Synthesis Lectures on
Mathematics & Statistics

SYNTHESIS
COLLECTION OF TECHNOLOGY

George Tourlakis

Discrete Mathematics

A Concise Introduction

Springer

Synthesis Lectures on Mathematics & Statistics

This series includes titles in applied mathematics and statistics for cross-disciplinary STEM professionals, educators, researchers, and students. The series focuses on new and traditional techniques to develop mathematical knowledge and skills, an understanding of core mathematical reasoning, and the ability to utilize data in specific applications.

George Tourlakis

Discrete Mathematics

A Concise Introduction

Springer

George Tourlakis
Department of Electrical Engineering
and Computer Science
York University
Toronto, ON, Canada

ISSN 1938-1743 ISSN 1938-1751 (electronic)
Synthesis Lectures on Mathematics & Statistics
ISBN 978-3-031-30487-3 ISBN 978-3-031-30488-0 (eBook)
https://doi.org/10.1007/978-3-031-30488-0

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

για την Δέσποινα

Preface

This volume is an introduction to *discrete mathematics*, an area of study that is required by most computer science and computer, software as well as electrical engineering curricula. It is also often prescribed for mathematics majors as the first course where you "do mathematics with proofs".

Discrete mathematics studies properties of "discrete sets", finite or infinite, and their objects. The major representatives of such sets are all the finite sets, and not only the set of natural numbers, $\mathbb{N}$ —but also the set of all integers, $\mathbb{Z} = \{\ldots, -1, 0, 1, \ldots\}$— in the infinite case.

What makes $\mathbb{N}$ (or $\mathbb{Z}$ or finite sets) "discrete" is that the standard order "<" on natural numbers (or on $\mathbb{Z}$) and any order on finite sets has "gaps" between consecutive members. By contrast the standard order, also denoted by "<", on the set of reals ($\mathbb{R}$) has the property that if a and b are any two reals such that $a < b$, then there is a real c between the two: Just take $c = (a + b)/2$ and you have $a < c < b$. This is not so with natural numbers. For example, there is no c in $\mathbb{N}$ that lies between 1 and 2: For a c in $\mathbb{N}$, $1 < c < 2$ is false.

A looser definition of a "discrete" set might allow $\mathbb{Q}$, the set of rationals (fractions with integers as numerator and denominator), as a set of interest in the discipline of discrete mathematics.[1] Allowing this set in due to another of its attributes (not shared by the set of reals that is the domain of study in *calculus* and *analysis*), namely the members of $\mathbb{Q}$ can be *enumerated* using natural number indices; it is *enumerable* as we say.

We will not study the properties of $\mathbb{Q}$ in this volume.

All books on discrete mathematics, big or small (this one is deliberately small), include mostly *topics* on set theory.

The area of study denoted by the term *set theory* is vast, non-elementary, and entire volumes have been written to cover *just set theory* and nothing else (e.g., cf. Jech (1978), Levy (1979), Kunen (1980), Tourlakis (2003b)). As it turns out, most discrete mathematics texts have the majority of their topics in the area of set theory, doing so under various chapter titles.

[1] Rational numbers have no gaps with respect to the order <. If a, b are rationals, there is always a rational c between the two: For example, take $c = (a + b)/2$.

The present volume engages *all* but *two* of its chapters on set theory topics —the two exceptions being Chaps. 4 and 7. In particular, we include operations on sets, functions, and relations, finite and infinite sets, uncountable sets, and *diagonalisation* that is a must-have tool in theory of computation, equivalence relations, orders, induction and inductive (or *recursive*) *definitions*, inductively defined sets, and "structural induction". I omit advanced topics such as ordinal and cardinal *numbers* (these are omitted also in every other book on discrete mathematics that I know of).

