod_{k=1}^{0} k^2 = 1.$$

Technically this argument only works if the product being canceled is nonzero, but the conclusion that the product of no things is 1 is true in all cases. We can give other arguments for it too. For instance, we have

$$2^{\sum_{k=1}^{3} k} = 2^{1+2+3} = 2^1 \cdot 2^2 \cdot 2^3 = \prod_{k=1}^{3} 2^k$$

(here $\sum_{k=1}^{3} k$ is a "condensed" way of writing $\displaystyle\sum_{k=1}^{3} k$ that takes up less vertical space, often used in exponents and in paragraphs). Therefore, by the power of triviality, we should expect also

$$\prod_{k=1}^{0} 2^k = 2^{\sum_{k=1}^{0} k} = 2^0 = 1.$$

Another way of thinking about it is that multiplication is repeated addition:

$$\sum_{k=1}^{5} 4 = 4 + 4 + 4 + 4 + 4 = 5 \cdot 4$$

and therefore we should have

$$\sum_{k=1}^{0} 4 = 0 \cdot 4 = 0,$$

while similarly exponentiation is repeated addition:

$$\prod_{k=1}^{5} 4 = 4 \cdot 4 \cdot 4 \cdot 4 \cdot 4 = 4^5$$

and therefore we should have

$$\prod_{k=1}^{0} 4 = 4^0 = 1.$$

By the way, this also answers the eternally vexing question of 0^0:

$$0^0 = \prod_{k=1}^{0} 0 = 1.$$

In contrast to what you may have been told by algebra books, this is the correct answer to 0^0. Another way to see this is that the function $f(x) = x^0$ is a constant function, so by the power of triviality this should also be the case at $x = 0$, so $f(0) = 0^0 = 1$. The only thing that's weird about 0^0 is that the expression[6]

[6] By the way, this expression x^y is pronounced as "ecks to the why", or, if you prefer, "ecks to the power why". The words "to the" *may not be omitted*. If you pronounce it as "ecks why", I will eject you from my classroom. (Not really.)

x^y is not continuous as x and y *both* approach 0; but in some sense, that's not surprising, since when $x < 0$ this function is not even *defined* except when y is a rational number with odd denominator.[7]

Returning to Σs and Πs, note that as with any bound variable, the index variable can be renamed without changing the value of the sum or product:

$$\sum_{k=1}^{3} k^2 = \sum_{\ell=1}^{3} \ell^2 = \sum_{p=1}^{3} p^2 = 1^2 + 2^2 + 3^2.$$

More interestingly, however, the indices can also be *shifted* or *re-indexed*. For instance, in the sum

$$\sum_{k=1}^{n} k(k+1)$$

we can substitute $k = \ell - 1$ everywhere as long as we also change the upper and lower bounds 1 and n accordingly. Specifically, if $k = \ell - 1$, then when $k = 1$ we have $\ell = 2$; and when $k = n$ we have $\ell = n + 1$. Thus,

$$\sum_{k=1}^{n} k(k+1) = \sum_{\ell=2}^{n+1} (\ell - 1)((\ell - 1) + 1) = \sum_{\ell=2}^{n+1} (\ell - 1)\ell.$$

Finally, we can also "peel off" the first or last term(s) in a sum or product:

$$\sum_{k=1}^{n} k^2 = \left(\sum_{k=1}^{n-1} k^2 \right) + n^2$$

since if we add up all the squares as k goes from 1 to n, we could equivalently add up all the squares as k goes from 1 to $n - 1$ and then add the last term n^2. This is a special case of (∗) when $m = n - 1$ or $m = 1$, but it's worth noting separately since it will be useful in chapter 5.

Exercises

Exercise 3.2.1. Evaluate $\displaystyle\sum_{k=1}^{4} k(k+1)$.

Exercise 3.2.2. Evaluate $\displaystyle\sum_{k=-3}^{3} 2^k$.

Exercise 3.2.3. Evaluate $\displaystyle\prod_{k=-n}^{n} x^k$.

[7] Here I'm talking about real numbers. If you're willing to allow the result to be a *complex* number, then you can define x^y for any complex numbers x and y, but the twist is that in general it will have infinitely many different values, and will still not be continuous as x and y both approach 0. You can learn about this in a subject called *complex analysis*.

Exercise 3.2.4. Evaluate $\prod_{k=1}^{4} 2^k$.

Exercise 3.2.5. Evaluate $\prod_{k=-3}^{3} k$.

Exercise 3.2.6. Re-index $\sum_{k=-3}^{3} 2^k$ using a bound variable ℓ that starts at $\ell = 1$.

Exercise 3.2.7. Re-index $\prod_{k=1}^{n} 2^k$ using a bound variable ℓ that starts at $\ell = 0$.

Exercise 3.2.8. For a fixed $n \in \mathbb{N}$, let $f(x) = \sum_{k=0}^{n} x^k$. Find the derivative $f'(x)$, expressed as a Σ. Then re-index the latter Σ so that its new bound variable ℓ is exactly the exponent of x in each term.

Exercise 3.2.9. Write $\sum_{k=0}^{n} 2^{k-1}$ as a shorter sum plus its last term.

Exercise 3.2.10. Write $\prod_{k=-3}^{3} k$ as a shorter product times its last term.

3.3 Quantifiers ($\forall$ and $\exists$)

In chapter 2 we discussed how the logical operators $\wedge$ and $\vee$ on truth values can be considered as analogous to addition and multiplication on numbers. Thus, it's natural to consider "indexed" versions of $\wedge$ and $\vee$, analogously to Σ and Π for addition and multiplication. For example, we might write something like

$$\begin{aligned}\bigwedge_{k=1}^{5} (k \leq 4) &= (1 \leq 4) \wedge (2 \leq 4) \wedge (3 \leq 4) \wedge (4 \leq 4) \wedge (5 \leq 4) \\ &= \top \wedge \top \wedge \top \wedge \top \wedge \bot \\ &= \bot.\end{aligned}$$

$$\begin{aligned}\bigvee_{k=1}^{5} (k^2 = 9) &= (1^2 = 9) \vee (2^2 = 9) \vee (3^2 = 9) \vee (4^2 = 9) \vee (5^2 = 9) \\ &= \bot \vee \bot \vee \top \vee \bot \vee \bot \\ &= \top.\end{aligned}$$

More generally, if $P(k)$ is a proposition involving a variable k, then $\bigwedge_{k=1}^{n} P(k)$ would mean to evaluate $P(k)$ for all integer values of k from 1 to n inclusive, and

then combine them all with $\wedge$, and similarly $\bigvee_{k=1}^{n} P(k)$ would mean to combine them all with $\vee$. Given the meaning of $\wedge$, these combinations mean that

- $\bigwedge_{k=1}^{n} P(k)$ is true if *all* the $P(k)$ are true, and false otherwise.
- $\bigvee_{k=1}^{n} P(k)$ is true if *at least one* of the $P(k)$ is true, and false otherwise.

These last descriptions, however, don't depend on there being only finitely many truth values $P(k)$. But if there are infinitely many, then we can't in general expect to be able to index them by a sequence of integers. Thus, the binding operators we actually use in practice specify an arbitrary *collection* of values that the variable might take on. If the name of this collection is A (which, for us, will usually be something like $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, etc.), and $P(x)$ is a proposition involving the variable $x \in A$, then these binding operators are written as follows.

- $\forall x \in A, P(x)$ is true if all the $P(k)$ are true, for all $k \in A$, and false otherwise. It is pronounced "for all x in A, $P(x)$." Equivalent phrasings are "for any x in A, $P(x)$" and "for every x in A, $P(x)$", but *not* (contrary to the instincts of many students) "for all of x in A, $P(x)$."
- $\exists x \in A, P(x)$ is true if at least one of the $P(k)$ are true, for some $k \in A$, and false otherwise. It is pronounced "there exists an x in A such that $P(x)$." Equivalent phrasings are "there is an x in A such that $P(x)$" and "there is some x in A such that $P(x)$".

We call these operators *quantifiers* because they "quantify" P by telling us *how many* values of x satisfy $P(x)$ — specifically, all of them or at least one of them. The symbol $\forall$ is an upside-down "A" as in "for All", and the symbol $\exists$ is a backwards "E" as in "there Exists".[8]

If the collection A is obvious from context, we can omit it and write $\forall x, P(x)$ or $\exists x, P(x)$. But it is always present, even if not mentioned, since the variable x (like every variable) always has a type. Also, a sequence of the same quantifier binding different variables with the same type can be compressed into one: $\forall x, y, z \in A, P(x, y, z)$ is an abbreviation of $\forall x \in A, \forall y \in A, \forall z \in A, P(x, y, z)$.

As a concrete example, $\forall x \in \mathbb{R}, x^2 \geq 0$ states that the square of every real number x is nonnegative, which (we know) is true. Whereas $\forall x \in \mathbb{N}, x^2 > x$ states that the square of every natural number is greater than itself, which is false (since $0^2 \not> 0$ and $1^2 \not> 1$). On the other hand,, $\exists x \in \mathbb{R}, x^2 = 2$ states that there exists at least one real number whose square is 2, which is true (in fact, there are two of them, $\sqrt{2}$ and $-\sqrt{2}$). Similarly, $\exists x \in \mathbb{R}, x^2 = -1$ states that there exists a real number whose square is -1, which is false (although it would be true if we replaced $\mathbb{R}$ by $\mathbb{C}$).

[8] You can probably guess why we don't use a backwards "A" or an upside-down "E".

Importantly, if the collection A is infinite, we can't expect to *evaluate* the truth value of a proposition like $\forall x \in A, P(x)$ or $\exists x \in A, P(x)$ in a finite amount of time, as there are infinitely many statements to check. (Although, if a $\forall$ statement is *false* or an $\exists$ statement is *true*, it's possible to *verify* that in a finite amount of time if we can find the example, as we did above for $\forall x \in \mathbb{N}, x^2 > x$ and $\exists x \in \mathbb{R}, x^2 = 2$.) This provides the more general reason for the necessity of proof that I promised in section 1.1: the truth value of propositions involving $\forall$ and $\exists$ cannot be simply computed, so we need proofs to establish their truth. For example, the statement

$$\forall x, y, z, n \in \mathbb{N}, \Big(\big(x > 0 \wedge y > 0 \wedge z > 0 \wedge n > 2 \big) \Rightarrow x^n + y^n \neq z^n \Big),$$

known as *Fermat's Last Theorem*, was guessed to be true by Pierre de Fermat in 1637, but mathematicians tried and failed to prove it for hundreds of years, until at the end of the 20th century it was finally proven by a combination of hundreds of pages of work by Ken Ribet and Andrew Wiles.[9] So even a very simple-looking statement with a few $\forall$s in it can be very difficult to prove!

In the next few sections, we will introduce the proof rules for the quantifiers one by one.

Exercises

Exercise 3.3.1. Test your power of triviality! Suppose A is empty (contains no elements); what are the truth values of the following statements?

(a) $\forall x \in A, P(x)$

(b) $\exists x \in A, P(x)$

(Neither of the answers depend on P).

Exercise 3.3.2. Test your power of triviality again! What are the truth values of the following statements? Each answer *might* depend on whether A is empty... or it might not.

(a) $\forall x \in A, \top$

(b) $\exists x \in A, \top$

(c) $\forall x \in A, \bot$

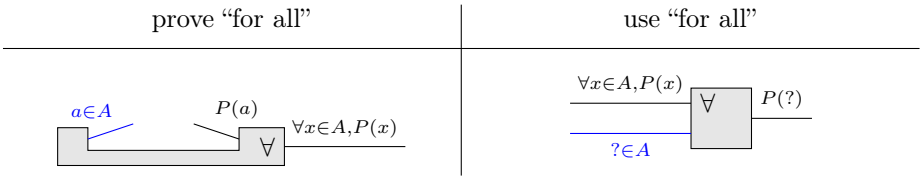
(d) $\exists x \in A, \bot$

[9] Amusingly, Fermat's Last Theorem is another case where it matters what number system we're working in. The corresponding statement in which x, y, z are 10-adic integers is false: there is a 10-adic integer $z = \cdots 60569$ such that $1^3 + 2^3 = z^3$.

3.4 For all (∀)

For the same reason that we can't compute the value of $\forall x \in A, P(x)$ when A is infinite, we can't expect to prove it by proving P individually for each element of A. Instead, we can give a *generic* or *schematic proof* that could be specialized to any element of A that we might want. The way we do this is to introduce a new *free variable* whose type is A, say we call it a, and prove that $P(a)$ for this *specific but unspecified* element of A. If we do this, then because a free variable can be substituted by anything having its type, if we're given any particular element of A, we could substitute it for the free variable a in our proof of $P(a)$ and obtain a proof of P for that element. Therefore, we'll know that $P(x)$ is indeed true for every $x \in A$.

To represent this in our graphical notation, we introduce a new kind of wire labeled by an *object* rather than a proposition, and we make the *prove "for all"* block a "bracket" that introduces a new wire labeled by our new variable $a \in A$ from the left and uses it to prove $P(a)$. Dually, if we know that $\forall x \in A, P(x)$, the *use "for all"* rule says that if we have any particular element of A we can conclude P of that element. So here are the ∀ rules in graphical notation:



We have colored the new "element" wires blue, to distinguish them from the old "proposition" wires. Note that apart from this color, these rules look very similar to those for ⇒; this is not a coincidence, but explaining it would take us too far afield into the realm of "dependent type theory".

The most important thing to note about the *prove "for all"* rule is that a must be a *new free variable* that doesn't appear as a free variable anywhere else in the proof. If you use the *prove "for all"* rule more than once, for each use you must choose a different variable.[10] Moreover, as with the *prove "if-then"* rule, the wire labeled by $a \in A$ cannot "escape" from the bracket: it can only be used to prove the goal $P(a)$, which is obtained from the statement to prove $\forall x \in A, P(x)$ by substituting the *new* variable a for the ∀-*bound* variable x.

In the *use "for all"* rule, I have labeled the input element wire with a question-mark "?" rather than a letter. I do this to emphasize that this wire is *not a new variable*, but rather an *expression* built out of variables that *already exist* in the proof at that point (which could be just a single one of those variables, or a constant not using any of them, but could also put together more than one of them). The label on that wire is something that *you must choose* to replace the question-mark when you apply that rule; and when you do choose it, you should

[10] It is officially, technically, possible to relax this requirement if the brackets of the two rules don't "overlap" at all, but I don't recommend it. The technical name for this requirement is the *Barendregt convention*.

also replace the question-mark in the output proposition $P(?)$ by the expression you chose. In other words, the output proposition is obtained by replacing the $\forall$-bound variable x in its scope $P(x)$ by the expression that you choose. *When your proof is complete, there should no longer be any question-marks in it.*[11]

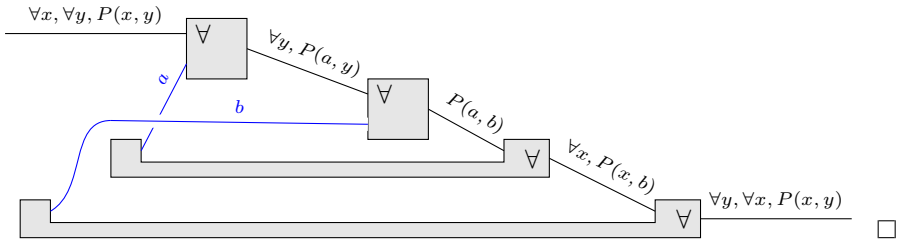
If this chosen label is simply a free variable that already exists in the proof, then you can connect it to the source of the wire labeled by that variable. If instead the chosen label is an expression, we can connect it to the output of a box labeled "algebra" (or "alg") but colored blue to indicate that it computes a value rather than an equation, and whose inputs are wires labeled by all the variables used in that expression.

This means we need wires labeled by all the available variables. We automatically have such wires for variable introduced by some other *prove "for all"* block, and once we introduce $\exists$ in section 3.6 we will also have them for variables introduced by a *use "exists"* block. Many of our algebraic examples have also included given variables for the entire theorem; we will discuss this further in section 3.9, but for now we will simply include a wire labeled by each of those variables as inputs on the left of the overall graph. In both cases, connecting these wires serves to remind us that wires coming from inside a bracket cannot "escape" outside that bracket.

This is a lot of verbiage, so here's an example to make it clearer. Let's prove that $\forall$ "commutes with itself".

Theorem 3.1 (Olorin 3-1-5)**.** *Suppose* $\forall x \in A, \forall y \in B, P(x, y)$*. Then* $\forall y \in B, \forall x \in A, P(x, y)$*.*

Graphical Proof of 3.1. For conciseness, we omit the types "$\in A$" and "$\in B$" from the wire labels in the graphical proof.



□

To describe the *use "for all"* rule in English we can simply use "since" (PEP 1), although sometimes we emphasize what's going on with a phrase like "in particular". For *prove "for all"*, we use the following variation of PEP 3.

Principle of English Proof 4 (Let)**.** When introducing a *new free variable*, which is represented graphically by a new blue wire from the left, indicate it with a word like "suppose", "assume", or "let". You should generally indicate its type as well, as in "let x be a real number" or "let $x \in \mathbb{R}$".

We will also find the following helpful.

[11] In technical parlance, the question-mark is called a *meta-variable*.

Principle of English Proof 5 (Want to show)**.** When the "current goal" (that is, the wire coming from the right that you are trying to connect up with) changes, such as when applying a *prove "if-then"* or *prove "for all"*, it may be helpful to remind the reader of what the new goal is. But *whenever* you quote a statement in a proof that is *not yet known to be true*, such as the current goal, you *MUST* label it as such with a phrase like "we want to show that" or "we will show that" or "we must show that". Otherwise, the reader will assume that you are claiming you already *know* that statement to be true, and will be quite confused for a while trying to figure out how you know it, because you don't.

This gives us the following.

English Proof of 3.1. Let $b \in B$; we must show $\forall x \in A, P(x, b)$. Now let $a \in A$; we must show $P(a, b)$. But since $\forall x \in A, \forall y \in B, P(x, y)$, we have $\forall y \in B, P(a, y)$, and therefore also $P(a, b)$. □

Exercises

As always, give both a graphical proof and an English proof. This section pertains to Stages 3-1 and 3-2 of Olorin.[12]

Exercise 3.4.1 (Olorin 3-1-3)**.** Suppose $\forall x \in A, (P \Rightarrow Q(x))$.
Prove $P \Rightarrow (\forall x \in A, Q(x))$. *(Here P is a statement not involving x.)*

Exercise 3.4.2 (Olorin 3-1-4)**.** Suppose $P \Rightarrow (\forall x \in A, Q(x))$.
Prove $\forall x \in A, (P \Rightarrow Q(x))$.

Exercise 3.4.3 (Olorin 3-1-6)**.** Prove $\forall x \in A, \top$.

Exercise 3.4.4 (Olorin 3-1-7)**.** Suppose $\forall x \in A, \forall y \in A, P(x, y)$.
Prove $\forall x \in A, P(x, x)$.

Exercise 3.4.5 (Olorin 3-1-8)**.** Suppose $\forall x \in A, (P(x) \Rightarrow Q(x))$.
Prove $(\forall x \in A, P(x)) \Rightarrow (\forall x \in A, Q(x))$.

Exercise 3.4.6 (Olorin 3-2-2)**.** Suppose $P \wedge (\forall x \in A, Q(x))$.
Prove $\forall x \in A, (P \wedge Q(x))$.

Exercise 3.4.7 (Olorin 3-2-3)**.** Suppose $P \vee (\forall x \in A, Q(x))$.
Prove $\forall x \in A, (P \vee Q(x))$.

[12] If you are using The Incredible Proof Machine instead, you should drop it here and switch to Olorin or to paper (physical or electronic). The Incredible Proof Machine does include a session (Session 6) with proofs involving ∀ and ∃, but unfortunately it represents these proofs differently, without any blue wires to indicate where the variables go. This is not only more confusing, but actually *wrong* in some cases; specifically, the fourth and fifth exercises in its Session 6 are incorrect and should not be provable. Its rules for quantifiers are victims of the trap of triviality: they are correct only when the collection A is nonempty.

Exercise 3.4.8 (Olorin 3-2-4)**.** Suppose $\forall x \in A, (P(x) \wedge Q(x))$.
Prove $(\forall x \in A, P(x)) \wedge (\forall x \in A, Q(x))$.

Exercise 3.4.9 (Olorin 3-2-5)**.** Suppose $(\forall x \in A, P(x)) \wedge (\forall x \in A, Q(x))$.
Prove $\forall x \in A, (P(x) \wedge Q(x))$.

Exercise 3.4.10 (Olorin 3-2-6)**.** Suppose $(\forall x \in A, P(x)) \vee (\forall x \in A, Q(x))$.
Prove $\forall x \in A, (P(x) \vee Q(x))$.

Exercise 3.4.11 (Olorin 3-2-7)**.** Suppose $a \in A$, and $\forall x \in A, (P \wedge Q(x))$.
Prove $P \wedge (\forall x \in A, Q(x))$.

Exercise 3.4.12. Suppose only that $\forall x \in A, (P \wedge Q(x))$.
Try to prove $P \wedge (\forall x \in A, Q(x))$. Can you explain why you failed? Can you find a collection A and statements P and $Q(x)$ such that $\forall x \in A, (P \wedge Q(x))$ is true but $P \wedge (\forall x \in A, Q(x))$ is not?

Exercise 3.4.13. Suppose $\forall x \in A, (P(x) \vee Q(x))$.
Try to prove $(\forall x \in A, P(x)) \vee (\forall x \in A, Q(x))$.
Can you explain why you failed? Can you find an example of a collection A and statements $P(x)$ and $Q(x)$ such that $\forall x \in A, (P(x) \vee Q(x))$ is true but $(\forall x \in A, P(x)) \vee (\forall x \in A, Q(x))$ is not?

Exercise 3.4.14. Suppose $(\forall x \in A, P(x)) \Rightarrow (\forall x \in A, Q(x))$. Try to prove $\forall x \in A, (P(x) \Rightarrow Q(x))$. Can you explain why you failed? Can you find a collection A and statements P and $Q(x)$ such that $(\forall x \in A, P(x)) \Rightarrow (\forall x \in A, Q(x))$ is true but $\forall x \in A, (P(x) \Rightarrow Q(x))$ is not?

3.5 $\forall$ with algebra

The proofs in section 3.4 were all "abstract", involving arbitrary statements $P(x)$ and $Q(x)$ denoted by letters. But, as in chapter 2, in practice we are more interested in "concrete" proofs, involving specific statements such as equations and inequalities, which we can write by combining the proof rules for logical operators with those of algebra.

Since statements involving quantifiers start to look rather long and complicated, here is also where we start to encounter more *definitions*. A "definition" in mathematics is radically different from the sort of definition you find in the dictionary. The definition of a word in a dictionary is intended as a *description* of *how a word is already used*; no lexicographer contemplates *creating a new definition* for a word by writing it into the dictionary and expecting people to start using it that way. But in mathematics that is exactly what we do: a mathematical definition *creates a new meaning* for a word or phrase, and henceforth (at least in the paper or book where the definition appears), that word or phrase will always and only ever be used with *exactly that meaning*.

For example, here is a definition involving a $\forall$. In a definition we generally italicize or bold the word or phrase being defined.

Definition 3.2. [13] *A surreal number x is **positive infinite** if $\forall u \in \mathbb{R}, x > u$.*

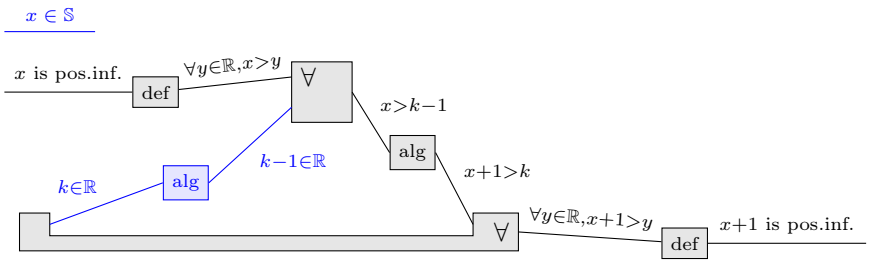
Now, whenever we encounter a statement in a proof (either a given or a goal) that some surreal number is "positive infinite", we can — and usually should — replace it by the given ∀-statement. This is important enough to emphasize:

Proof Guidance 7 (Use definitions)**.** If you don't see a way to proceed at the beginning of a proof, and it's not clear how to apply Proof Guidance 1, look for words that have definitions and replace them by those definitions.

In a graphical proof, we may indicate the use of a definition with a block labeled "def", in either direction. In an English proof, we often don't even mention that a definition is getting expanded, but if we want we can indicate it with a "since". For example, here is a simple proof.

Theorem 3.3. *Suppose $x \in \mathbb{S}$ is positive infinite. Then $x + 1$ is also positive infinite.*

Graphical Proof of 3.3.



Algebra:

$$x > k - 1$$
$$x + 1 > k$$

□

What is that unconnected blue wire with "$x \in \mathbb{S}$" on it doing at the top? Well, you probably skimmed over it without noticing, but back in section 3.4 when I introduced blue wires, I said

> Many of our algebraic examples have also included given variables for the entire theorem; we will discuss this further in section 3.9, but for now we will simply include a wire labeled by each of those variables as inputs on the left of the overall graph.

[13] Note that this definition is numbered 3.2 even though there have not been any previous definitions in this chapter. This is because there was already a Theorem 3.1. It is good practice to number theorems (including lemmas, corollaries, and so on) and definitions with the same sequence; this makes it easier for the reader to find things. When (not "if") you learn to typeset mathematics in LaTeX, there are packages that will do this for you, like `thmtools`.

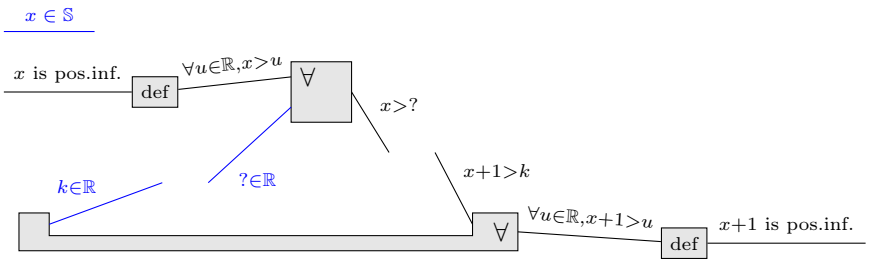
In the present notes, I decided to exclude Proof Guidances and Principles of English Proof from this numbering sequence so as to emphasize the total *number* of such things we have. All Proof Guidances and Principles of English Proof are restated in appendices A and B, so you should have no trouble finding them when needed.

This is just the first time we have had occasion to implement that remark. In this case, our value $k-1$ doesn't involve this given variable x, so we don't need to connect the x wire to anything; but we include it there for consistency.

English Proof of 3.3. Let $k \in \mathbb{R}$; we must show $x+1 > k$. Since x is positive infinite, in particular we have $x > k-1$. Therefore, $x+1 > k$. □

Now how, you may ask, do we *construct* this proof? How do we guess that the value to put into the *use "for all"* rule should be $k-1$?

We start by writing out as much of the proof as we can without knowing what that wire should be. Namely, we use the definitions of both the given and the goal, and (following Proof Guidance 1) the *prove "for all"* rule for the goal, introducing a variable $k \in \mathbb{R}$. Then we guess that since we need to prove an inequality $x+1 > k$, and we have the statement $\forall y \in \mathbb{R}, x > y$ that tells us about inequalities, it will be useful to use that, so we write down a *use "for all"* rule with question-marks:



Now we apply Proof Guidance 4. In our scratch work, we start from the current goal $x+1 > k$ (which, remember, we couldn't do *in* the proof) and do algebra to it:

$$\begin{aligned} x+1 &> k \\ x &> k-1 \end{aligned}$$

and *voilà*, we have an inequality of the form $x > ?$ that tells us what ? should be, namely $k-1$. Now we can replace both question-marks with $k-1$ and fill in the rest of the proof with algebra blocks.

Exercises

Exercise 3.5.1. Suppose $x \in \mathbb{S}$ is positive infinite. Prove that $x-1$ is also positive infinite.

Exercise 3.5.2. Suppose $x \in \mathbb{S}$ is positive infinite. Prove that $2x$ is also positive infinite.

Exercise 3.5.3. Suppose $x, y \in \mathbb{S}$ are both positive infinite (that is, x is positive infinite and y is positive infinite). Prove that $x+y$ is also positive infinite.

Exercise 3.5.4. Suppose $x, y \in \mathbb{S}$ are both positive infinite. Prove that xy is also positive infinite.

Of course, we define "negative infinite" dually to "positive infinite".

Definition 3.4. *A surreal number x is **negative infinite** if $\forall u \in \mathbb{R}, x < u$.*

Exercise 3.5.5. Suppose $x \in \mathbb{S}$ is negative infinite and $y \in \mathbb{R}$. Prove that $x+y$ is negative infinite.

Exercise 3.5.6. Suppose $x \in \mathbb{S}$ is negative infinite. Prove that $2x$ is also negative infinite.

Exercise 3.5.7. Suppose $x, y \in \mathbb{S}$ are both negative infinite. Prove that $x + y$ is also negative infinite.

Exercise 3.5.8. Suppose $x, y \in \mathbb{S}$ are both negative infinite. Prove that xy is positive infinite.

Exercise 3.5.9. Suppose $x \in \mathbb{S}$ is positive infinite and $y \in \mathbb{S}$ is negative infinite. Prove that xy is negative infinite.

The next few exercises use the following additional definition. We write $\mathbb{R}_{>0}$ for the collection of *positive* real numbers. Recall the definition and properties of absolute value for surreal numbers from Exercises 2.6.8, 2.6.9 and 2.6.10.

Definition 3.5. *An $x \in \mathbb{S}$ is **infinitesimal** if $\forall u \in \mathbb{R}_{>0}, |x| < u$.*

Exercise 3.5.10. Prove that 0 is infinitesimal.

Exercise 3.5.11. Suppose $x, y \in \mathbb{S}$ are infinitesimal. Prove that $x + y$ is also infinitesimal.

Exercise 3.5.12. Suppose $x, y \in \mathbb{S}$ are infinitesimal. Prove that $x - y$ is also infinitesimal.

Exercise 3.5.13. Suppose $x, y \in \mathbb{S}$ are infinitesimal. Prove that xy is also infinitesimal.

At this point we're already starting to get into proofs that are too large to be practical in a graphical representation. So for the rest of the exercises, feel free to write them only in English. You may also want to review section 2.6.

Definition 3.6. *For $x, y \in \mathbb{S}$ we say x and y are **infinitely close**, and write $x \approx y$, if $|x - y|$ is infinitesimal, i.e. $\forall u \in \mathbb{R}_{>0}, |x - y| < u$.*

Exercise 3.5.14. Suppose $x \in \mathbb{S}$. Prove $x \approx x$.

Exercise 3.5.15. Suppose $x, y \in \mathbb{S}$ and $x \approx y$. Suppose $y \approx x$.

Exercise 3.5.16. Suppose $x, y \in \mathbb{S}$ and $x \approx y$ and $y \approx z$. Prove $x \approx z$.

Exercise 3.5.17. Suppose $x, y, z \in \mathbb{S}$ and $x \approx y$. Suppose $x + z \approx y + z$.

Exercise 3.5.18. Suppose $x, y \in \mathbb{S}$ and $x \approx y$. Suppose $2x \approx 2y$.

Exercise 3.5.19. Suppose $x, y \in \mathbb{S}$ and $x \approx y$. Suppose $-x \approx -y$.

Exercise 3.5.20. Suppose $x, y, z \in \mathbb{S}$ and $x \approx y$. Suppose $x - z \approx y - z$.

Definition 3.7. *A surreal number x is **infinite** if it is either positive infinite or negative infinite.*

Exercise 3.5.21. Suppose $x \in \mathbb{S}$ is infinite. Prove that $\frac{1}{x}$ is infinitesimal.

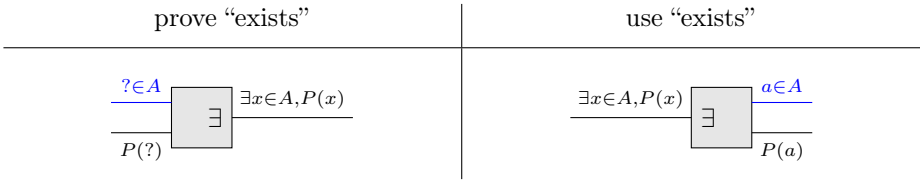
Exercise 3.5.22. Suppose $x \in \mathbb{S}$ is infinitesimal and $x > 0$. Prove that $\frac{1}{x}$ is positive infinite.

Exercise 3.5.23. Suppose $x \in \mathbb{S}$ is infinitesimal and $x < 0$. Prove that $\frac{1}{x}$ is negative infinite.

Exercise 3.5.24. Suppose $x \in \mathbb{S}$ is infinitesimal and $y \in \mathbb{R}$. Prove that xy is infinitesimal.

3.6 There exists ($\exists$)

The proof rules for $\exists$ are related to those for $\vee$ the same way the rules for $\forall$ are related to those for $\wedge$. Here are their graphical representations:



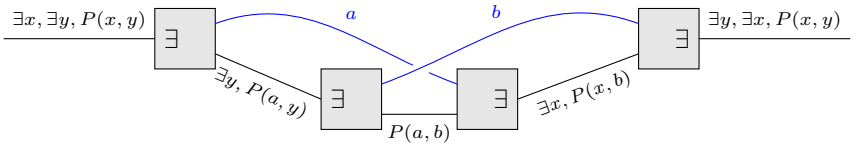
Just as to prove $P \vee Q$ we must choose whether to prove P or to prove Q, to prove $\exists x \in A, P(x)$ we must choose one element of A about which to prove P. Thus, the *prove "exists"* rule has two input wires, one of which is a blue "value" wire that specifies the element, and the other of which is a "proposition" wire that proves P about that element. As with the *use "for all"* rule, I've labeled the value wire in this rule with a question-mark to emphasize that *you must choose the element*, which could be an arbitrary expression involving the free variables that are available at that point of the proof. As before, in general you should expect to do some scratch work to figure out the value of ?, at which point you should replace the ? by that value. Again, *when your proof is complete there should no longer be any question-marks in it.*

For the *use "exists"* rule, since A could have infinitely many elements, we can't expect to divide a proof into one case for each of them. Instead, as with the *prove "for all"* rule, we introduce a *new free variable* belonging to A, about which we assume nothing *except* that it satisfies P. As before, this variable must be different from all other variables occurring in the proof.

Let's look at an example.

Theorem 3.8 (Olorin 3-3-4)**.** *Suppose* $\exists x \in A, \exists y \in B, P(x, y)$.
Then $\exists y \in B, \exists x \in A, P(x, y)$.

Graphical Proof of 3.8. For conciseness, we omit the types "$\in A$" and "$\in B$" from the wire labels in the graphical proof.



□

To describe the *use "exists"* rule in English, we need another principle.

Principle of English Proof 6 (Such that)**.** When introducing a new variable *along with* a property of that variable (that is, a value wire along with a proposition wire involving it coming from the left), as in the *use "exists"* rule, label the variable with "let", "assume", or "suppose" as in PEP 4, and then label its property with "such that", as in "let x be a real number such that $x^2 = 2$". This is the *only* place that you should use "such that".

For the *prove "exists"* rule, we can often simply use the "since" principle (PEP 1), referring only to the statement $P(?)$. This works as long as the reader can easily guess from $P(?)$ what the chosen value ? is. For example, we have the following.

English Proof of 3.8. Since $\exists x \in A, \exists y \in B, P(x, y)$, we can let $a \in A$ be such that $\exists y \in B, P(a, y)$. Thus, we can let $b \in B$ be such that $P(a, b)$. Hence $\exists x \in A, P(x, b)$, and therefore $\exists y \in B, \exists x \in A, P(x, y)$. □

Exercises using ∃

As always, give both a graphical proof and an English proof. The following exercises pertain to Stages 3-3 and 3-4 of Olorin.

Exercise 3.6.1 (Olorin 3-3-5)**.** Suppose $\exists x \in A, (P \Rightarrow Q(x))$ and P.
Prove $\exists x \in A, Q(x)$.

Exercise 3.6.2 (Olorin 3-3-6)**.** Suppose $a \in A$ and $(\exists x \in A, P(x)) \Rightarrow Q$.
Prove $P(a) \Rightarrow Q$.

Exercise 3.6.3 (Olorin 3-3-7)**.** Suppose $\exists x \in A, P(x)$. Prove $\exists x \in A, \top$.

Exercise 3.6.4 (Olorin 3-3-8)**.** Suppose $\exists x \in A, \bot$. Prove P.

Exercise 3.6.5 (Olorin 3-3-9)**.** Suppose $\exists x \in A, P(x, x)$.
Prove $\exists x \in A, \exists y \in A, P(x, y)$.

Exercise 3.6.6 (Olorin 3-4-1)**.** Suppose $\exists x \in A, (P \wedge Q(x))$.
Prove $P \wedge (\exists x \in A, Q(x))$.

Exercise 3.6.7 (Olorin 3-4-2)**.** Suppose $P \wedge (\exists x \in A, Q(x))$.
Prove $\exists x \in A, (P \wedge Q(x))$.

Exercise 3.6.8 (Olorin 3-4-3)**.** Suppose $\exists x \in A, (P(x) \wedge Q(x))$.
Prove $(\exists x \in A, P(x)) \wedge (\exists x \in A, Q(x))$.

Exercise 3.6.9 (Olorin 3-4-4)**.** Suppose $\exists x \in A, (P \vee Q(x))$.
Prove $P \vee (\exists x \in A, Q(x))$.

Exercise 3.6.10 (Olorin 3-4-5)**.** Suppose $a \in A$ and $P \vee (\exists x \in A, Q(x))$.
Prove $\exists x \in A, (P \vee Q(x))$.

Exercise 3.6.11 (Olorin 3-4-6)**.** Suppose $\exists x \in A, (P(x) \vee Q(x))$.
Prove $(\exists x \in A, P(x)) \vee (\exists x \in A, Q(x))$.

Exercise 3.6.12 (Olorin 3-4-7)**.** Suppose $(\exists x \in A, P(x)) \vee (\exists x \in A, Q(x))$.
Prove $\exists x \in A, (P(x) \vee Q(x))$.

Exercise 3.6.13. Suppose $(\exists x \in A, P(x)) \wedge (\exists x \in A, Q(x))$.
Try to prove $\exists x \in A, (P(x) \wedge Q(x))$. Can you explain why you failed? Can you find an example of a collection A and statements $P(x)$ and $Q(x)$ such that $(\exists x, P(x)) \wedge (\exists x, Q(x))$ is true but $\exists x, (P(x) \wedge Q(x))$ is not?

This is a good point at which to go back and start solving the levels from "Advanced Proposition World" on the Adept difficulty setting, and/or those from "Proposition World" on the Master difficulty setting.

Exercises combining $\forall$ and $\exists$

These exercises pertain to Stages 3-5 and 3-6 of Olorin.

Exercise 3.6.14 (Olorin 3-5-1)**.** Suppose $\exists x \in A, \forall y \in B, P(x, y)$.
Prove $\forall y \in B, \exists x \in A, P(x, y)$.

Exercise 3.6.15 (Olorin 3-5-2)**.** Suppose $(\exists x \in A, P(x)) \Rightarrow Q$.
Prove $\forall x \in A, (P(x) \Rightarrow Q)$.

Exercise 3.6.16 (Olorin 3-5-3)**.** Suppose $\forall x \in A, (P(x) \Rightarrow Q)$.
Prove $(\exists x \in A, P(x)) \Rightarrow Q$.