But I do include several non-elementary —yet not utterly esoteric— topics from set theory in this volume. For example, the chapter on induction includes *induction along arbitrary relations* that are *not* orders, and also the recursive *definition* topics in this volume are introduced thoroughly and completely —*not* as it is commonly done, namely "here are some recursive definition *examples*"— to the extent that we *prove* that recursive definitions *do* (each) define a function and that said function is *unique*. Our recursive definitions include (with proofs) the case of recursion along a *non-order relation*. As an illustration, we give the example defining the so-called *support function* of set theory by induction along the non-order relation $\in$. The *support* is a function that returns the set of *atoms* that were used to build the function's set input. For example, the support of $\{\{\{3\}\}, \{\{\{1\}\}\}\}$ is $\{1,3\}$.

Notably, we also include a brief introduction to *set operators* (not to be confused with set *operations*) through which we prove the so-called *Schröder-Bernstein theorem* —which actually is due to Dedekind— that proves the difficult, but "obvious", statement "if the infinite set A has *at most* as many elements as the infinite set B, and if also B has *at most* as many members as A, then the two sets have the same number of elements".

Above all, I note that set theory —the small part that we cover— in this volume is *founded* so that is "safe", as I call it, meaning that I *build sets by stages* following the idea of Bertrand Russell in order to avoid the obvious contradictions (known as paradoxes or antinomies) of *Cantor's set theory*. For example, the universe of all sets and atoms, $\mathbb{U}$ (see Wilder (1963) for extensive historical commentary) is not a "self-contradictory set" (ibid.) in this book —in fact it is (easily, cf. Sect. 2.2) *provably* not a set, that is, it is a *proper class*. No harm done!

Other discrete mathematics texts contain much less coverage on set theory, and normally they omit *uncountable sets* and *diagonalisation*, *proofs* that induction definitions work, and the *generalisation* of induction along non-orders.

Outside set theory topics, other discrete mathematics texts are usually deficient in their presentation of predicate logic (reduced to recipes), and they normally stay away from solving recurrence equations *with* generating functions or *without*. This is a must-have (in our opinion) topic that supports sequel courses on the theory of algorithms.

Invariably, I see other texts on discrete mathematics defining functions *wrongly*: in those definitions, they require *all* functions to be *total*[2] by something like "[the function f from S to T] … for every x in S returns a unique element $y = f(x)$ in T as output …".

[2] That is, totally defined.

But this function is *total* on its "left field" S (that is, *supply* of inputs) unlike many of the functions used in *theory of computation*, for example, this one: "for all x, y return $\lfloor x/y \rfloor$". This is *undefined* for all inputs (x, y) of the form $(a, 0)$ —so is it not a "function"?

The reader will find here a rigorous, correct, and simple chapter on predicate (first-order) mathematical logic *for the user*. I have to admit a shortcut I took in my logic chapter (maybe two), which is (are) made in the interest of saving space and minimising formalism-fatigue. It is usual —given that most books on algebra or calculus or *discrete mathematics* do not offer this definition *either*— for one to adopt the belief that "one learns the syntax of formulas via practise".

Thus, I do not *define* the *syntax* of mathematical formulas in its full abstract generality within predicate logic (the reader that wishes to see this definition may consult Tourlakis (2008)), but the reader will quickly learn *by use* from Chaps. 1–3 that, e.g., $x \in A \to x \in A \cup B$, $A \in \mathbb{F} \to \bigcap \mathbb{F} \subseteq A$, and $A \subseteq A \cup B$ are formulas of interest in set theory, while $1 + 2 + 3 + \ldots + n = \dfrac{n(n+1)}{2}$ is *also* a formula of number theory (over $\mathbb{N}$).

I note that the precise "shape" (syntax) of *Boolean* formulas —but not of predicate logic formulas— *is* defined in Exercise 6.4.5 and several syntactic properties of Boolean formulas are proved (by the reader) using structural induction.

The related second shortcut is motivated by the lack of a formal definition of formula syntax! "Perhaps we can do away with the introduction of *formal* counterparts of the *truth values* **false** and **true**, namely the Boolean logic *constants* $\bot$ and $\top$" I thought! This thought is behind my elevating the *metasymbols* of (Boolean) logic **t** (true) and **f** (false) to *atomic Boolean formula* status in the logic chapter.