Exercise 3.6.17 (Olorin 3-6-1)**.** Suppose $\forall x \in A, P(x)$ and $\exists x \in A, Q(x)$.
Prove $\exists x \in A, (P(x) \wedge Q(x))$.

Exercise 3.6.18 (Olorin 3-5-4)**.** Suppose $\forall x \in A, P(x)$.
Prove $(\exists x \in A, (P(x) \Rightarrow Q)) \Rightarrow Q$.

Exercise 3.6.19 (Olorin 3-5-5)**.** Suppose $\exists x \in A, (P(x) \Rightarrow Q)$ and $\forall x \in A, P(x)$. Prove Q.

Exercise 3.6.20 (Olorin 3-6-2)**.** Suppose $\forall x \in A, P(x)$ and $\exists x \in A, \top$.
Prove $\exists x \in A, P(x)$.

Exercise 3.6.21 (Olorin 3-5-6)**.** Suppose $\forall x \in A, (P(x) \Rightarrow Q(x))$ and $\exists x \in A, P(x)$. Prove $\exists x \in A, Q(x)$.

Exercise 3.6.22 (Olorin 3-5-7)**.** Suppose $\exists x \in A, (P(x) \Rightarrow Q)$.
Prove $(\forall x \in A, P(x)) \Rightarrow Q$.

Exercise 3.6.23 (Olorin 3-6-3)**.** Suppose $P \Rightarrow \forall x \in A, \bot$ and $\exists x \in A, \top$.
Prove $P \Rightarrow Q$.

Exercise 3.6.24 (Olorin 3-5-8)**.** Suppose $\forall x \in A, \exists y \in B, P(x, y)$ and $\forall y \in B, \exists z \in C, Q(y, z)$ and $\forall x \in A, \forall y \in B, \forall z \in C, ((P(x, y) \wedge Q(y, z)) \Rightarrow R(x, z))$.
Prove $\forall x \in A, \exists z \in C, R(x, z)$.

Exercise 3.6.25. Suppose $\forall y \in B, \exists x \in A, P(x, y)$.
Try to prove $\exists x \in A, \forall y \in B, P(x, y)$. Can you explain why you failed? Can you find an example of collections A and B and a statement $P(x, y)$ such that $\forall y \in B, \exists x \in A, P(x, y)$ is true but $\exists x \in A, \forall y \in B, P(x, y)$ is not?

3.7 ∃ with algebra

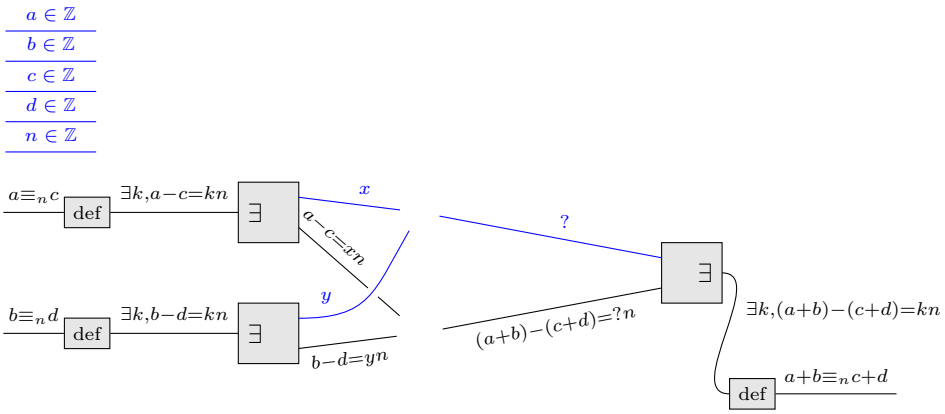
As in section 3.5, when we deal with concrete statements involving ∃ we start to use definitions. As an example, we can start to make the number system $\mathbb{Z}_{/n}$ precise. In section 1.5 I said that in that world, we consider a number to be unchanged if we "go around the n-hour clock", i.e. we add n to it. If we repeat this, we find that two numbers should be equivalent if we can get from one to the other by adding some *multiple* of n, or equivalently if the *difference* between those numbers is a multiple of n. This leads to the following definition.

Definition 3.9. *For integers a, b, n, we say a **is congruent to** b **modulo** n, and write $a \equiv_n b$, if $\exists k \in \mathbb{Z}, (a - b = kn)$.*

We should then have $[a]_n = [b]_n$ precisely when $a \equiv_n b$. However, in order for $\mathbb{Z}_{/n}$ to make sense as a "number system", this relation must *behave like equality should*. I'm not going to make precise what that means here (although your instructor and supplementary textbook will probably discuss it, either here or at some other point; look out for *equivalence relations*). But at least, it means we should be able to replace a number in an algebraic expression by anything that we "consider the same" without changing its value, or at least only changing it to something that we also consider the same. For addition, this is the following statement:

Theorem 3.10. *Suppose a, b, c, d, n are integers and $a \equiv_n c$ and $b \equiv_n d$. Then $a + b \equiv_n c + d$.*

As in section 3.5, we start to create this proof by writing out as much of it as we can using Proof Guidances 7 and 1.

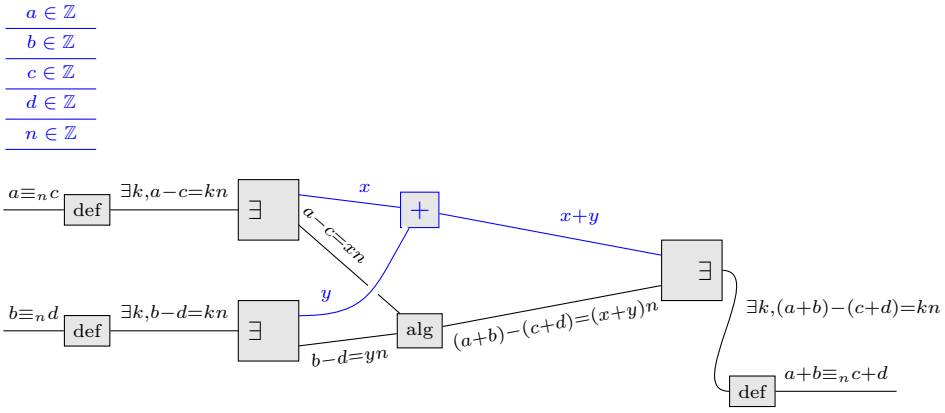


Now we have to figure out the value of ?. We do some scratch work, starting from the goal:

$$\begin{aligned}(a+b)-(c+d) &= ?n \\ (a-c)+(b-d) &= ?n \\ xn+yn &= ?n \\ (x+y)n &= ?n \\ x+y &= ?\end{aligned}$$

Thus, we should pick $? = x + y$. Plugging this in, we can complete the proof:

Graphical Proof of 3.10.



Algebra: Adding $a - c = xn$ and $b - d = yn$, we get

$$\begin{aligned}(a-c)+(b-d) &= xn+yn \\ (a+b)-(c+d) &= (x+y)n.\end{aligned}$$

□

English Proof of 3.10. Since $a \equiv_n c$, we can let $x \in \mathbb{Z}$ be such that $a-c = xn$. And since $b \equiv_n d$, we can let $y \in \mathbb{Z}$ be such that $b-d = yn$. Adding $a-c = xn$ and $b-d = yn$, we get

$$\begin{aligned}(a-c)+(b-d) &= xn+yn\\ (a+b)-(c+d) &= (x+y)n.\end{aligned}$$

Therefore, $a+b \equiv_n c+d$. □

As suggested in section 3.6, we indicate the *prove "exists"* rule with the simple "therefore", figuring that the reader can guess from the preceding statement $(a+b)-(c+d) = (x+y)n$ that the value we're using is $x+y$. But if we want to help the reader out by being more explicit, we can use the following principle.

Principle of English Proof 7 (Let, again)**.** When specifying a value for a bound variable, as in the *use "for all"* and *prove "exists"* rule, if you want to explicitly indicate the value, you can say "let $x = \langle$the value$\rangle$", where x is the bound variable, *as long as there is no other x around that could create confusion.* If there is a potential for confusion, you should rename the bound variable (which, as we know, is always possible). When you specify a value explicitly like this, you should usually also state the property that this value makes the bound variable satisfy, either with or without its binder, but using the name you chose for the bound variable.

For example:

More Verbose English Proof of 3.10. Since $a \equiv_n c$, we can let $x \in \mathbb{Z}$ be such that $a-c = xn$. And since $b \equiv_n d$, we can let $y \in \mathbb{Z}$ be such that $b-d = yn$. Let $k = x+y$. Then adding $a-c = xn$ and $b-d = yn$, we get

$$\begin{aligned}(a-c)+(b-d) &= xn+yn\\ (a+b)-(c+d) &= (x+y)n\\ &= kn.\end{aligned}$$

Therefore $\exists k \in \mathbb{Z}, (a+b)-(c+d) = kn$, so $a+b \equiv_n c+d$. □

Exercises

Prove the following exercises with both a graphical proof and an English proof.

Exercise 3.7.1. Suppose a and n are integers. Prove $a \equiv_n a$.

Exercise 3.7.2. Suppose a, b, n are integers and $a \equiv_n b$. Prove $b \equiv_n a$.

Exercise 3.7.3. Suppose a, b, n are integers and $a \equiv_n b$. Prove $a \equiv_{-n} b$.

Exercise 3.7.4. Suppose a, b, c, n are integers and $a \equiv_n b$ and $b \equiv_n c$. Prove $a \equiv_n c$.

Exercises 3.7.1, 3.7.2 and 3.7.4 say that $\equiv_n$ is an *equivalence relation*.

Exercise 3.7.5. Suppose a, b, c, n are integers and $a \equiv_n b$. Prove $ac \equiv_n bc$.

Exercise 3.7.6. Suppose $a, b \in \mathbb{Z}$. Prove $a \equiv_1 b$.

Exercise 3.7.7. Suppose $a, b \in \mathbb{Z}$ and $a \equiv_0 b$. Prove $a = b$.

Exercise 3.7.8. Suppose $a, b, n \in \mathbb{Z}$. and $a \equiv_n b$. Prove $a^2 \equiv_n b^2$.

Exercise 3.7.9. Suppose $a, b, c, d, n \in \mathbb{Z}$ and $a + c \equiv_n b + d$, and $c \equiv_n d$. Prove $a \equiv_n b$.

Here is another simple definition using $\exists$.

Definition 3.11. *For integers a, b, we say a **divides** b, and write $a \mid b$, if $\exists k \in \mathbb{Z}, (b = ka)$.*

For example, $2 \mid b$ is the same as saying that b is even. Also, $a \equiv_n b$ is the same as saying that $n \mid (a - b)$; thus b is even just when $b \equiv_2 0$, and odd just when $b \equiv_2 1$.

Exercise 3.7.10. Suppose $a, b, c \in \mathbb{Z}$ and $a \mid b$. Prove $ac \mid bc$.

Exercise 3.7.11. Suppose $a, b, c \in \mathbb{Z}$ and $a \mid b$. Prove $a \mid bc$.

Exercise 3.7.12. Suppose $a, b, c \in \mathbb{Z}$ and $a \mid b$ and $a \mid c$. Prove $a \mid (b + c)$.

Exercise 3.7.13. Suppose $a \in \mathbb{Z}$. Prove $a \mid 0$.

Exercise 3.7.14. Suppose $b \in \mathbb{Z}$. Prove $1 \mid b$.

Exercise 3.7.15. Suppose $b \in \mathbb{Z}$ and $0 \mid b$ Prove $b = 0$.

Exercise 3.7.16. Suppose $a, b, m, n \in \mathbb{Z}$ and $a \equiv_n b$ and $m \mid n$. Prove $a \equiv_m b$.

Exercise 3.7.17. Suppose $a, b, c \in \mathbb{Z}$ and $a \mid b$ and $b \mid c$. Prove $a \mid c$.

Exercise 3.7.18. Suppose $a, b, c, m, n \in \mathbb{Z}$ and $a \equiv_n b$ and $m \mid c$. Prove $ac \equiv_{mn} bc$.

Exercise 3.7.19. Suppose $a, b, c, m, n \in \mathbb{Z}$ and $a \equiv_m b$ and $n \mid m$. Prove $a \equiv_n b$.

Here is a different kind of definition using $\exists$. Recall that $\mathbb{R}_{>0}$ denotes the collection of positive real numbers.

Definition 3.12. *An $x \in \mathbb{S}$ is **finite** if $\exists u \in \mathbb{R}_{>0}, |x| < u$.*

Once again we are getting into proofs that are too large for a graphical representation to be practical, so feel free to write these last exercises only in English. The definitions of "infinite" and "infinitesimal" and "infinitely close" ($\approx$) were given in section 3.5 and its exercises.

Exercise 3.7.20. Suppose $x \in \mathbb{R}$. Prove x is finite.

Exercise 3.7.21. Suppose $x \in \mathbb{S}$ is finite. Prove $x + 1$ is also finite.

Exercise 3.7.22. Suppose $x, y \in \mathbb{S}$ are both finite. Prove $x + y$ is also finite.

Exercise 3.7.23. Suppose $x, y \in \mathbb{S}$ are both finite. Prove xy is also finite.

Exercise 3.7.24. Suppose $x \in \mathbb{S}$ is infinitesimal. Prove x is finite.

Exercise 3.7.25. Suppose $x, y \in \mathbb{S}$ and that x is finite and y is infinitesimal. Prove xy is infinitesimal.

Exercise 3.7.26. Suppose $x \in \mathbb{S}$ is finite and $y \in \mathbb{S}$ is infinitesimal. Prove $x+y$ is finite.

Exercise 3.7.27. Suppose $x \in \mathbb{S}$ is infinite and $y \in \mathbb{S}$ is finite. Prove $x + y$ is infinite.

Exercise 3.7.28. Suppose $x, y, z \in \mathbb{S}$ and $x \approx y$ and z is finite. Prove $xz \approx yz$.

Definition 3.13. *For $x, y \in \mathbb{S}$ we write $x \sim y$ if $x - y$ is finite.*

Exercise 3.7.29. Suppose $x \in \mathbb{S}$. Prove $x \sim x$.

Exercise 3.7.30. Suppose $x, y \in \mathbb{S}$ and $x \sim y$. Prove $y \sim x$.

Exercise 3.7.31. Suppose $x, y, z \in \mathbb{S}$ and $x \sim y$ and $y \sim z$. Prove $x \sim z$.

Exercise 3.7.32. Suppose $x, y, z \in \mathbb{S}$ and $x \sim y$. Prove $x + z \sim y + z$.

Exercise 3.7.33. Suppose $x, y, z \in \mathbb{S}$ and $x \sim y$ and z is finite. Prove $xz \sim yz$.

Exercise 3.7.34. Give examples of the following:

(a) Infinite numbers x, y such that $x + y$ is infinite.

(b) Infinite numbers x, y such that $x + y$ is finite.

(c) Infinite numbers x, y such that $x + y$ is infinitesimal.

(d) An infinite x and an infinitesimal y such that xy is finite.

(e) An infinite x and an infinitesimal y such that xy is infinite.

(f) An infinite x and an infinitesimal y such that xy is infinitesimal.

3.8 Unique existence

3.9 Implicit universals and implications

I promise, this is the last time in these notes that I will call you a *bourgeois gentilhomme*. Our rewritten Theorem 1.1 from section 2.8:

Theorem 1.1. *Suppose* $x, y, z \in \mathbb{R}$. *Then*

$$\big((x+1=y) \wedge (x-1=z)\big) \Rightarrow (x^2 = yz+1).$$

can now be rewritten one last time as:

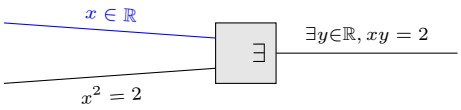
Theorem 1.1 (with $\forall$)**.**

$$\forall x, y, z \in \mathbb{R}, \Big(\big((x+1=y) \wedge (x-1=z)\big) \Rightarrow (x^2 = yz+1)\Big).$$

Once again, the proof of this version is just like that of the previous version, except with a *prove "for all"* at the start (or, technically, three of them) introducing the variables $x, y, z \in \mathbb{R}$. But since we now have all the tools to describe this transformation completely, let's consider a slightly more logically substantial example. Compare the following two theorems and their proofs.

Theorem 3.14. *Suppose* $x \in \mathbb{R}$ *and* $x^2 = 2$. *Then* $\exists y \in \mathbb{R}, xy = 2$.

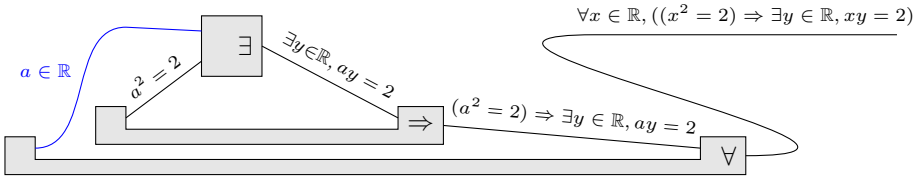
Graphical Proof of 3.14.



□

Theorem 3.15. $\forall x \in \mathbb{R}, ((x^2 = 2) \Rightarrow \exists y \in \mathbb{R}, xy = 2)$.

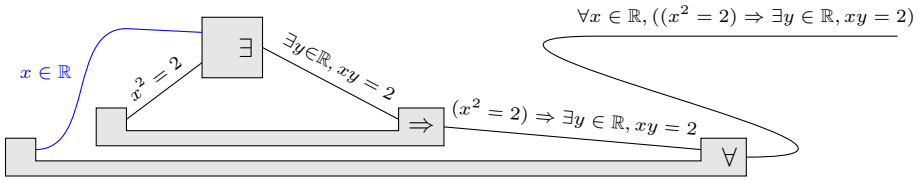
Graphical Proof of 3.15.



□

You can see visually that the "core" of the second proof is exactly the first proof. We can make it look even more similar if we re-use the letter x for the free variable introduced by *prove "for all"* rather than choosing a different variable a; this is potentially dangerous but can be okay as long as there is no other free variable (wire) labeled x.

Better Graphical Proof of 3.15.



□

Now the only difference is that the "suppose" parameters of Theorem 3.14 have been moved into a $\forall$ and an $\Rightarrow$ in Theorem 3.15, so that in the proof we have to introduce them with *prove "for all"* and *prove "if-then"*. In fact it is *always* possible to do this with the "suppose" parameters of a theorem. (As noted in section 2.8, if there is more than one assumed fact, we usually combine them with $\wedge$, so that the proof will start with a *use "and"* as well.)

Importantly, Theorem 3.15 has an advantage over Theorem 3.14: it is more evident how to *use* it in proving something else. Namely, we can apply *use "for all"* to specify a value of x (such as, for instance, $-\sqrt{2}$), and then apply *use "if-then"* by giving a proof that this value of x satisfies $x^2 = 2$ (e.g. by simple algebra), and obtain the resulting conclusion (such as $\exists y \in \mathbb{R}, (-\sqrt{2})y = 2$). (For theorems with multiple hypotheses, we would also use a *prove "and"* here.)

In particular, all the "facts" we mentioned in section 2.6 surrounding Proof Guidance 3 are actually *theorems* that someone proved, which are stated with $\forall$ and $\Rightarrow$. For instance, one of them is $\forall x \in \mathbb{N}, (x = 0) \vee (x \geq 1)$. The principle mentioned there, that we use a fact by substituting arbitrary expressions for the variables in it, can now be seen as just an instance of the *use "for all"* rule.

For these reasons, *we will henceforth regard theorems like Theorem 3.14 as implicit abbreviations for the corresponding theorem like Theorem 3.15, with all the parameters shifted into $\forall$s and $\Rightarrow$s.* This provides the wider context for "instances of theorems" that I promised in section 1.2: a theorem involving supposed variables is actually an implicit abbreviation for a $\forall$-statement, which explains why it makes a claim about *all* values of those variables. It likewise explains why I said there that *an example is not a proof*, since the *prove "for all"* rule requires us to introduce an arbitrary new variable and prove the statement about it, unlike the *use "for all"* and *prove "exists"* rules that allow us to specify a particular value.[14]

We will continue to feel free to write simpler proofs like our above proof of Theorem 3.14, *without* explicitly writing the *prove "for all"* and *prove "if-then"* rules that surround the whole thing, regardless of whether the theorem is stated with "suppose" or "for all" and "if-then". However, we will remember that these logical operators and proof rules are nevertheless "officially" there, and sometimes this will be helpful.

[14]Sometimes people refer to the *prove "exists"* rule as "a situation in which an example *is* a proof". I can see their point, although I don't like to use language that way myself. To me, the value that you specify in the *prove "exists"* rule is not an *example* but rather a "witness". An *example* is, by definition, an instance of a $\forall$ statement.

Exercises

Congratulations! You're done with Olorin's "Quantifier World". Now is a good time to make sure you can solve all the levels in "Proposition World" on the Master difficulty setting, and all the levels in "Advanced Proposition World" on the Adept difficulty setting. On to Negation World!

Chapter 4

Negation and contradiction

Finally, we discuss the last basic logical operator, "not". Unlike the other connectives discussed in chapter 2, this is a "unary" operator, acting on a single statement and simply switching truth values: not-true is false, and not-false is true. We write "not P" as $\neg P$, using a minus-sign with a little hook at the end.[1] Its truth table is thus

P	$\neg P$
$\top$	$\bot$
$\bot$	$\top$

This has the important consequence that, like algebraic negation in ordinary arithmetic, $\neg$ is *involutive*:

$$\neg\neg P = P.$$

This is called the *law of double negation*, and is the source of much of the subtlety around negation.

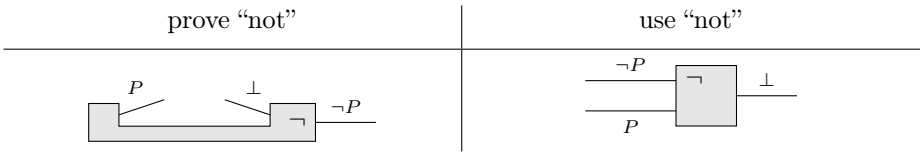
Syntactically, $\neg$ "binds tighter" than any other logical operator. Thus, $\neg P \vee Q$ means $(\neg P) \vee Q$, and $\neg P \Rightarrow Q$ means $(\neg P) \Rightarrow Q$. If you want to negate a compound statement, use parentheses as in $\neg(P \vee Q)$.

4.1 Not ($\neg$)

How do we prove that something is false? The most general answer is: by proving that it cannot possibly be true, i.e. that *if it were true* then something clearly impossible would happen. What is "clearly impossible"? Well, the most basic thing that's clearly impossible is for something else to be both true and false at the same time; we call this a *contradiction*.

We already have a symbol that means "something impossible": $\bot$. So if we use that to mediate these two rules for $\neg$, we get the following graphical representations:

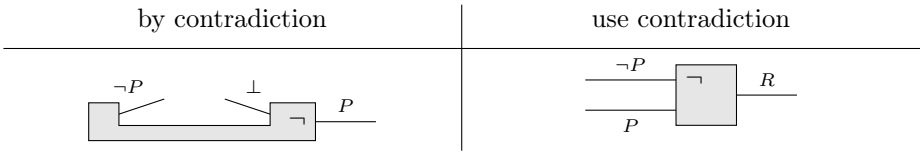
[1] Some people write it instead as $\sim P$ or as $\overline{P}$. Programming languages often denote it `!P`.



That is, to prove $\neg P$, we assume hypothetically that P is true and deduce an impossibility; and to use $\neg P$, if we also know or can prove that P, we get an impossibility. (Note that when phrased this way, the rules for $\neg P$ are the same as the rules for $P \Rightarrow \bot$.)

However, there are more general versions of both of these rules that result from combining them with other principles. Firstly, recall from section 2.10 that the *use "false"* rule says that if we have $\bot$, we can deduce any other statement, which we think of as "closing off" some case of a proof as being impossible because it leads to something impossible. Since this is the only way we can ever use $\bot$, we often combine it with the *use "not"* rule, and call the result the *use contradiction* rule.

Secondly, and even more importantly, since as noted above any statement P is equivalent to $\neg\neg P$, we can actually use the *prove "not"* rule to prove *anything at all*, by first replacing that goal by its double-negation. In this form, we call it a *proof by contradiction*. Thus, the rules we will actually use in practice are:

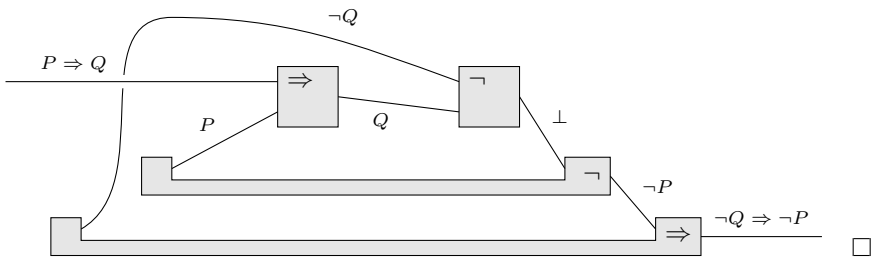


In words, if we have a contradiction, we can formally "deduce anything" and thereby close off any case in which that happens. And we can prove *anything* "by contradiction": just assume the opposite of the goal and deduce a contradiction.

We start with an example of the easier situation in which the goal *is* of the form $\neg P$. In this case, proof by contradiction is suggested by Proof Guidance 1.

Theorem 4.1 (Olorin 4-1-5)**.** *Suppose $P \Rightarrow Q$. Then $\neg Q \Rightarrow \neg P$.*

Graphical Proof of 4.1.



□

Note that the *by contradiction* block appears *inside* the *prove "if-then"* block, as Proof Guidance 1 instructs us: first we see the "if-then" statement $\neg Q \Rightarrow \neg P$

in our goal and use *prove "if-then"*, then we see the "not" statement $\neg P$ in our goal and use *prove "not"*.

To write this proof in English, we need a principle for proofs by contradiction.

Principle of English Proof 8 (By contradiction)**.** When doing a proof by contradiction, introduce the assumption with a phrase like "assume for contradiction that ...". Once you've reached a contradiction, you can simply say "this is a contradiction", remind the reader what it contradicts, or you can emphasize the result, as in "this is a contradiction, so it must be that ...".

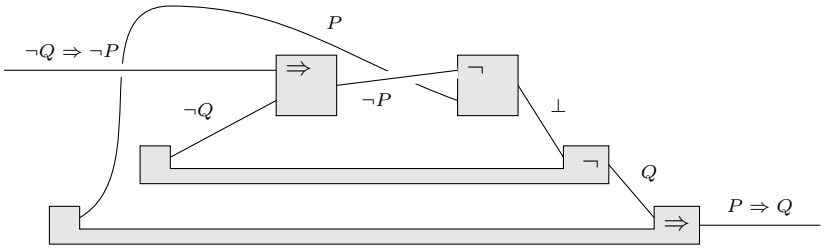
English Proof of 4.1. Assume $\neg Q$, and assume for contradiction that P is true. Since $P \Rightarrow Q$ and P, we have Q. But this contradicts our assumption that $\neg Q$, so it must be that $\neg P$. □

We can also prove the converse of this theorem, but now we have to "guess" that a proof by contradiction will be useful.

Proof Guidance 8 (By Contradiction)**.** If you are stuck, try a proof by contradiction: assume the opposite of the goal and try to deduce a contradiction. You can do this at any point in a proof: at the beginning or in any case or subproof, whatever the current goal is. It is powerful because it gives you a new assumption for free: the negation of the goal. But it is tricky to use because it negates Proof Guidance 1: the logical structure of the goal can't tell you when to do a proof by contradiction, and after you've done it, the goal is now $\bot$ so it has no more logical structure to help you.

Theorem 4.2 (Olorin 4-4-2)**.** *Suppose* $\neg Q \Rightarrow \neg P$*. Then* $P \Rightarrow Q$*.*

Graphical Proof of 4.2.



□

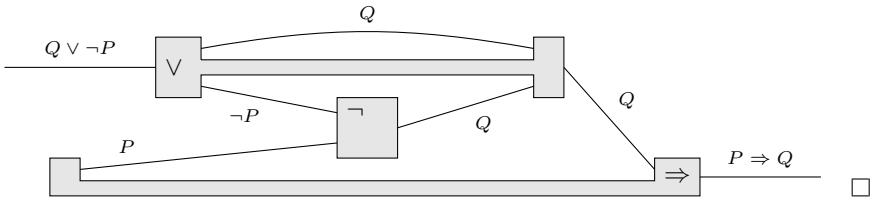
This proof has almost exactly the same structure as the previous one: the only difference is that the input wires of the *use "not"* block come in the other order. But this time, when constructing the proof, while Proof Guidance 1 again applies to $P \Rightarrow Q$, at that point we have a goal of Q and it gives us no help. But now Proof Guidance 8 saves the day: we can prove *anything* by contradiction, so we try assuming $\neg Q$ and see where it leads us. In this case we luck out, because we have an assumption of $\neg Q \Rightarrow \neg P$, so we can get $\neg P$ and thereby a contradiction.

English Proof of 4.2. Assume P, and assume for contradiction that $\neg Q$. Since $\neg Q \Rightarrow \neg P$ and $\neg Q$, we have $\neg P$. But this contradicts our assumption that P, so it must be that Q. □

Finally, here is an example of how the *use contradiction* rule allows us to close off a case.

Theorem 4.3 (Olorin 4-2-1)**.** *Suppose $Q \vee \neg P$. Then $P \Rightarrow Q$.*

Graphical Proof of 4.3.



□

English Proof of 4.3. Assume P. Since $Q \vee \neg P$, we have two cases.
Case 1: Assume Q. Then Q.
Case 2: Assume $\neg P$. This contradicts our assumption that P, so this case is impossible. □

Exercises

As usual, write your proofs both graphically and in English. This section pertains to all of Olorin's "Negation World": Stages 4-1, 4-2, 4-3, and 4-4.

Exercise 4.1.1 (Olorin 4-2-2)**.** Assume $\neg(P \vee Q)$. Prove $\neg P \wedge \neg Q$.

Exercise 4.1.2 (Olorin 4-2-3)**.** Assume $\neg P \wedge \neg Q$. Prove $\neg(P \vee Q)$.

Exercise 4.1.3 (Olorin 4-2-4)**.** Assume $\neg P \vee \neg Q$. Prove $\neg(P \wedge Q)$.

Exercise 4.1.4 (Olorin 4-3-1)**.** Assume $\neg\exists x \in A, P(x)$. Prove $\forall x \in A, \neg P(x)$.

Exercise 4.1.5 (Olorin 4-3-2)**.** Assume $\forall x \in A, \neg P(x)$. Prove $\neg\exists x \in A, P(x)$.

Exercise 4.1.6 (Olorin 4-3-3)**.** Assume $\exists x \in A, \neg P(x)$. Prove $\neg\forall x \in A, P(x)$.

Exercise 4.1.7 (Olorin 4-2-5)**.** Assume $P \wedge \neg Q$. Prove $\neg(P \Rightarrow Q)$.

Exercise 4.1.8 (Olorin 4-2-6)**.** Assume $P \vee Q$ and $\neg P$. Prove Q.
(This is called the *disjunctive syllogism*; we will return to it in section 4.2.)

Exercise 4.1.9 (Olorin 4-3-4)**.** Suppose $(\exists x \in A, \neg P(x)) \vee Q$.
Prove $(\forall x \in A, P(x)) \Rightarrow Q$.

Exercise 4.1.10 (Olorin 4-3-5)**.** Suppose $(\exists x \in A, \neg P(x)) \vee Q$ and $\exists x \in A, \top$.
Prove $\exists x \in A, (P(x) \Rightarrow Q)$.

This is a good point at which to go back and start re-doing the levels from "Advanced Proposition World" on the Master difficulty setting, and those from "Quantifier World" on the Adept difficulty setting.

The remaining exercises in "Negation World" are tricky! However, they are tricky in similar ways, so once you've done one or two of them you should find the others easier.

Exercise 4.1.11 (Olorin 4-4-3)**.** Assume $\neg(P \Rightarrow Q)$. Prove $P \wedge \neg Q$.

Exercise 4.1.12 (Olorin 4-4-4)**.** Assume $\neg(P \wedge Q)$. Prove $\neg P \vee \neg Q$.

Exercise 4.1.13 (Olorin 4-4-5)**.** Assume $\neg\forall x \in A, P(x)$. Prove $\exists x \in A, \neg P(x)$.

Exercise 4.1.14 (Olorin 4-4-6)**.** Assume $P \Rightarrow Q$. Prove $Q \vee \neg P$.

Exercise 4.1.15 (Olorin 4-4-7)**.** Prove $P \vee \neg P$. (This is called the *law of excluded middle*.)

Exercise 4.1.16 (Olorin 4-5-1)**.** Prove $((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$.

Exercise 4.1.17 (Olorin 4-5-2)**.** Prove $(P \Rightarrow Q) \vee (Q \Rightarrow P)$.

Exercise 4.1.18 (Olorin 4-5-3)**.** Suppose $P \Rightarrow Q$ and $R \Rightarrow S$.
Prove $(P \Rightarrow S) \vee (R \Rightarrow Q)$.

Exercise 4.1.19 (Olorin 4-5-4)**.** Suppose $(P \wedge Q) \Rightarrow R$.
Prove $(P \Rightarrow R) \vee (Q \Rightarrow R)$.

Exercise 4.1.20 (Olorin 4-5-5)**.** Prove $(\neg P \Rightarrow P) \Rightarrow P$. (This is known as *consequentia mirabilis*.)

Exercise 4.1.21 (Olorin 4-5-6)**.** Prove $(P \Rightarrow Q) \vee (Q \Rightarrow R)$.

Exercise 4.1.22 (Olorin 4-5-7)**.** Suppose $\exists x \in A, \top$.
Prove $\exists x \in A, (P(x) \Rightarrow \forall y \in A, P(y))$.
(This is called the *drinker paradox*: if A is the collection of people in a pub, and $P(x) =$ "x is drinking", then this is the statement "there exists a person in the pub such that if that person is drinking, then everyone is drinking". Of course, it is only true when the pub isn't empty.)

4.2 The algebra of negation

Exercises 4.1.1, 4.1.2, 4.1.3 and 4.1.12 show that $\neg$ "distributes" over $\wedge$ and $\vee$ in a sense, but with a twist: when we "push it inside", $\wedge$ and $\vee$ flip.

$$\neg(P \wedge Q) = \neg P \vee \neg Q$$
$$\neg(P \vee Q) = \neg P \wedge \neg Q$$

Similarly, Exercises 4.1.4, 4.1.5, 4.1.6 and 4.1.13 show that $\neg$ "pushes inside" quantifiers, flipping $\exists$ and $\forall$:

$$\neg(\forall x \in A, P(x)) = \exists x \in A, \neg P(x)$$
$$\neg(\exists x \in A, P(x)) = \forall x \in A, \neg P(x)$$

Finally, Exercises 4.1.7 and 4.1.11 give a rule for the negation of an implication:

$$\neg(P \Rightarrow Q) = P \wedge \neg Q.$$

We refer to these rules for "computing negations" collectively as *De Morgan's laws*, although arguably that name properly refers only to the first two. These rules are possibly the most important of all the "algebraic" laws satisfied by the logical operators. You *must* memorize them.

However, memorizing them should *not* be too onerous, because they should also be intuitive. For example, for $P \wedge Q$ to be false, it suffices for *at least one* of P and Q to be false; while for $P \vee Q$ to be false, it must be that *both* P and Q are false. Similarly, for $\forall x, P(x)$ to be false, there must be some x for which $P(x)$ is false (a *counterexample* — recall section 1.2); and for $\exists x, P(x)$ to be false, there cannot be any x such that $P(x)$ is true, hence all $P(x)$ must be false. And the rule for $\neg(P \Rightarrow Q)$ aligns with our conclusion from Connor's con game in section 2.7, that the only way for $P \Rightarrow Q$ to be false (i.e. to catch Connor in a lie) is if P is true and Q is false.

Note that in the De Morgan's laws for quantifiers, the symbol $\in$ does not get negated. Intuitively, we can say this is because the $x \in A$ is not a *statement*, but a label or "typing declaration" that says what *kind of thing* the bound variable x denotes, and x still denotes the same *kind* of thing regardless of what we are saying about how many such x's do or don't satisfy some property. In fact, since the inner statement $P(x)$ in a quantified statement like $\forall x \in A, P(x)$ generally *only makes sense* when $x \in A$, it doesn't even make *sense* to negate the typing declaration. For instance, what would it mean to say $\forall x \notin \mathbb{R}, (x^2 < 0)$, that is "for every x that is *not* a real number, $x^2 < 0$"? Could x be a complex number? A function? A graph? A panda? In none of these cases does "$x^2 < 0$" make sense; so $\forall x \notin \mathbb{R}, (x^2 < 0)$ doesn't even typecheck (see section 1.5) and cannot be the negation of any statement.

There are some other remarks we should make about how $\neg$ relates to the other operators. One is that once we have $\neg$, given an implication statement $P \Rightarrow Q$ we can now associate to it *three* related statements:

- The *converse* of $P \Rightarrow Q$ is $Q \Rightarrow P$. (We encountered this already in section 2.9.)
- The *inverse* of $P \Rightarrow Q$ is $\neg P \Rightarrow \neg Q$.
- The *contrapositive* of $P \Rightarrow Q$ is $\neg Q \Rightarrow \neg P$.

Continuing our example from section 2.9 where P is "it rained last night" and Q is "the grass is wet", we observe that:

- As noted in section 2.9, the converse $Q \Rightarrow P$ claims that "if the grass is wet, then it rained last night". This is *not necessarily true*: the grass could have gotten wet in other ways, for instance maybe the sprinklers ran this morning.

- The inverse $\neg P \Rightarrow \neg Q$ claims that "if it didn't rain last night, then the grass isn't wet." This is also not necessarily true, for the same reason that the converse is not necessarily true: the sprinklers could have run. Importantly, the inverse is also *not* the same as the negation $\neg(P \Rightarrow Q)$; as we have seen, that is $P \wedge \neg Q$.

- The contrapositive $\neg Q \Rightarrow \neg P$ is "if the grass isn't wet, then it didn't rain last night." This *is* necessarily true (assuming the original statement $P \Rightarrow Q$ was true). Since the original statement is the contrapositive of its contrapositive, the contrapositive has the *same truth value* as the original statement. We also proved this equivalence in Theorems 4.1 and 4.2.

The equivalence or non-equivalence of these statements can also be seen from our discussion of Connor's con game in section 2.7. Recall the game:

> Connor the Con Man has a deck of cards, each of which has a letter on one side and a digit on the other. He puts four cards from this deck on the table:
>
> 
>
>
> "If any of these four cards has a vowel on one side, then it has an even number on the other side," says Connor. Which of the cards do you need to turn over to check whether he's telling the truth?

If P = "this card has a vowel on one side" and Q = "this card has an even number on one side", then each of these cards corresponds to one of the four statements

$$\neg P \Rightarrow \neg Q \qquad P \Rightarrow Q \qquad Q \Rightarrow P \qquad \neg Q \Rightarrow \neg P.$$

In each case, the antecedent ("if" part) of the statement represents the visible side of the card, while the consequent ("then" part) represents the hidden side, and asking whether we need to turn that card over is the same as asking whether the implication is equivalent to Connor's original statement $P \Rightarrow Q$. Specifically:

- The card $\boxed{\text{E}}$ that we obviously *do* have to turn over corresponds to the original statement $P \Rightarrow Q$, since we can see that P is true and we must check whether Q is true. And of course, $P \Rightarrow Q$ *is* equivalent to itself.

- The card $\boxed{\text{K}}$ that we obviously *don't* have to turn over corresponds to the inverse $\neg P \Rightarrow \neg Q$, so that is *not* equivalent to Connor's statement.