This is not more odd —or more unusual— than using the symbol 3 *in algebra, say, to denote* both *the* name *and the* value *of the* constant, *we pronounce "three"*.

The logic chapter explains fully, and gives several examples of, the *application* of *generalisation* and *specialisation*, the use of the "ping-pong" theorem used to prove equivalences, the *deduction theorem* and *proof by contradiction* techniques, which are introduced (and *proved* to be valid —metatheorems— in the Exercises section with the help of extensive Hints) and the (complicated) technique of the elimination of an $\exists$-prefix of a formula. This also is proved in the Exercises section with my help (Hints). The *variant theorem* (4.1.32) —or *bound variable renaming* theorem— is also proved.

What to Include? What to Omit?

My undertaking, by agreement with the editor, was for a small-length volume. This makes the above two questions very important.

I hold that absolutely central to any discrete mathematics volume —indeed, any course on the subject— is *induction* and *inductive definitions* (the latter nowadays being increasingly called "*recursive definitions*"). Then I need material to support the proper introduction of these topics, *and* I need to do it correctly without reproducing/re-bumping

into the *contradictions* (cutely called *paradoxes*[3] and *antinomies* back then) of Cantor's *set theory*. Enter "safe set theory" in the style of Russell where sets are *defined by stages* —only a short step away, this, from modern axiomatic set theory.

Then I must include *enough* set theory —to do a good job on induction— e.g., I must cover well the topics on *relations* and *functions* that lead to *induction* (along relations that are "well-orderings"[4] or at the other extreme might not even be orders but do have, as we technically say in this volume, "MC") and *recursive definitions* (again along well-orderings but also along non-order relations that have MC, in the general case).

Incidentally, as this volume aspires to serve, among others, courses on limitations of computing at the 2nd, 3rd, or 4th year level, it chooses to *steer away from the practice prevalent* in other discrete mathematics books where functions are introduced in a manner that makes them defined everywhere. Neither in practice nor in theory (of computation) are *all* functions and relations defined on *all* possible "legal" inputs.[5] Thus, our functions and relations are defined to be "*partial*", a term that allows *both* totally defined ("total") but also otherwise ("nontotal") functions and relations.

Our topics on *cardinality* are just about the right amount, however we do not cover the advanced topic of cardinal and ordinal *numbers*. We include mathematical definitions and the use of finite and infinite sets and countable and uncountable sets. It makes sense to include the topic of *diagonalisation* —invented by Cantor— which computer science students should be able to see and understand its application *before* they get into a course on the theory of computation where the concept is extensively *used*.

Computer science (and computer and software engineering) students also need to take courses on the *analysis of algorithms*, where recurrence equations are set up to compute the run times (usually worst-case upper bounds) of algorithms.

This motivates the chapter on recurrence equations —and their *closed form solutions*— *generating functions*, and *trees*, the latter not as data structures of interest (they are that too!) but rather as a *tool* towards computing some interesting but scary sums that are of use in the solution of recurrence relations. The justification for including these here —rather than hoping they will be covered in the analysis of algorithms course and do nothing— is that these solution techniques are numerous and involved, and we do not believe that they can be easily embedded as teaching topics in the course that *uses* them.

The informed reader of this preface will notice that I omit *combinatorics*, *graphs*, and *automata*.

---

[3] The catastrophic failure of a theory that leads to a *contradiction* —an *inconsistent* theory, as we say— is sugar-coated if we call it a "paradox" from the Greek παρά, that is, *against*, and δοκώ that is, I *believe* or I *know*. If it is something that only betrays our beliefs, then probably it is not that bad?

[4] This ungrammatical terminology is imposed on us by the literature.

[5] In a theory about functions with natural number inputs and natural number outputs, all natural numbers are "legal" inputs. But for some functions, *not all* legal inputs cause an output.