- The card $\boxed{7}$ that we less obviously *do* have to turn over corresponds to the contrapositive $\neg Q \Rightarrow \neg P$, so that *is* equivalent to Connor's statement.
- The card $\boxed{4}$ that we least obviously *don't* have to turn over corresponds to the converse $Q \Rightarrow P$, so that is *not* equivalent to Connor's statement.

It's also worth noting that the inverse is the contrapositive of the converse. Thus, the converse and the inverse also have the same truth value. A particular application of this is that when proving $P \Leftrightarrow Q$, that is, a statement $P \Rightarrow Q$ and also its converse $Q \Rightarrow P$, it is equivalent to prove the statement $P \Rightarrow Q$ and also its *inverse* $\neg P \Rightarrow \neg Q$.

Finally, using $\neg$ we can define other logical operators in terms of our basic ones. For example:

- The *exclusive or*, which we write as $P \oplus Q$, is true if one of P and Q is true and the other is false. This can be defined as

$$\begin{aligned} P \oplus Q &= (P \vee Q) \wedge (\neg P \vee \neg Q) \qquad \text{or as} \\ P \oplus Q &= (P \wedge \neg Q) \vee (\neg P \wedge Q). \end{aligned}$$

  It is worth noting that $\Leftrightarrow$ and $\oplus$ are each other's negations:

$$\neg(P \oplus Q) = (P \Leftrightarrow Q) \qquad \neg(P \Leftrightarrow Q) = P \oplus Q.$$

- The *nand* can be defined as $P \uparrow Q = \neg(P \wedge Q) = \neg P \vee \neg Q$.
- The *nor* can be defined as $P \downarrow Q = \neg(P \vee Q) = \neg P \wedge \neg Q$.

The operators $\oplus$, $\uparrow$, and $\downarrow$ do not play a very significant role in mathematics, but are occasionally used in computer science and electrical engineering.

In fact, once we have $\neg$, any of our so-called basic operators $\wedge, \vee, \Rightarrow$ could also be defined in terms of any of the others:

$$\begin{aligned} P \wedge Q &= \neg(\neg P \vee \neg Q) & P \wedge Q &= \neg(P \Rightarrow \neg Q) = \neg(Q \Rightarrow \neg P) \\ P \vee Q &= \neg(\neg P \wedge \neg Q) & P \vee Q &= (\neg P \Rightarrow Q) = (\neg Q \Rightarrow P) \\ P \Rightarrow Q &= \neg P \vee Q & P \Rightarrow Q &= \neg(P \wedge \neg Q) \end{aligned}$$

Nevertheless, we treat $\wedge, \vee, \Rightarrow$ as basic, and others such as $\Leftrightarrow$ and $\oplus$ as "derived", because each of $\wedge, \vee, \Rightarrow$ has its own *proof rules*, whereas to prove or use $\Leftrightarrow$ or $\oplus$ we always just use its definition in terms of our basic operators. However, these equivalences can sometimes be useful in proofs:

Proof Guidance 9 (Use a logical equivalence)**.** If you don't see a productive way to use the logical structure of some given or goal (as in Proof Guidance 1), you can try replacing it with a different statement that's logically equivalent.

The most common example is to use De Morgan's laws to "push $\neg$ inside". This is usually done immediately after proof by contradiction to simplify the

new assumption and expose more logical structure to make Proof Guidance 1 applicable to it.

The next most common example is to replace an implication by its contrapositive. Doing this to the goal followed by *prove "if-then"* is called *proof by contrapositive*; doing this to a given followed by *use "if-then"* is called (by afficionados of fancy Latin phrases) *modus tollens*. Proof by contrapositive is very similar to proof by contradiction, and it's often just a matter of taste which you prefer in a given case.

The next next most common example is to replace $P \vee Q$ by $\neg P \Rightarrow Q$ or $\neg Q \Rightarrow P$. Doing this to to a given followed by *use "if-then"* is called the *disjunctive syllogism*. Doing this to the *goal* followed by *prove "if-then"* seems equally important to me, but it doesn't seem to have a standard name.

Of course, using Proof Guidance 9 would make most of the exercises in section 4.1 trivial. So don't use it there. But we will use it freely going forward from this point.

As an example, we sketch a proof of the *Pigeonhole Principle*.[2] Like many other so-called "proof strategies", this is *not* a fundamental rule of proof, but simply a *theorem* that can be proven once, and then used thereafter in other proofs with *use "for all"* and *use "if-then"*.

Theorem 4.4 (The Pigeonhole Principle)**.** *For any positive integers m, n, and p, if $m > pn$, and m objects are divided into n groups, then at least one group must have at least $p + 1$ objects in it.*

To give a *completely* formal proof of this would require making precise what exactly it means to have "m objects" and "divide them into n groups". This is the province of set theory, which is beyond the scope of these notes. However, if we assume that these notions behave in an intuitive way, we can give a straightforward proof of the theorem.

Proof of 4.4. Suppose m and n are positive integers with $m > pn$, and that m objects are divided into n groups. Number the groups of objects $1, 2, \ldots, n$, and let $f(k)$ be the number of elements in the k^{th} group. We must prove that $\exists k, f(k) \geq p+1$. Assume for contradiction the opposite of this, $\neg\exists k, f(k) \geq p+1$, or equivalently $\forall k, \neg(f(k) \geq p + 1)$, which is the same as $\forall k, f(k) \leq p$.

Now, since there were m objects to begin with, we have $m = \sum_{k=1}^{n} f(k)$. But by assumption, each $f(k) \leq p$, and therefore

$$m = \sum_{k=1}^{n} f(k) \leq \sum_{k=1}^{n} p = pn.$$

Thus $m \leq pn$, contradicting our assumption that $m > pn$. □

[2] Contrary to popular belief, the word "pigeonhole" in the name of this theorem does not refer directly to putting *pigeons* into holes, but only at one remove. Rather, a *pigeonhole* is a small cubby or mailbox used to deposit letters or messages, so-named because it is roughly the *size and shape* of a hole where a pigeon might nest (which is actually called a "dovecote").

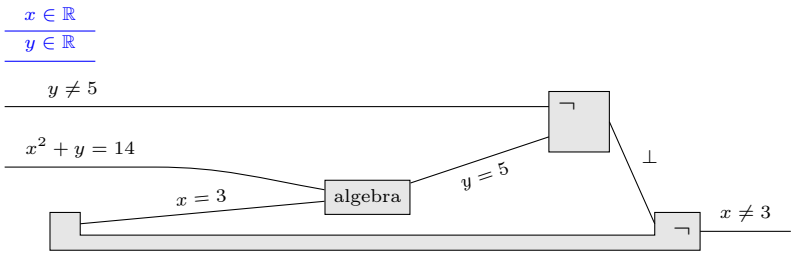
4.3 ¬ with algebra

As in chapters 2 and 3, we now combine the rules of ¬ (and the other rules of logic) with the rules of algebra. When working with equalities, we have a more familiar notation for negations: $\neg(x = y)$ is usually written $x \neq y$. And, of course, it follows that $\neg(x \neq y)$ is the same as $x = y$. Similarly, for inequalities, $\neg(x < y)$ is the same as $x \geq y$, while $\neg(x \leq y)$ is the same as $x > y$.[3] We also often abbreviate the negations of other statements and relations with a slash: for instance, $\nexists x \in A, P(x)$ means $\neg\exists x \in A, P(x)$ (although $\not\forall$ is rarely used, perhaps for aesthetic reasons), and $a \nmid b$ means $\neg(a \mid b)$.

Here's a simple example of a proof using disequalities.

Theorem 4.5. *Suppose x and y are real numbers, $y \neq 5$, and $x^2 + y = 14$. Then $x \neq 3$.*

Graphical Proof of 4.5.



Algebra: Substituting $x = 3$, we get

$$\begin{aligned} x^2 + y &= 14 \\ 9 + y &= 14 \\ y &= 5. \end{aligned}$$

□

English Proof of 4.5. Suppose for contradiction that $x = 3$. Substituting this in the assumption $x^2 + y = 14$, we get

$$\begin{aligned} x^2 + y &= 14 \\ 9 + y &= 14 \\ y &= 5. \end{aligned}$$

This contradicts the assumption that $y \neq 5$, so it must be that $x \neq 3$. □

Next we consider an example where De Morgan's laws are useful. Here we omit the graphical proof and go right to the English one.

Theorem 4.6. *Prove that for any real number x, if $x^2 = x$, then either $x = 0$ or $x = 1$.*

[3] This is true for all our ordered number systems, but not in a more general partial order; see footnote 10 on page 50.

This may seem obvious, but it requires proof! Also, we have stated this theorem with an explicit ∀ and ⇒, but in line with our discussion in section 3.9 we don't mention the corresponding *prove* rules in the proof.

English Proof of 4.6. Suppose for contradiction that $x \neq 0$ and $x \neq 1$. Since $x \neq 0$, we can divide both sides of the assumption $x^2 = x$ by x, getting $x = 1$. But this contradicts $x \neq 1$. □

Note that in the first sentence we used De Morgan's law without even mentioning it. The direct assumption that proof by contradiction would give us is $\neg(x = 0 \vee x = 1)$, but we immediately replaced this with $x \neq 0 \wedge x \neq 1$ (and also immediately used *use "and"* without mentioning it, as usual). This proof also serves as a good reminder that we can only divide by things that are nonzero (and only in number systems where division is allowed).

A classic application of proof by contradiction in algebra is the proof that certain numbers are irrational. An *irrational number* is a real number that is *not* a rational number; thus, usually to prove that a number is irrational we assume it is rational and derive a contradiction.

Theorem 4.7. $\sqrt{2}$ *is irrational.*

In the following proof, we will use the fact that if x is a rational number, we can write $x = \frac{p}{q}$ where p, q are integers with no common factor greater than 1 (a fraction in "lowest terms"). This can be proven by starting with an arbitrary fraction $\frac{a}{b}$ and dividing the top and bottom by $\gcd(a, b)$.

English Proof of 4.7. Suppose for contradiction that $\sqrt{2}$ is rational. Then there exist integers p, q with no common factor greater than 1 such that $\sqrt{2} = \frac{p}{q}$. Therefore,

$$\begin{aligned} \sqrt{2} &= \tfrac{p}{q} \\ (\sqrt{2})^2 &= (\tfrac{p}{q})^2 \\ 2 &= \tfrac{p^2}{q^2} \\ 2q^2 &= p^2. \end{aligned}$$

Therefore, $p^2 \equiv_2 0$. By Exercise 5.3.8, $p \equiv_2 0$. Therefore, there exists an integer a such that $p = 2a$. Now we have

$$\begin{aligned} 2q^2 &= (2a)^2 \\ 2q^2 &= 4a^2 \\ q^2 &= 2a^2. \end{aligned}$$

Therefore, $q^2 \equiv_2 0$. So by Exercise 5.3.8, $q \equiv_2 0$, and there exists an integer b such that $q = 2b$. But then p and q have the common factor 2, contradicting our assumption that $\frac{p}{q}$ is in lowest terms. □

Exercises

Exercise 4.3.1. Suppose $x, y \in \mathbb{R}$, that $y + x = 2y - x$, and that x and y are not both zero. *(Warning! This means $\neg(x = 0 \wedge y = 0)$.)* Prove $y \neq 0$.

Exercise 4.3.2. Suppose p and q are positive real numbers and $\sqrt{pq} \neq \frac{1}{2}(p+q)$. Prove $p \neq q$.

Exercise 4.3.3. Prove that there do not exist positive real numbers x and y such that $x \neq y$ and $x^2 - y^2 = 0$.

Exercise 4.3.4. Prove that for any $x \in \mathbb{R}$, if $x^2 \neq 1$, then $x \neq 1$ and $x \neq -1$.

Exercise 4.3.5. Prove that for any $x, y \in \mathbb{R}$, if $3x^2 + xy < 0$, then either $x > 0$ or $y > 0$.

Exercise 4.3.6. Prove that for any $x, y \in \mathbb{R}$, if $x^2 - 2xy + y^2 \neq 0$, then $x \neq y$.

Exercise 4.3.7. Prove that for any $x, y \in \mathbb{R}$, if $x^2 - y^2 \neq 0$, then $x \neq y$ and $x \neq -y$.

Exercise 4.3.8. Prove that for any $x, y \in \mathbb{R}$, if $x^2 + y^2 \neq 2x + 2y - 2$, then $x \neq 1$ or $y \neq 1$.

Exercise 4.3.9. Prove that for any $x, y \in \mathbb{R}$, if $x^2y - 2y = 4y + xy$, then $x \neq 3$ and $y \neq 0$.

Exercise 4.3.10. Prove that for any real number x, if $x^2 = 0$, then $x = 0$.

Exercise 4.3.11. Prove that $\sqrt{3}$ is irrational. *(You'll need an analogue of Exercise 5.3.8 for divisibility by 3.)*

Exercise 4.3.12. It's a fact that for any positive real numbers a, b, there is a real number $\log_b a$ with the property that $b^{\log_b a} = a$. Prove that $\log_2 3$ is irrational.

Exercise 4.3.13. Suppose x is irrational and y is rational. Prove $x + y$ is irrational.

Exercise 4.3.14. Suppose x is irrational and y is rational and nonzero. Prove xy is irrational.

Exercise 4.3.15. Prove that there does not exist a positive real number x such that for every positive real number y we have $x \leq y$.

4.4 Constructivity

I invite you to ponder the following proof.

Theorem 4.8. *There exist irrational numbers a and b such that a^b is rational.*

Proof of 4.8. Assume for contradiction that for all irrational numbers a and b, we have that a^b is irrational. Therefore, in particular letting $a = \sqrt{2}$ and $b = \sqrt{2}$, we have that $\sqrt{2}^{\sqrt{2}}$ is irrational. So now we can let $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$ in the assumption, hence $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}}$ is also irrational. But

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = (\sqrt{2})^{\sqrt{2}\cdot\sqrt{2}} = (\sqrt{2})^2 = 2,$$

which is rational, a contradiction. Thus, there must exist irrational numbers a and b such that a^b is rational. □

This is a perfectly good proof. However, it proves that *there exist* some numbers a and b *without giving us any idea what those numbers are*. That is, it shows that an alternate world in which there do *not* exist such numbers is impossible, but it doesn't tell us anything about what the numbers that actually exist in *our* world are.

A proof like this is called a *nonconstructive* proof, and many mathematicians find it somewhat unsatisfying. It's better to give a constructive proof if you can, because it's more informative.

In fact, there are constructive proofs of Theorem 4.8.

Constructive Proof of 4.8. Let $a = \sqrt{2}$ and $b = 2\log_2 3$, which are irrational by Theorem 4.7 and Exercise 4.3.12. Then

$$a^b = (\sqrt{2})^{2\log_2 3} = (2^{\frac{1}{2}})^{2\log_2 3} = 2^{\frac{1}{2}\cdot 2\log_2 3} = 2^{\log_2 3} = 3,$$

which is rational. □

However, there are other theorems that have *no* constructive proof. For example, the *Intermediate Value Theorem* from calculus:

Theorem 4.9. *For any continuous function f with domain $[a, b]$ and any real number c, if $f(a) < c < f(b)$, then there exists $x \in (a, b)$ such that $f(x) = c$.*

It is *impossible* to give a proof of this theorem that is constructive, in the sense that there is an algorithm to find x given f and c. (Although proving that, as well as proving the theorem itself, is beyond the scope of these notes.)

Interestingly, the rules of negation considered in this chapter are the *only way* to produce nonconstructive proofs. That is, if we stick to the rules of proof from chapters 2 and 3 (and also chapter 5), we are guaranteed that *all* our proofs will automatically be constructive. Although of course, if we did that, then we wouldn't be able to reason about ¬ at all.

There are alternative rules for ¬ that also prevent us from creating nonconstructive proofs, but they are a bit weird. Actually the *rules* aren't weird: they are just the original *prove "not"* and *use "not"* rules from section 4.1 before we modified them. What's weird is that we have to avoid using the law of double negation $\neg\neg P = P$; thus we can only use "proof by contradiction" to prove a

negated statement,[4] and we can't replace an arbitrary statement by a negated one. This makes us also unable to prove various other facts, notably all the "tricky" exercises in section 4.1, including the *law of excluded middle* $P \vee \neg P$. The resulting logic is called *constructive logic*, and as weird as it may seem, it is actually quite useful. In addition to guaranteeing that all proofs automatically come with an algorithm (which is obviously of interest to computer scientists), there are many "nonclassical models" where constructive logic works but ordinary logic doesn't. It's beyond the scope of these notes to say any more about constructive logic; but if you're interested in it, I recommend the article *Five stages of accepting constructive mathematics* by Andrej Bauer [Bau16].

I end by inviting you to also ponder the following nonconstructive proof.

Theorem 4.10. *There exists a computer program that will terminate in less than a year, and print "*`yes`*" if there is intelligent extraterrestrial life somewhere in the universe, and "*`no`*" if there is not.*

Proof of 4.10. By the law of excluded middle, there are two cases:
Case 1: Assume there is intelligent extraterrestrial life somewhere in the universe. Then the program is `print "yes"`.
Case 2: Assume there is not intelligent extraterrestrial life somewhere in the universe. Then the program is `print "no"`. □

Exercises

Congratulations! You're done with Olorin's "Negation World", and thereby with all the worlds it has to offer at this time. This is a good time to go back and finish solving all the levels from previous worlds on the Master difficulty setting. After you've done that, come back to Negation World and solve its levels on Adept and Master difficulty. Then you will truly be a Master of Predicate Logic!

There's one remaining basic proof rule, but unfortunately Olorin doesn't yet handle it, since unlike all the other rules it pertains specifically to numbers. This rule is *mathematical induction*.

[4] In fact, constructive mathematicians generally reserve the phrase "proof by contradiction" for the case where the statement being proven is *not* negated, so that we *have* to use double-negation to apply the rule. They refer to the ordinary use of *prove "not"* as a *proof of negation*. With this terminology, we can say that "proof by contradiction" is not permitted in constructive mathematics. However, most mathematicians do not make this distinction.

Chapter 5

Recursion and induction

5.1 Recursive definitions

Until now, your main mathematical experience with *functions* has probably been from algebra and calculus, such as

$$f(x) = \sqrt{x-3}.$$

For these kinds of functions, the input and output are both real numbers, although not *every* real number is allowed as an input. For instance, for the function f defined above, the allowed inputs are all real numbers x with $x \geq 3$. The collection of allowed inputs is called the *domain*, and the collection of allowed outputs is called the *codomain*. (The codomain should not be confused with the *range*, which is the collection of outputs that *actually* come out for some actual input.) If the domain of a function f is A and the codomain is B, we write $f : A \to B$. Thus, using the standard interval notation $[3, \infty)$ for the collection of real numbers ≥ 3, for the function f defined above we have $f : [3, \infty) \to \mathbb{R}$.

If you have done any programming, you've probably encountered functions whose inputs and outputs are other kinds of things, like integers, lists, or strings. In this section, we are concerned with functions whose domain is $\mathbb{N}$, the natural numbers. A function like this can also be thought of as a *sequence*

$$f(0), f(1), f(2), f(3), \ldots.$$

For example, the function $f : \mathbb{N} \to \mathbb{N}$ defined by $f(n) = n^2$ corresponds to the sequence of square numbers:

$$0, 1, 4, 9, 16, 25, \ldots.$$

This sequence is defined by a simple formula. But frequently we encounter sequences that are not (or, at least, not *apparently*) defined by a simple formula for $f(n)$ in terms of n only, but rather by a rule that specifies how to compute

each $f(n)$ *assuming that we already know the previous values of* f. For example, the sequence of *Fibonacci numbers*

$$1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

is defined by the rule that the first two numbers are 1, and after that each number is the sum of the *previous two* numbers:

$$\begin{aligned} 2 &= 1 + 1 \\ 3 &= 1 + 2 \\ 5 &= 2 + 3 \\ 8 &= 3 + 5 \\ &\vdots \end{aligned}$$

This can be written as a definition by cases that refers to previous values:

$$F(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ F(n-2) + F(n-1) & \text{if } n > 1 \end{cases}$$

This is called a *recursive definition*: a definition of a function that refers to previous values of the same function. These previous values are called *recursive calls*. Recursive functions are very common in programming, such as the following Python code for computing the Fibonacci sequence:

```
def fibonacci(n):
    if n <= 1:
        return(1)
    else:
        return fibonacci(n-2) + fibonacci(n-1)
```

If you *actually* give Python this definition, then it will evaluate each recursive call the same way it would the original function call, which may involve evaluating more recursive calls, and so on. For example, evaluating $F(4)$ will lead to the tree of calls shown in Figure 5.1 on page 115. This is very inefficient: in Figure 5.1 we already compute $F(2)$ twice, and for larger values of $F(n)$ we'll end up computing $F(2)$ exponentially[1] many times, and similarly for other val-

[1]Here "exponentially" means that there is a real number $a > 1$ such that in evaluating $F(n)$ we have to compute $F(2)$ about a^n times. More generally, to say that anything grows "exponentially" means that it grows as a function of its input x approximately like a^x, for some real number $a > 1$. This is the *only* thing that "exponentially" should mean.

Modern dictionaries, in accord with their mandate of simply describing the way language *is currently* used, record another meaning of "exponential" as "characterized by an extremely rapid increase". But this recent usage is an execrable development that does violence to the very concept of number. Imagine if people started saying "a plus b" to mean "somewhat more than a and b". The whole point of numbers, arithmetic, and mathematics is that we can say *precisely* what something like "addition" means, and it's not just "somewhat more". Similarly, we know *precisely* what "exponential" means, and it's *not* just "increasing quickly".

You might think that we could recover the correct mathematical meaning of "exponential" by instead saying "literally exponential". Unfortunately, some folks have also had the bright idea of misusing "literally" to mean "figuratively". Please, gentle reader: can you not see that this way lies madness?

To evaluate $F(4)$, we have to evaluate $F(2)$ and $F(3)$.

- To evaluate $F(2)$, we have to evaluate $F(0)$ and $F(1)$.
  - $F(0) = 1$ by definition.
  - $F(1) = 1$ by definition.

  Now we can compute $F(2) = 1 + 1 = 2$.
- To evaluate $F(3)$, we have to evaluate $F(1)$ and $F(2)$.
  - $F(1) = 1$ by definition.
  - To evaluate $F(2)$, we have to evaluate $F(0)$ and $F(1)$.
    - $F(0) = 1$ by definition.
    - $F(1) = 1$ by definition.

    Now we can compute $F(2) = 1 + 1 = 2$.

  Now we can compute $F(3) = 1 + 2 = 3$.

Now we can compute $F(4) = 2 + 3 = 5$.

Figure 5.1: Evaluating the 4th Fibonacci number naïvely

ues of F. The better approach is the one we took above: compute the sequence $F(0), F(1), F(2), F(3), \ldots$ in order and save each value as we compute it, so that at each step we can just look up the previous two and add them together. It's easy to program a computer to do this too, although it takes a little more effort; if you do it "as needed" and save the results between separate calls, it's called *memoization*.

However, when speaking about a mathematical *function* (rather than an *algorithm*), the method of evaluation and its efficiency (or lack thereof) are irrelevant. All that matters is the output *value* associated to every input, and for this purpose the simple recursive definition is easiest to work with.

Whichever method of evaluation we use, we can see two aspects of a definition by recursion that are essential for it to make sense:

1. When the computation of $f(n)$ involves a recursive call to $f(k)$, the input k of the recursive call is always an expression that is *smaller than* n.

2. There are one or more "smallest cases" in which there are *no* recursive calls.

The combination of these two properties ensures that $f(n)$ is well-defined for all n. Under the naïve approach, these properties say that any call to the function eventually stops calling itself recursively, since the arguments of recursive calls decrease with every call and hence must eventually end up in one of the smallest cases where there are no more recursive calls. And under the sequence approach,

the smallest cases get the sequence started, and the restriction on recursive calls ensures that all the necessary previous values of f have already been computed by the time we come to compute $f(n)$.

Another important recursively defined function arises in modular arithmetic. We have defined the *relation* $a \equiv_n b$ by $\exists k \in \mathbb{Z}, (a - b = kn)$. For instance, $7 \equiv_3 10$ since $7 - 10 = (-1) \cdot 3$. However, in addition to a relation, each integer a has a canonical "smallest" representative that is congruent to it modulo n, which we call "$a \bmod n$", and we can test whether $a \equiv_n b$ by checking whether these representatives are *equal*. For instance, $7 \bmod 3 = 1$ and $10 \bmod 3 = 1$.

At least when a is a natural number, we can find $a \bmod n$ by subtracting copies of n from a over and over again until we get a result less than n. Now, *whenever you hear "over and over again" in the description of a method or algorithm, you should think immediately of recursion.*[2] To express such a method precisely as a recursive function, try to formulate it as a sufficiently general question so that instead of "do X over and over again" you can say "do X once, and then maybe do the whole thing over again". In this case, if we start with an a that is greater than or equal to n, we can subtract n from it *once* and get a smaller number, and then repeat until we get a number smaller than n. Thus, a recursive definition of $a \bmod n$ is

$$a \bmod n = \begin{cases} a & \text{if } a < n \\ (a - n) \bmod n & \text{if } a \geq n. \end{cases}$$

As noted, this definition depends on a being a *natural number* (i.e. a nonnegative integer). It also requires that n is a *positive* integer, since otherwise $a - n$ wouldn't be less than a. We will lift the first restriction in section 5.3. The second is more controversial: different programming languages have different conventions about the value of $a \bmod n$ when n is negative, and mathematicians rarely use that case at all, so I recommend avoiding it.

Exercises

5.2 Inductive proofs

Although a recursively defined function isn't *given by* a formula, it might still *happen to be equal to* a formula. As a simple example, if we define $g : \mathbb{N} \to \mathbb{N}$ recursively by

$$g(n) = \begin{cases} 0 & \text{if } n = 0 \\ g(n-1) + 2n - 1 & \text{if } n > 0. \end{cases}$$

[2] Another way to achive "over and over again" in a programming language, which may be more familiar to you, is *iteration*, such as a "for" or "while" loop. However, this is much harder to make sense of and analyze mathematically, and many algorithms are more naturally expressed recursively than iteratively. Becoming comfortable with recursion, as an alternative to iteration, is an essential skill for anyone who wants to be more than a novice programmer.

then by computing the first few values of g:

$$0, 1, 4, 9, 16, 25, \ldots$$

it's easy to guess that $g(n) = n^2$ for all n. However, at this point we don't have any rules that will allow us to prove this.

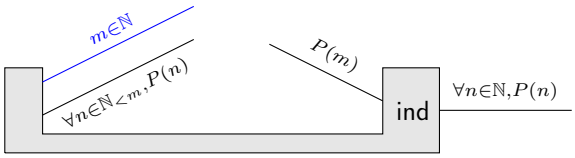
More generally, we don't have any rules for proving much of *anything* about recursive functions. For instance, we might want to prove that $a \bmod n < n$ and $a \equiv_n a \bmod n$, but we don't have any tools to do that yet.

The missing proof rule is called *proof by induction*. The basic idea is that just as we can define a sequence of *numbers* recursively, in which the definition of each $f(n)$ can use the previously computed *values* of f, we can prove a sequence of *statements* recursively (or "inductively"), in which the *proof* of each $P(n)$ can use the truth of the previously *proven statements*. Just as the above recursive definition of g is a simple finite rule that produces, when evaluated, a sequence of values, a *proof by induction* that $\forall n \in \mathbb{N}, g(n) = n^2$ will be a finite argument that can produce, when "evaluted", a *sequence of proofs* of the desired statements

$$g(0) = 0^2,\ g(1) = 1^2,\ g(2) = 2^2,\ g(3) = 3^2,\ g(4) = 4^2,\ \ldots$$

And just as the recursive definition of $g(n)$ can use the values of g at smaller inputs than n, the inductive proof of the statement $g(n) = n^2$ can *assume* that we have already proven the corresponding statement at smaller inputs than n.

The graphical representation of proof by induction looks like this:



The output wire tells us that induction applies when we want to prove some statement about *all natural numbers*,[3] $\forall n \in \mathbb{N}, P(n)$. The rest of the induction block is similar to the *prove "for all"* block in that we assume an arbitrary natural number $m \in \mathbb{N}$ and we must prove $P(m)$. The difference is that we get an extra assumption: that $P(n)$ is true (or "has already been proven") for all natural numbers n that are less than m. This is called the *inductive hypothesis*. Here $\mathbb{N}_{<m}$ denotes the collection of all natural numbers less than m, although instead of $\forall n \in \mathbb{N}_{<m}$ we can also write $\forall n < m$ if we remember that n must also be a natural number.

To see how this works in practice, let's prove our above claim about g.

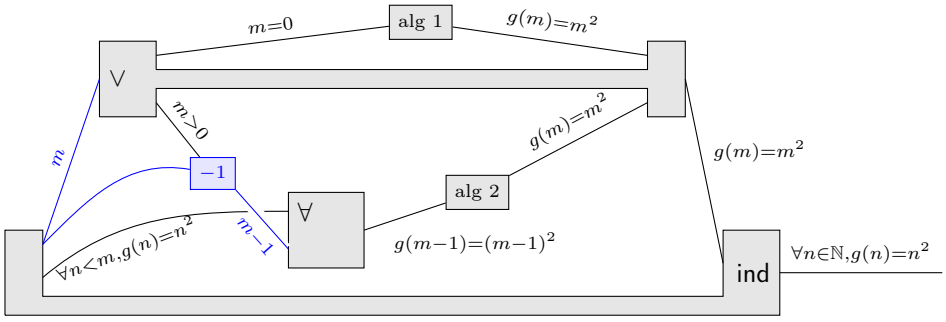
Theorem 5.1. *Let the function $g : \mathbb{N} \to \mathbb{N}$ be defined recursively by*

$$g(n) = \begin{cases} 0 & \text{if } n = 0 \\ g(n-1) + 2n - 1 & \text{if } n > 0. \end{cases}$$

[3] This restriction can be generalized somewhat; see section 5.3. But we can't use induction to prove an arbitrary ∀-statement; there is always a restriction on the type of the variable.

Then $\forall n \in \mathbb{N}, (g(n) = n^2)$.

Graphical Proof of 5.1.



Algebra 1: $g(0) = 0 = 0^2$ by definition.
Algebra 2:

$$\begin{aligned} g(m) &= g(m-1) + 2m - 1 \\ &= (m-1)^2 + 2m - 1 \\ &= (m^2 - 2m + 1) + 2m - 1 \\ &= m^2. \end{aligned}$$

□

The inductive structure of the proof matches the recursive structure of g. Following Proof Guidance 5, after drawing the outer induction block, we divide into cases based on whether $m = 0$ or $m > 0$, just as in the piecewise definition of g. This is an instance of the sort of "or" fact we discussed in section 2.6; here for simplicity we simply connect m to the input of the *use "or"* block rather than indicating the fact specifically.

In the case when $m = 0$, the claim follows by easy algebra. The case when $m > 0$ is when we have to use the inductive hypothesis. Since the recursive definition of $g(m)$ in this case involves $g(m-1)$, we expect to need to use the inductive hypothesis when $n = m - 1$. Thus, $m - 1$ is the value input of the *use "for all"* block; we have connected the $m > 0$ assumption to the -1 block just to indicate that this assumption is necessary to ensure that $m - 1$ is still a natural number. As always, the output of *use "for all"* is obtained by substituting the value $m - 1$ for the $\forall$-variable n in the statement $g(n) = n^2$, yielding $g(m-1) = (m-1)^2$. And finally, we just have to do some algebra to put this equation together with the recursive definition of $g(m)$ to obtain the desired goal.

This close connection between the recursive definition of g and our inductive proof about it is very general:

Proof Guidance 10 (Recursion and induction)**.** When the statement of a theorem involves a function or operation that is defined recursively, often it will be useful to prove the theorem by induction in an analogous way.

To write this proof in English, we need another basic principle.

Principle of English Proof 9 (Induction)**.** When doing a proof by induction, state the inductive hypothesis clearly as an assumption with a phrase like "assume inductively". Then, when using this hypothesis later, you can refer to it with "by the inductive hypothesis".

In addition, when dividing into cases in an inductive proof, the case(s) not using the inductive hypothesis are often called the *base case(s)*, while the case(s) that do use it are called the *inductive step(s)*.

English Proof of 5.1. Let $m \in \mathbb{N}$, and assume inductively that $g(n) = n^2$ for all natural numbers $n < m$.
Base case: Assume $m = 0$. Then $g(0) = 0 = 0^2$.
Inductive step: Assume $m > 0$. Then $m-1 < m$, so by the inductive hypothesis, $g(m-1) = (m-1)^2$. Therefore,

$$\begin{aligned} g(m) &= g(m-1) + 2m - 1 \\ &= (m-1)^2 + 2m - 1 \\ &= (m^2 - 2m + 1) + 2m - 1 \\ &= m^2. \end{aligned}$$

□

If you recall the notions of Σ and Π from section 3.2, you may notice that the definition of g is equivalent to saying that $g(n)$ adds up the first n odd numbers, so we can equivalently write

$$g(n) = \sum_{k=1}^{n} (2k-1).$$

Thus, our proof that $g(n) = n^2$ can equivalently be regarded as a proof that

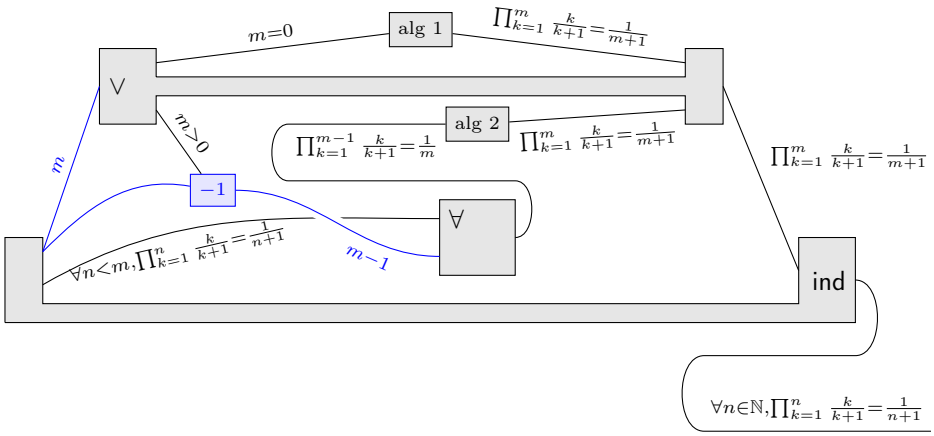
$$\sum_{k=1}^{n} (2k-1) = n^2.$$

More generally, induction is often useful for proving the value of indexed sums and products. In the base case, we often want to use the fact that the sum of no things is 0 and the product of no things is 1, while in the inductive step we often want to use the fact that we can "peel off" the last element of a sum or product, as discussed in section 3.2. For example:

Theorem 5.2. *For any $n \in \mathbb{N}$,*

$$\prod_{k=1}^{n} \frac{k}{k+1} = \frac{1}{n+1}.$$

Graphical Proof of 5.2.



Algebra 1:

$$\prod_{k=1}^{0} \frac{k}{k+1} = 1 = \frac{1}{0+1}.$$

Algebra 2:

$$\begin{aligned} \prod_{k=1}^{m} \frac{k}{k+1} &= \left(\prod_{k=1}^{m-1} \frac{k}{k+1} \right) \cdot \frac{m}{m+1} \\ &= \frac{1}{(m-1)+1} \cdot \frac{m}{m+1} \\ &= \frac{1}{m+1}. \end{aligned}$$

□

English Proof of 5.2. Let $m \in \mathbb{N}$, and assume inductively that

$$\prod_{k=1}^{n} \frac{k}{k+1} = \frac{1}{n+1} \qquad \text{for all natural numbers } n < m.$$

Base case: Assume $m = 0$. Then

$$\prod_{k=1}^{0} \frac{k}{k+1} = 1 = \frac{1}{0+1}.$$

Inductive step: Assume $m > 0$. Then

$$\begin{aligned} \prod_{k=1}^{m} \frac{k}{k+1} &= \left(\prod_{k=1}^{m-1} \frac{k}{k+1} \right) \cdot \frac{m}{m+1} \\ &= \frac{1}{(m-1)+1} \cdot \frac{m}{m+1} \\ &= \frac{1}{m+1}. \end{aligned}$$

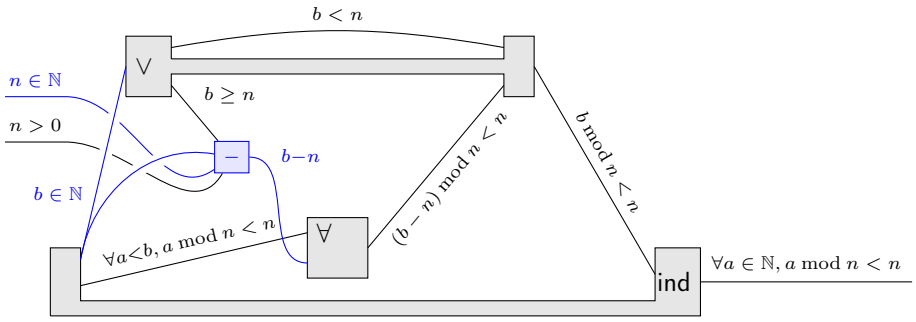
□

Induction is tricky, so let's do a few more examples. Recall the recursive definition of $a \bmod n$ from section 5.1:

$$a \bmod n = \begin{cases} a & \text{if } a < n \\ (a-n) \bmod n & \text{if } a \geq n. \end{cases}$$

Theorem 5.3. *Suppose* $a, n \in \mathbb{N}$ *and* $n > 0$. *Then* $a \bmod n < n$.

Graphical Proof of 5.3.



□

In this proof we have made use of the "implicit ∀ rule" described in section 3.9 to shift the variable $a \in \mathbb{N}$ into a ∀ so that we can prove it by induction, while leaving n as a parameter and $n > 0$ as an assumption. (Formally, there is still a *prove "for all"* for n and a *prove "if-then"* for $n > 0$, but as usual we omit them.)

Following Proof Guidance 5 again, the cases are $b < n$ and $b \geq n$, matching those in the recursive definition. And once again, rather than referring explicitly to the ∨ fact we are using, we just connect b directly to the *use "or"* block. We have also simplified the goal $b \bmod n < n$ in both cases to make it clear what we have to prove. And finally, in the inductive step, we need both assumptions $b \geq n$ and $n > 0$ to ensure that $b - n$ is a natural number less than b, so we can apply the inductive hypothesis to it. But the algebra in both cases is trivial.

English Proof of 5.3. Let $b \in \mathbb{N}$ and assume inductively that $a \bmod n < n$ for all natural numbers $a < b$.
Base case: Assume $b < n$. Then $b \bmod n = b$, so $b \bmod n < n$.
Inductive step: Assume $b \geq n$, so $b \bmod n = (b-n) \bmod n$. Then $b - n$ is a natural number, and since $n > 0$ we have that $b - n < b$. Thus, by the inductive hypothesis, $(b-n) \bmod n < n$, hence $b \bmod n < n$. □

Our last examples are starting to get too large for a graphical representation to be practical, so we write them only with English.

Theorem 5.4. *Suppose* $a, n \in \mathbb{N}$ *and* $n > 0$. *Then* $a \equiv_n a \bmod n$.

English Proof of 5.4. Let $b \in \mathbb{N}$ and assume inductively that $a \equiv_n a \bmod n$ for all natural numbers $a < b$.

Base case: Assume $b < n$. Then $b \bmod n = b$, so $b \equiv_n b \bmod n$.
Inductive step: Assume $b \geq n$, so $b \bmod n = (b-n) \bmod n$. Then $b - n$ is a natural number, and since $n > 0$ we have that $b - n < b$. Thus, by the inductive hypothesis, $b - n \equiv_n (b-n) \bmod n$.