Some discrete mathematics books include these topics. However, automata does not fit the purposes of a book on discrete mathematics *tools*. Third year courses in software engineering develop in situ what material they need from automata theory and so do courses on compilers. Automata is not a discrete math *tools* topic on par with induction, recursive definitions, diagonalisation, relations and functions, logic, and recurrence equation solving. The topics included here *are* here because students need preparatory practice in these before one uses them in a course that follows.

But what about graph theory? In practice, *graphs* —as opposed to graph *theory* which is a very extensive subject— are normally introduced quickly where they are needed, whether it is a course in *data structures* (no more than the definition of graphs is usually needed in such a course) or *analysis of algorithms* where some topics on graph theory might be needed (paths, cycles, spanning trees).

Regarding combinatorics, if one needs to "count", then techniques other than the sophisticated one via *generating functions* (covered extensively in the present volume but almost in no other discrete mathematics book) will be covered in situ in a course where they are called for, that is, typically an *analysis of algorithms* course.

Under the above heading "what to include", I felt obliged to introduce the so-called Axiom of Choice, which guarantees that I can fit in a (finite!) proof *infinitely many choices of elements* from a set —even if there is *no obvious methodology* to describe *the sequence of my choices* in a *finite* manner. Cf. Remark 3.5.28.[6]

How much mathematical rigour and how much intuition is a good mix in a book like this? We favour both!

Intuition helps us conceptualise and formulate the elements (rough details) of solutions to mathematical problems —"napkining it", as it were, that is, just as we would do a rough calculation on a napkin. Rigour is the expression and discipline of being *mathematically careful*. Thus using rigorous arguments, the extra care that this entails might help to avoid errors.

The Chapters

Chapter 1 is very short and retells the story of the *Russell paradox*. This is our motivation to practise safe set theory, of which the exposition and foundation begin with Chap. 2.

Chapter 2 —endowed with the experience of Chap. 1— states at the outset that we have *two* types of collections: *sets* and *non sets*. *All* collections we shall call *classes* but the *non-set* ones we call *proper classes*. To *prove* that a class is a set, we use three Principles, 0, 1, and 2, of *set formation by stages*. Further, the chapter introduces the class

[6] I should note that I can make such a finite-description without the Axiom of Choice helping, if I wanted to *finitely describe* infinitely many choices from an infinite set of *natural numbers A*: Just *choose* the *smallest* a_0 in A, then, for all $n \geq 0$, choose the *smallest* a_{n+1} in $A - \{a_0, a_1, \ldots, a_n\}$. This *two-line* recursive *definition* is good for infinitely many choices.

(unordered) pair $\{A, B\}$ and proves that indeed *it is* a set, then proceeds to defining *union*, *intersection*, *difference*, *power set*, *ordered pair* —the latter *as a set* not as a new strange object— and *Cartesian product*. All the italicised terms represent operations on sets that provably produce sets.

Chapter 3 is on relations and functions. The transitive closure of a relation is included and for relations on finite sets, algorithms for its computation are proposed including the well-known Warshall's algorithm. Equivalence relations and order relations are included. The former (in an illustrative example) leads to our first acquaintance of the "least principle" on the set of natural numbers. The latter starts our study of orders that culminates to *induction* and *inductive* (recursive) *definitions* in a later chapter.

This chapter also introduces us to the concepts "finite" and "infinite" (set), diagonalisation, *operators* on sets, and the Schröder-Bernstein (or Cantor-Bernstein) theorem —which is actually Dedekind's.

Chapter 4 is an about 20-pages long chapter, which outlines the elements of predicate logic *for the user*, including the techniques of adding/removing $\forall$ and $\exists$ prefixes of formulas in proofs. It contains a good number of illustrative examples.

Chapter 5 introduces the concept of "inductiveness condition" (*IC*) of a relation and proves its equivalence to the "minimal condition" (MC) of a relation. This has as a special case that induction on the natural number set $\mathbb{N}$ is equivalent with the least principle on $\mathbb{N}$.