Let $x \in \mathbb{Z}$ be such that $b - n - (b-n) \bmod n = xn$. Then

$$\begin{aligned} b - n - b \bmod n &= xn \\ b - b \bmod n &= xn + n \\ b - b \bmod n &= (x+1)n \end{aligned}$$

Therefore, $b \equiv_n b \bmod n$. □

In this proof, the inductive step looks like one of the ∃ proofs from section 3.7. This shouldn't be surprising, since in that step our assumption (the inductive hypothesis) and our goal are both ∃-statements (after writing out the definition of $\equiv_n$). In general, we can put together any proof methods however the wires match up, following the logical structure. In particular, each case of an inductive proof might be any other kind of proof, as needed.

Finally, recall the definition of the Fibonacci sequence from section 5.1:

$$F(n) = \begin{cases} 1 & \text{if } n \leq 1 \\ F(n-2) + F(n-1) & \text{if } n > 1 \end{cases}$$

In fact, it turns out that there is also a direct *formula* for the Fibonacci numbers. Surprisingly, that formula involves general real numbers, not just integers. Define the following:

$$\phi = \frac{1+\sqrt{5}}{2} \qquad \overline{\phi} = \frac{1-\sqrt{5}}{2}$$

and note that $\phi - \overline{\phi} = \sqrt{5}$ and $\phi + \overline{\phi} = 1$ and $\phi^2 = \phi + 1$ and $\overline{\phi}^2 = \overline{\phi} + 1$.

Theorem 5.5. *For all $n \in \mathbb{N}$, we have $F(n) = \frac{1}{\sqrt{5}}(\phi^{n+1} - \overline{\phi}^{n+1})$.*

English Proof of 5.5. Let $m \in \mathbb{N}$, and assume inductively that

$$F(n) = \tfrac{1}{\sqrt{5}}(\phi^{n+1} - \overline{\phi}^{n+1}) \qquad \text{for all natural numbers } n < m.$$

Base case 1: Assume $m = 0$. Then $F(0) = 1$, and

$$\tfrac{1}{\sqrt{5}}(\phi^{0+1} - \overline{\phi}^{0+1}) = \tfrac{1}{\sqrt{5}}(\sqrt{5}) = 1.$$

Base case 2: Assume $m = 1$. Then $F(1) = 1$, and

$$\tfrac{1}{\sqrt{5}}(\phi^{1+1} - \overline{\phi}^{1+1}) = \tfrac{1}{\sqrt{5}}((\phi+1) - (\overline{\phi}+1)) = \tfrac{1}{\sqrt{5}}(\phi - \overline{\phi}) = 1.$$

Inductive step: Assume $m > 1$. Then using the inductive hypothesis for $m-2$ and $m-1$, we have

$$\begin{aligned}
F(m) &= F(m-2) + F(m-1) \\
&= \tfrac{1}{\sqrt{5}}(\phi^{(m-2)+1} - \overline{\phi}^{(m-2)+1}) + \tfrac{1}{\sqrt{5}}(\phi^{(m-1)+1} - \overline{\phi}^{(m-1)+1}) \\
&= \tfrac{1}{\sqrt{5}}(\phi^{m-1} - \overline{\phi}^{m-1}) + \tfrac{1}{\sqrt{5}}(\phi^{m} - \overline{\phi}^{m}) \\
&= \tfrac{1}{\sqrt{5}}(\phi^{m-1} - \overline{\phi}^{m-1} + \phi^{m} - \overline{\phi}^{m}) \\
&= \tfrac{1}{\sqrt{5}}((\phi^{m} + \phi^{m-1}) - (\overline{\phi}^{m} + \overline{\phi}^{m-1})) \\
&= \tfrac{1}{\sqrt{5}}(\phi^{m-1}(\phi+1) - \overline{\phi}^{m-1}(\overline{\phi}+1)) \\
&= \tfrac{1}{\sqrt{5}}(\phi^{m-1}\phi^2 - \overline{\phi}^{m-1}\overline{\phi}^2) \\
&= \tfrac{1}{\sqrt{5}}(\phi^{m+1} - \overline{\phi}^{m+1}).
\end{aligned}$$

□

Of course, while this proof establishes the truth of the theorem, we are left wondering how in the world we might guess such an odd-looking formula. In fact there is a systematic way to "solve" recursive definitions of this sort, which you can read about in many places. One is https://discrete.openmathbooks.org/dmoi4/sec_seq-exponential.html in the free online book *Discrete Mathematics: An Open Introduction* by Oscar Levin.

Exercises

As usual, write your proofs using both graphical notation and in English.

Exercise 5.2.1. Let the function $f : \mathbb{N} \to \mathbb{N}$ be defined recursively by

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ f(n-1) + 2^n & \text{if } n > 0. \end{cases}$$

Prove $\forall n \in \mathbb{N}, (f(n) = 2^{n+1} - 1)$. (A footnote[4], with some unnecessary words added so that it doesn't look like it's raising something to the 4th power.)

Exercise 5.2.2. Let the function $f : \mathbb{N} \to \mathbb{N}$ be defined recursively by

$$f(n) = \begin{cases} 2 & \text{if } n = 0 \\ 2 \cdot f(n-1) - 1 & \text{if } n > 0. \end{cases}$$

Prove $\forall n \in \mathbb{N}$, $f(n) = 2^n + 1$.

Exercise 5.2.3. Let the function $f : \mathbb{N} \to \mathbb{N}$ be defined recursively by

$$f(n) = \begin{cases} 2 & \text{if } n = 0 \\ \frac{1}{2} f(n-1) + 1 & \text{if } n > 0. \end{cases}$$

Prove $\forall n \in \mathbb{N}$, $f(n) = 2$.

[4] When talking about this problem with your friends or your instructor, remember that 2^n is pronounced "two to the n" (or "two to the power n" if you want to be pedantic). I have heard far too many students pronounce it as "two n" which is *something totally different*.

Exercise 5.2.4. Let the function $f : \mathbb{N} \to \mathbb{R}$ be defined recursively by

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ f(n-1) \cdot \frac{(n+1)}{2n} & \text{if } n > 0. \end{cases}$$

Prove $\forall n \in \mathbb{N}$, $f(n) = \dfrac{n+1}{2^n}$.

Exercise 5.2.5. Let the function $f : \mathbb{N} \to \mathbb{N}$ be defined recursively by

$$f(n) = \begin{cases} 3 & \text{if } n = 0 \\ 3 \cdot f(n-1) - 4 & \text{if } n > 0. \end{cases}$$

Prove $\forall n \in \mathbb{N}$, $f(n) = 3^n + 2$.

Exercise 5.2.6. Let the function $f : \mathbb{N} \to \mathbb{N}$ be defined recursively by

$$f(n) = \begin{cases} 3 & \text{if } n = 0 \\ \frac{1}{3} f(n-1) + 2 & \text{if } n > 0. \end{cases}$$

Prove $\forall n \in \mathbb{N}$, $f(n) = 3$.

Exercise 5.2.7. Prove that for any $n \in \mathbb{N}$,

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}.$$

Exercise 5.2.8. Prove that for any $n \in \mathbb{N}$,

$$\sum_{k=0}^{n} 3^k = \frac{3^{n+1} - 1}{2}.$$

Exercise 5.2.9. Prove $\forall n \in \mathbb{N}, (2 \mid (3^n - 1))$.

Exercise 5.2.10. Prove $\forall n \in \mathbb{N}, (3 \mid (4^n - 1))$.

Exercise 5.2.11. Prove $\forall n \in \mathbb{N}, (n^3 \equiv_3 n)$.

Exercise 5.2.12. Prove $\forall n \in \mathbb{N}, (2 \mid (n^2 - n))$.

Exercise 5.2.13. Prove that $\forall n \in \mathbb{N}$, $n^4 \equiv_2 n$.
(Hint: for any x we have $(x-1)^4 = x^4 - 4x^3 + 6x^2 - 4x + 1$.)

Exercise 5.2.14. Prove that $\forall n \in \mathbb{N}, (6 \mid (n^3 - n))$. There are at least two ways to do this:

- Use induction with a split into the cases $m = 0$ and $m > 0$, and use Exercise 5.2.12.

- Use induction with a split into the cases $m \leq 1$ and $m > 1$, with the inductive hypothesis applied to $m - 2$ in the second case.

At this point, feel free to switch to writing proofs only in English.

Exercise 5.2.15. Let the sequence $f : \mathbb{N} \to \mathbb{N}$ be defined by

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 4 & \text{if } n = 1 \\ 5 \cdot f(n-1) - 6 \cdot f(n-2) & \text{if } n > 1 \end{cases}$$

Prove that $\forall n \in \mathbb{N}, f(n) = 2 \cdot 3^n - 2^n$.

Exercise 5.2.16. Let the sequence $f : \mathbb{N} \to \mathbb{N}$ be defined by

$$f(n) = \begin{cases} 2 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ 2 \cdot f(n-2) + f(n-1) & \text{if } n > 1 \end{cases}$$

Prove that $\forall n \in \mathbb{N}, f(n) = 2^n + (-1)^n$.

Exercise 5.2.17. Let the sequence $f : \mathbb{N} \to \mathbb{N}$ be defined by

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 7 & \text{if } n = 1 \\ 5 \cdot f(n-1) - 4 \cdot f(n-2) & \text{if } n > 1 \end{cases}$$

Prove that $\forall n \in \mathbb{N}, f(n) = 2 \cdot 4^n - 1$.

5.3 More general induction

Recall from section 5.1 the two basic requirements for a recursive function to make sense:

1. When the computation of $f(n)$ involves a recursive call to $f(k)$, the input k of the recursive call is always an expression that is *smaller than* n.

2. There are one or more "smallest cases" in which there are *no* recursive calls.

These requirements also apply to induction. However, they are actually significantly more general than it may appear. Up until now, we have considered only functions whose input is a natural number, and where "smaller than" refers to the ordinary ordering on natural numbers. But we could, in principle, consider recursion and induction with more general inputs and orderings.

In complete generality, this doesn't work. For example, suppose we try to define a function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ (where $\mathbb{R}_{\geq 0}$ denotes the nonnegative real numbers) by

¿¿¿ $$f(x) = \begin{cases} 1 & \text{if } x = 0 \\ f(\frac{x}{2}) + 1 & \text{if } x > 0. \end{cases}$$???

Clearly there is a "smallest case" $x = 0$ that makes no recursive calls, and each recursive call supplies a smaller argument $\frac{x}{2}$ than x. But we can't actually compute the value of $f(x)$ for any nonzero x. For example,

$$\begin{aligned} f(1) &= f(\tfrac{1}{2}) + 1 \\ &= f(\tfrac{1}{4}) + 2 \\ &= f(\tfrac{1}{8}) + 3 \\ &= f(\tfrac{1}{16}) + 4 \\ &= \dots \end{aligned}$$

Evidently this process never stops: even though the argument decreases at every step, and there is a "smallest case", we never *reach* the smallest case.

What we need for recursion and induction to make sense is the following.

Definition 5.6. *An ordered collection of numbers is* ***well-founded*** *if it does* not *contain any infinite strictly decreasing chains*

$$x_0 > x_1 > x_2 > x_3 > \cdots .$$

This condition precisely guarantees that if the argument of a recursive call decreases at every step, there must eventually be no more recursive calls. The natural numbers are well-founded: starting from a natural number x_0, we can decrease it at most x_0 times before we get to 0, and then we can't decrease it any more. And the problem with our f above is that $\mathbb{R}_{>0}$ is not well-founded, because we have an infinite strictly decreasing chain

$$1 > \tfrac{1}{2} > \tfrac{1}{4} > \tfrac{1}{8} > \tfrac{1}{16} > \cdots .$$

We have encountered one other number system that *is* well-founded: the *ordinal numbers* Ω. I haven't defined these precisely for you, but all you need to know right now is what I said about them in section 1.6:

- Ω contains all the natural numbers as well as ω, which is greater than all the natural numbers.
- The sum and product of two ordinal numbers are also ordinal numbers.
- There is an operation taking an ordinal number x to another ordinal number ω^x, such that $\omega^0 = 1$, $\omega^1 = \omega$, and $\omega^{x+y} = \omega^x \omega^y$.

If we list the ordinal numbers in order, with ellipses to skip over infinite sequences of them, we get something like this:

$$0, 1, 2, 3, \ldots, \omega, \omega+1, \omega+2, \ldots, 2\omega, 2\omega+1, \ldots, 3\omega, 3\omega+1, \ldots, 4\omega, \ldots,$$
$$\omega^2, \omega^2+1, \omega^2+2, \ldots, \omega^2+\omega, \omega^2+\omega+1, \ldots, \omega^2+2\omega, \ldots, \omega^2+3\omega, \ldots,$$
$$2\omega^2, \ldots, 3\omega^2, \ldots 4\omega^2, \ldots, \omega^3, \ldots, \omega^4, \ldots, \omega^\omega, \ldots, \omega^{\omega+1}, \ldots, \omega^{\omega+2}, \ldots,$$
$$\omega^{2\omega}, \ldots, \omega^{3\omega}, \ldots, \omega^{\omega^2}, \ldots, \omega^{\omega^3}, \ldots, \omega^{\omega^\omega}, \ldots, \omega^{\omega^{\omega^\omega}}, \ldots, \omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}}, \ldots$$

and so on.[5] In fact the ordinal numbers are almost *defined* by the phrase "and so on": given any set of ordinals, there's always a next greater one. (That is, when using Conway's counting technique from section 1.6, you can never run out of ordinal numbers to count with.)

It may seem counterintuitive that a collection containing infinitely large numbers like ω can be well-founded, but the point is that ω is the *least* infinite ordinal number: surreal numbers like $\omega - 1$ and $\frac{\omega}{2}$ and $\sqrt{\omega}$ are not *ordinal* numbers. Thus, if we start a descending chain from ω, the next element after ω will have to be some *finite* natural number, and after that the sequence can only go on for finitely many terms since $\mathbb{N}$ is well-founded. Similarly, if we start from 2ω, the next element can be at most $\omega + n$ for some finite n, after which point there can be only finitely many terms until we get to ω itself, and then finitely many more after that. And if we start from ω^2, the next element can be at most $a\omega + b$ for some finite natural numbers a and b, and so on.

Here is an example of a function defined recursively on ordinal numbers. Let D be the collection of ordinal numbers of the form $a\omega + b$, for $a, b \in \mathbb{N}$. Define $f : D \to \mathbb{N}$ by

$$f(a\omega+b) = \begin{cases} b+1 & \text{if } a = 0 \\ f\big((a-1)\omega+1\big) & \text{if } a > 0 \text{ and } b = 0 \\ f\big((a-1)\omega + f(a\omega+b-1)\big) & \text{if } a > 0 \text{ and } b > 0. \end{cases}$$

Note that all the recursive calls are on smaller inputs: $a\omega + b - 1 < a\omega + b$, and $(a-1)\omega + c < a\omega + b$ no matter how large c is. (Also, all the numbers are ordinals, since we only subtract one from natural numbers that are known to be positive.) Thus, this recursive definition is valid.

Note also that the input of f is determined by the two natural numbers a and b. The corresponding function of these two variables:

$$A(a, b) = f(a\omega + b)$$

---

[5] The notation $\omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}}$, while evocative, is not really correct. The ordinal number that goes where I've written that is actually called ε_0 (not to be confused with the "permittivity of free space", a physical constant also denoted ε_0 with a value of $\approx 8.854 \times 10^{-12}\,\mathsf{F{\cdot}m^{-1}}$). In fact, there are infinitely many ordinal numbers ε that satisfy the equation $\varepsilon = \omega^\varepsilon$ and hence could be written as $\omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}}$, of which ε_0 is merely the ~~first~~ zeroth.

is called the *Ackermann function*. It is famous for getting very large very quickly; for example, $A(4,3) = 2^{2^{65536}} - 3$ is much larger than a "googolplex" $10^{10^{100}}$. There's a certain sense in which the Ackermann function is "impossible" to define by recursion solely over $\mathbb{N}$ rather than over the ordinals.[6]

It's common to be in a situation like this where the function we "really" want to define, like A, doesn't have its domain being exactly a subset of $\mathbb{N}$ or $\mathbb{N}$, so we seem to have to rearrange it into a related function like the above f. However, the rearrangement isn't actually necessary: as long as we have a way to assign a natural number or ordinal number "size" to every input of the function, such that the *size* decreases with every recursive call, we are guaranteed that a recursive function definition will terminate. Such a "size" is sometimes called a *termination measure*. Thus, we could have defined the Ackermann function directly:

$$A(a,b) = \begin{cases} b+1 & \text{if } a = 0 \\ A\big(a-1, 1\big) & \text{if } a > 0 \text{ and } b = 0 \\ A\big(a-1, A(a,b-1)\big) & \text{if } a > 0 \text{ and } b > 0. \end{cases}$$

and justified the recursion with the termination measure $a\omega + b$.

The notion of termination measure can be useful even when the sizes are ordinary natural numbers. For example, the function gcd that computes the *greatest common divisor* of two positive integers can be defined by

$$\gcd(a,b) = \begin{cases} a & \text{if } a = b \\ \gcd(a{-}b,\, b) & \text{if } a > b \\ \gcd(a,\, b{-}a) & \text{if } b > a. \end{cases}$$

Here the termination measure is $\max(a,b)$: the definition ensures that whichever of a and b is greater decreases with each recursive call. It's possible to rearrange this into an ordinary definition by recursion, but that requires testing and keeping track of *which* of the arguments is larger, in contrast to the simple definition above. (There's also a more efficient version of the definition that uses $a \bmod b$ rather than $a - b$.)

As another example, suppose we want to extend the definition of $a \bmod n$ to the case when a is negative. Just as when a is positive we *subtract* copies of n until we get a value between 0 and $n-1$, when a is negative we should *add* copies of n until we get such a result. So we can define

$$a \bmod n = \begin{cases} a & \text{if } 0 \le a < n \\ (a-n) \bmod n & \text{if } a \ge n \\ (a+n) \bmod n & \text{if } a < 0. \end{cases}$$

[6] To be precise, it's impossible if the only kinds of functions you're allowed to define are functions of some number of natural number variables whose output is a natural number; these are called *primitive recursive*. You can define the Ackermann function by recursion on $\mathbb{N}$ if you're allowed to "curry" it and define it as a recursive function of *one* variable a whose output is another *function* $\mathbb{N} \to \mathbb{N}$ that takes as input the other variable b.

To show that this is well-defined, we can use the termination measure

$$\text{size}(a) = \begin{cases} a & \text{if } a \geq 0 \\ n - a & \text{if } a < 0. \end{cases}$$

This is always a natural number. It decreases with the recursive call when $a \geq n$ since $a - n < a$. And it decreases with the recursive call when $a < 0$ because, if $a + n < 0$ then $\text{size}(a + n) = n - (a + n) = -a < n - a = \text{size}(a)$, while if $a + n \geq 0$ then $a < -a$, so $\text{size}(a + n) = a + n < n - a = \text{size}(a)$. This is a good example to show how we can use termination measures for inductive proofs as well as recursive definitions:

Theorem 5.7. *For any $a \in \mathbb{Z}$ and positive integer n, we have $0 \leq a \bmod n < n$.*

English Proof of 5.7. Let $b \in \mathbb{Z}$, and assume inductively that $0 \leq a \bmod n < n$ for all integers a with $\text{size}(a) < \text{size}(b)$.
Base case: Assume $0 \leq b < n$. Then $b \bmod n = b$, so $0 \leq b \bmod n < n$.
Inductive step 1: Assume $b \geq n$, so $b \bmod n = (b - n) \bmod n$. Then, by the inductive hypothesis, $0 \leq (b - n) \bmod n < n$, hence $0 \leq b \bmod n < n$.
Inductive step 2: Assume $b < 0$, so $b \bmod n = (b + n) \bmod n$. Then, by the inductive hypothesis, $0 \leq (b + n) \bmod n < n$, hence $0 \leq b \bmod n < n$. □

One last example of the utility of ordinal termination measures is known as *Goodstein's Theorem*. This starts from the observation that for any integer $b > 1$, any positive integer n can be written in "base b", analogous to the ordinary "base 10" representation. Starting from the right, the base b representation has a units digit, then a b's digit, then a b^2's digit, and so on. For instance, the digit string 324_5 in base 5 (the subscript "5" means that it's in base 5, so we don't confuse it with the base 10 number three hundred twenty-four) means what we would ordinarily represent in base 10 as

$$3 \cdot 5^2 + 2 \cdot 5^1 + 4 \cdot 5^0 = 89.$$

The digits in a base b representation must all be less than b. When b is greater than 10, we don't have enough ordinary digits, so we start using letters; for instance, base 16 ("hexadecimal") uses digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. No one uses bases greater than 36 in practice, but in principle they are possible, and we can always write them in the above explicit form with powers of b. For instance,

$$17 \cdot 41^2 + 20 \cdot 41 + 39$$

is a number written in base 41, which equals 29,436 in base 10.

Of course, we can allow negative exponents of b also. If we allow an infinite sequence of negative exponents, but only a finite sequence of positive ones, then the numbers we can represent are precisely the real numbers $\mathbb{R}$, just as in the familiar case $b = 10$. And if we allow an infinite sequence of *positive* exponents,

but only a finite sequence of *negative* ones, then we get the b-adic numbers $\mathbb{Q}_b$, generalizing the 10-adic numbers that I mentioned in section 1.5.

Goodstein's Theorem is about *hereditary base-b notation*, where we write a natural number in base b and then proceed to write all the *exponents* in base b as well, and so on. For example, 35 written in base 2 is

$$1 \cdot 2^5 + 1 \cdot 2^1 + 1 \cdot 2^0$$

but the 5 here is not written in base 2. If we write it in base 2 as $1 \cdot 2^2 + 1 \cdot 2^0$, then we get

$$1 \cdot 2^{1 \cdot 2^2 + 1 \cdot 2^0} + 1 \cdot 2^1 + 1 \cdot 2^0$$

but the secondary exponent 2 is still not in base 2. If we write it in base 2 as $1 \cdot 2^1$, we finally get the hereditary base-2 notation of 35:

$$1 \cdot 2^{1 \cdot 2^{1 \cdot 2^1} + 1 \cdot 2^0} + 1 \cdot 2^1 + 1 \cdot 2^0.$$

Now, if we have a number written in hereditary base-b notation, we can increase the base and obtain a different number written in a different hereditary base. For instance, if we increase the base 2 to 3 in the representation of 35 above, we get

$$1 \cdot 3^{1 \cdot 3^{1 \cdot 3^1} + 1 \cdot 3^0} + 1 \cdot 3^1 + 1 \cdot 3^0$$

which equals 22876792454965 — quite a bit larger than 35!

Given an integer n, let $H(b, n)$ denote the result of writing n in hereditary base-b notation and replacing all the bs by $b + 1$. Thus $H(2, 35) = 22876792454965$, and as this example suggests, H often grows very quickly. However, if $n < b$, then the hereditary base-b representation of n is just $n \cdot b^0$, so in this case $H(b, n) = n \cdot (b + 1)^0 = n$ again.

Now we try to define the *Goodstein function* by

$$\mathcal{G}(b, n) = \begin{cases} b - 1 & \text{if } n = 0 \\ \mathcal{G}(b + 1, H(b, n) - 1) & \text{if } n > 0. \end{cases}$$

It's not at all obvious that this is well-defined: in the recursive call, b *increases* to $b+1$, and n changes to $H(b, n) - 1$ which is *often* much, much larger than n. However, it turns out that nevertheless, eventually b gets larger than n in these recursive calls, and so the function is well-defined. This is Goodstein's Theorem. It can be proven using the following ordinal-valued termination measure: given a pair (b, n), write n in hereditary base-b notation, and then replace all the bs by ω. For example, the size of $(2, 35)$ is

$$1 \cdot \omega^{1 \cdot \omega^{1 \cdot \omega^1} + 1 \cdot \omega^0} + 1 \cdot \omega^1 + 1 \cdot \omega^0 \;=\; \omega^{\omega^\omega + 1} + \omega + 1$$

while the size of $(3, 22876792454964)$ (note that I subtracted one from $H(2, 35)$, as in the definition of $\mathcal{G}$) is

$$\omega^{\omega^\omega + 1} + \omega$$

and the size of $(4, H(3, 22876792454964) - 1)$ is

$$\omega^{\omega^{\omega}+1} + 3,$$

which do, indeed, appear to be getting smaller! I will not give the complete proof here, but hopefully this example is convincing.

Finally, one of the most famous open problems in mathematics can be phrased as the well-definedness of a certain recursive definition. We try to define a function $C : \mathbb{Z}_{>0} \to \mathbb{Z}$ on the positive integers as follows:

$$C(x) = \begin{cases} 1 & \text{if } x = 1 \\ C(\frac{x}{2}) & \text{if } x \text{ is even} \\ C(3x+1) & \text{if } x \text{ is odd and } x > 1 \end{cases}$$

Is this recursive definition valid? It terminates at every input anyone has ever tried (up to about 10^{20}), but no one knows whether it will always do so. This is called the *Collatz conjecture*. Paul Erdös, one of the most brilliant and prolific mathematicians of the 20th century, reportedly said of the Collatz conjecture that "mathematics is not yet ready for such problems."

Exercises

Feel free to write these proofs only in English. I haven't found a source of drill exercises on this subject yet, so there are a limited number of these exercises and many of them require more thought. An instructor doing a unit on number theory may want to work through some of these exercises in class.

Exercise 5.3.1. Generalizing the proof of Theorem 5.4 to use a termination measure analogously to Theorem 5.7, prove that $a \equiv_n a \bmod n$ for any *integer* a and any positive integer n.

Exercise 5.3.2. The operation $a \bmod n$ is also known as the *remainder* when a is divided by n. The other part of "integer division" is the *integer quotient*, which we will write as "$a \operatorname{div} n$". Give a recursive definition of $a \operatorname{div} n$ that works for any integer a and any positive integer n, using the same termination measure as $a \bmod n$.

Exercise 5.3.3. Prove that for any $a, n \in \mathbb{Z}$ with $n > 0$, we have

$$a = (a \operatorname{div} n) \cdot n + (a \bmod n).$$

Exercise 5.3.4. Suppose $a, n \in \mathbb{Z}$ with $a > 0$ and $n \geq 2$. Prove $a \operatorname{div} n < a$.

Exercise 5.3.5. Prove, using induction with the same termination measure as the recursive definition of gcd, that $\gcd(a, b) \mid a$ and $\gcd(a, b) \mid b$ for any positive integers a, b.

Exercise 5.3.6. Prove that for any positive integers a, b, c, if $c \mid a$ and $c \mid b$, then $c \mid \gcd(a, b)$. *(It follows that* $\gcd(a, b)$ *really is the greatest common divisor of* a *and* b*, as intended.)*

Exercise 5.3.7. Prove that for any integer a we have $(a \equiv_2 0) \vee (a \equiv_2 1)$.

Exercise 5.3.8. Prove that for any integer a, if $a^2 \equiv_2 0$, then $a \equiv_2 0$.

Exercise 5.3.9. Prove that for any positive integers a, b, there exist integers (not necessarily positive) x, y such that $ax + by = \gcd(a, b)$.

Exercise 5.3.10. Prove that if $\gcd(a, n) = 1$ then there exists an integer x such that $ax \equiv_n 1$. *(Such an x is called a "multiplicative inverse" of a modulo n. When it exists, we can "divide by a" in $\mathbb{Z}_{/n}$, by multiplying by x.)*

Exercise 5.3.11. Formally, we define an **integral base-b expansion** to be a finite list $d = (d_k, d_{k-1}, \dots, d_0)$ of integers such that $0 \leq d_i < b$ for all i. The natural number represented by such an expansion is $[\![d]\!]_b = \sum_{i=0}^{k} d_i b^i$.

In the other direction, define a function E_b from natural numbers to integral base-b expansions, and prove that $[\![E_b(a)]\!]_b = a$ for all $a \in \mathbb{N}$.

Now try to evaluate your function at a negative integer. It won't terminate, but you should be able to see it producing an *infinite* base-b expansion, i.e. what we called a b-adic number. In particular, in this way you should be able to re-derive the expression of -1 as the 10-adic number $\cdots 99999$.

5.4 Strong vs weak induction

Now that we are about finished with our study of recursion and induction, it behooves me to tell you about the more roundabout way that most mathematicians and textbooks talk about induction. Note that in Theorem 5.1 and many of the exercises in section 5.2, we only needed to apply the inductive hypothesis at $n = m - 1$. What most mathematicians and textbooks call *induction* is restricted to this special case, for which they use a different notation and structure, which includes the division into cases $m = 0$ and $m > 0$.

Specifically, instead of introducing a variable m and an inductive hypothesis at the beginning, they start right away with a base case proving $P(0)$. Then they introduce the inductive step, writing k for our $m - 1$, assume as an inductive hypothesis *only* that $P(k)$ (our $P(m-1)$) is true, and then proceed to prove $P(k+1)$ (our $P(m)$). For example, here is a proof of Theorem 5.1 in that style:

Theorem 5.1. *Let the function $g : \mathbb{N} \to \mathbb{N}$ be defined recursively by*

$$g(n) = \begin{cases} 0 & \text{if } n = 0 \\ g(n-1) + 2n - 1 & \text{if } n > 0. \end{cases}$$

Then $\forall n \in \mathbb{N}, (g(n) = n^2)$.

Weak induction proof of Theorem 5.1. We use induction.
Base case: $g(0) = 0 = 0^2$.

Inductive step: Let $k \in \mathbb{N}$, and assume inductively that $g(k) = k^2$. Therefore,

$$\begin{aligned} g(k+1) &= g(k) + 2(k+1) - 1 \\ &= k^2 + (2k+2) - 1 \\ &= k^2 + 2k + 1 \\ &= (k+1)^2. \end{aligned}$$

□

With an eye towards weak induction, it would be more traditional to write the definition of g differently as well:

$$\begin{aligned} g(0) &= 0 \\ g(k+1) &= g(k) + 2k + 1. \end{aligned}$$

People who describe induction in this way call our sort of induction *strong induction*, and usually introduce it later and use it more rarely. If this distinction seems unnecessarily complicated, that's because it is. In its defense, weak induction is easier to justify formally from the foundations of mathematics; but for learning and using induction in practice, I believe it is simpler and easier to use only strong induction, as we have done. A manual transmission car may be easier to build, but someone learning to drive should probably start with an automatic.

Moreover, many very natural examples require strong induction. We have already used it to prove properties of the mod function and to solve recurrence relations such as the Fibonacci sequence. Other important applications of strong induction that often appear in introductory proof courses include the fact that every integer factors uniquely into primes, and various coloring and traversal theorems in graph theory.

Appendix A

Principles of English Proof

Principle of English Proof 1 (Since)**.** In an English proof, we write "Since P and Q, we have R" to mean that we already know P and Q, and we are deducing R from them by a rule that the reader should be able to guess.

There are many possible variations. Instead of two facts P and Q we can have one, three, or any number. In place of "since" we can use an equivalent word like "because". If one of the facts (or the only one) was just mentioned in the sentence immediately before, we don't need to restate it; in this case we generally use a word like "hence", "thus", or "therefore". (Some people use the symbol $\therefore$ to mean "therefore", but I find this very hard to notice on the page, so I do not recommend it.) The filler phrase "we have" can be omitted; it is mainly used when Q ends with a variable or symbol and R begins with a variable or symbol, according to a general readability principle that distinct mathematical formulae should be separate by words and not just punctuation. Finally, to indicate what rule is being used we can end the sentence with "by ⟨rule⟩".

Principle of English Proof 2 (Cases)**.** When dividing a proof into cases, label each case clearly with markers such as "Case 1:" and "Case 2:". If desired for clarity, you can introduce the cases with a phrase like "Since . . . we can divide into cases", and/or conclude them with a phrase like "Since R is true in both cases, we must have R."

Principle of English Proof 3 (Assume)**.** When introducing a *hypothetical assumption*, which is represented graphically by a new wire coming from the left that can only be used inside a "bracket" of some kind (like the two sides of a *prove "or"* rule), indicate it with a word like "suppose" or "assume".

This is the *only* situation in which we use these words. In particular, *do not* say "assume" or "suppose" when you are *deducing* something from previously known facts; for that use something like "since" (PEP 1).

Principle of English Proof 4 (Let)**.** When introducing a *new free variable*, which is represented graphically by a new blue wire from the left, indicate it with a word like "suppose", "assume", or "let". You should generally indicate its type as well, as in "let x be a real number" or "let $x \in \mathbb{R}$".

Principle of English Proof 5 (Want to show)**.** When the "current goal" (that is, the wire coming from the right that you are trying to connect up with) changes, such as when applying a *prove "if-then"* or *prove "for all"*, it may be helpful to remind the reader of what the new goal is. But *whenever* you quote a statement in a proof that is *not yet known to be true*, such as the current goal, you *MUST* label it as such with a phrase like "we want to show that" or "we will show that" or "we must show that". Otherwise, the reader will assume that you are claiming you already *know* that statement to be true, and will be quite confused for a while trying to figure out how you know it, because you don't.

Principle of English Proof 6 (Such that)**.** When introducing a new variable *along with* a property of that variable (that is, a value wire along with a proposition wire involving it coming from the left), as in the *use "exists"* rule, label the variable with "let", "assume", or "suppose" as in PEP 4, and then label its property with "such that", as in "let x be a real number such that $x^2 = 2$". This is the *only* place that you should use "such that".

Principle of English Proof 7 (Let, again)**.** When specifying a value for a bound variable, as in the *use "for all"* and *prove "exists"* rule, if you want to explicitly indicate the value, you can say "let $x = \langle$the value$\rangle$", where x is the bound variable, *as long as there is no other x around that could create confusion.* If there is a potential for confusion, you should rename the bound variable (which, as we know, is always possible). When you specify a value explicitly like this, you should usually also state the property that this value makes the bound variable satisfy, either with or without its binder, but using the name you chose for the bound variable.

Principle of English Proof 8 (By contradiction)**.** When doing a proof by contradiction, introduce the assumption with a phrase like "assume for contradiction that ...". Once you've reached a contradiction, you can simply say "this is a contradiction", remind the reader what it contradicts, or you can emphasize the result, as in "this is a contradiction, so it must be that ...".

Principle of English Proof 9 (Induction)**.** When doing a proof by induction, state the inductive hypothesis clearly as an assumption with a phrase like "assume inductively". Then, when using this hypothesis later, you can refer to it with "by the inductive hypothesis".

In addition, when dividing into cases in an inductive proof, the case(s) not using the inductive hypothesis are often called the *base case(s)*, while the case(s) that do use it are called the *inductive step(s)*.

Appendix B

Proof Guidance

Proof Guidance 1 (Follow the structure)**.** Follow the logical structure of the givens (wires currently coming from the left) and goal (unattached wire currently coming from the right). For instance, if you have a given that is a $\wedge$ statement, try connecting it to a *use "and"* block; and if you have a goal that is a $\wedge$ statement, try connecting it to a *prove "and"* block.

Proof Guidance 2 (The goal of a case split)**.** Very often, when breaking a proof into cases with the *use "or"* rule, the goal proposition R of that rule should be the *overall goal*: either the desired conclusion of the entire theorem, or the current goal (the wire on the right you're trying to connect to) at the moment when the "or" statement becomes available. In particular, "prove" rules that involve a choice (such as the choice of which *prove "or"* rule to use) usually shouldn't be *outside* the *use "or"* rule to the right: often you'll need to make *different choices in the two cases*, so the choice rules must be used inside the two branches separately.

Proof Guidance 3 (Use previous facts)**.** If it seems like you don't have enough information to complete a proof, ask yourself what general facts or previously proven theorems you know of that could be helpful.

Proof Guidance 4 (Do scratch work)**.** When you need to specify an expression in the course of a proof, don't just pick something at random. Set the proof aside, pick up another sheet of paper, and do some *scratch work* to figure out what the best choice is. Often, your scratch work will consist of "working backwards" from the goal or some other desired equation or inequality — the sort of algebra that is *not* valid in a proof.

Proof Guidance 5 (Follow function definitions)**.** When the statement of a theorem involves a function or operation that is defined in a "piecewise" way, often it will be useful to divide the proof into cases in an analogous way (using an arithmetic "or" fact such as trichotomy) so that in each case only one "piece" of the function applies.

Proof Guidance 6 (Make extra assumptions)**.** If it seems like you don't have enough information, ask yourself what additional hypothesis would be useful, and try proving the theorem with that extra hypothesis. If you can do that, there are multiple ways to proceed. First, try modifying your proof to eliminate or weaken the extra hypothesis. Second, look for a general fact or previously proven theorem saying that either your extra hypothesis is true or something else is true, and then try proving the theorem using the something else; if that works, you've completed a proof by cases. Third, if none of that works, you can go ahead and submit or publish the theorem you've proven with the extra hypothesis: it may still be interesting and nontrivial, and maybe someone else will be able to improve it further.

Proof Guidance 7 (Use definitions)**.** If you don't see a way to proceed at the beginning of a proof, and it's not clear how to apply Proof Guidance 1, look for words that have definitions and replace them by those definitions.

Proof Guidance 8 (By Contradiction)**.** If you are stuck, try a proof by contradiction: assume the opposite of the goal and try to deduce a contradiction. You can do this at any point in a proof: at the beginning or in any case or subproof, whatever the current goal is. It is powerful because it gives you a new assumption for free: the negation of the goal. But it is tricky to use because it negates Proof Guidance 1: the logical structure of the goal can't tell you when to do a proof by contradiction, and after you've done it, the goal is now $\bot$ so it has no more logical structure to help you.

Proof Guidance 9 (Use a logical equivalence)**.** If you don't see a productive way to use the logical structure of some given or goal (as in Proof Guidance 1), you can try replacing it with a different statement that's logically equivalent.

The most common example is to use De Morgan's laws to "push $\neg$ inside". This is usually done immediately after proof by contradiction to simplify the new assumption and expose more logical structure to make Proof Guidance 1 applicable to it.

The next most common example is to replace an implication by its contrapositive. Doing this to the goal followed by *prove "if-then"* is called *proof by contrapositive*; doing this to a given followed by *use "if-then"* is called (by aficionados of fancy Latin phrases) *modus tollens*. Proof by contrapositive is very similar to proof by contradiction, and it's often just a matter of taste which you prefer in a given case.

The next next most common example is to replace $P \vee Q$ by $\neg P \Rightarrow Q$ or $\neg Q \Rightarrow P$. Doing this to to a given followed by *use "if-then"* is called the *disjunctive syllogism*. Doing this to the *goal* followed by *prove "if-then"* seems equally important to me, but it doesn't seem to have a standard name.

Proof Guidance 10 (Recursion and induction)**.** When the statement of a theorem involves a function or operation that is defined recursively, often it will be useful to prove the theorem by induction in an analogous way.

Bibliography

[Bau16] Andrej Bauer. Five stages of accepting constructive mathematics. *Bull. Amer. Math. Soc.*, 2016. https://doi.org/10.1090/bull/1556. (Cited on p. 112)

[Bre16] Joachim Breitner. Visual theorem proving with the Incredible Proof Machine. In Jasmin Christian Blanchette and Stephan Merz, editors, *Interactive Theorem Proving*, pages 123–139, Cham, 2016. Springer International Publishing. (Cited on p. 5)

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. (Cited on p. 5)

[JS91] André Joyal and Ross Street. The geometry of tensor calculus. I. *Adv. Math.*, 88(1):55–112, 1991. (Cited on p. 5)

[New] Clive Newstead. An infinite descent into pure mathematics. https://infinitedescent.xyz/. (Cited on p. 4)

[Pen71] Roger Penrose. Applications of negative dimensional tensors. In *Combinatorial Mathematics and its Applications (Proc. Conf., Oxford, 1969)*, pages 221–244. Academic Press, London, 1971. (Cited on p. 5)

[Vel19] Daniel Velleman. *How to Prove It: A Structured Approach*. Cambridge University Press, 2019. (Cited on p. 4)