The statement that "$<$ has IC" is the expression of the fact that we can do induction along this $<$ to prove mathematical statements about the members of the class where $<$ acts.

The formula that expresses the "strong" or "course-of-values" induction proof principle is derived. There are many illustrative examples of induction as well as several end-of-chapter exercises. The chapter proves the theorem that recursive definitions lead to functions that uniquely exist. Both induction and recursion are extended to apply to *non-order* relations, as long as the latter have MC (equivalently IC). One such non-order relation is $\in$ and thus we not only can prove properties of sets by induction along the relation $\in$ but also can make recursive definitions along $\in$. As an illustration of the latter, the support function is discussed.

Chapter 6 introduces a generalisation of *definitions by induction* (recursion) of Chap. 5. According to this generalisation, a set is defined from a given set of *operations* —that is, *relations* $R(x_1, \ldots, x_n, y)$— where the x_i is where the inputs are "read" in and the y is where the outputs appear. The definition requires the *set* —it turns out it *is* a set— to be the $\subseteq$-smallest set that is *closed*[7] under *all given* operations R. We note that if $n = 0$, for some such operation R, then the *outputs* of R can be thought of as given *initial objects*.

[7] A set T is *closed under* a relation $R(x_1, \ldots, x_n, y)$ —by definition— iff for all specific $x_1, \ldots, x_n$ in T, all the produced y are also in T.

If the operations are all sets, then the $\subseteq$-smallest *set* so formed is unique and is called the *closure* $S = C1(\{..., R(x_1, ..., x_n, y), ...)$ of these relations. We also say that S is *inductively defined* by said operations.

The associated proof tool —*induction over a closure*, also termed *structural induction*— proves *properties* of inductively defined sets. We validate that this induction "works" in this chapter.

We also connect the *inductive definition* of sets with an appropriate *iterative* construction *by stages*, and we also connect it (in the chapter's Exercises section) with the definition of sets as *monotone operator fixpoints* (monotone operators were introduced in 3.8.1).

Chapter 7. Here, we discuss many classes of recurrence equations and the various techniques to obtain "closed form" solutions —that is, in terms of known functions such as $\lambda n.n^2$, $\lambda n.2^n$, $\lambda n.n \log_2 n$ etc.

The technique of generating functions is also outlined and demonstrated with several examples (the most nontrivial of which appearing in the following Chap. 8).

Chapter 8. In the area of mathematics known as *graph theory*, trees —in particular binary trees— play a central role as special cases of the so-called *directed graphs*. While trees are studied for their own merit in modelling important data structures in computing practise, they have also unexpected applications to discrete mathematics such as the one we will demonstrate in this chapter —using trees to compute a scary sum in closed form. The chapter concludes with an application of *generating functions* used to compute a simple expression that computes *the number of all extended trees* that have n internal nodes.

There exists a direct graph-theoretic definition of a tree —that is beyond the design of this volume— but it is arguably more convenient to offer the direct, graph-independent, recursive definition (as in Knuth (1973), but see Example 6.3.10) that we took in Chap. 6 if for no other reason, then at least for the fact that such definition enables us to prove tree properties by *structural induction*.

If I used this book in a first year undergraduate course on discrete mathematics —which I am approximately doing now and for the past few years, using a simplified *prequel* of this volume that I wrote— I would cover almost everything, except the most difficult recurrence equations (with or without the help of generating functions) and I would also skip induction and recursion along a non-order relation with MC. If by a(n unexpected) miracle discrete mathematics was taught in 2nd year (just as a course in logic, essentially, is), then I would want to cover everything. This is the advantage of a short volume; you *can* cover everything if the audience is prepared.

The reader will forgive, I hope, the many footnotes, which the "style police" may assess as "bad style"! However, there is always a story within a story that is best delegated to footnotes not to disrupt the flow of exposition. Incidentally, the book by Wilder (1963) on the foundations of mathematics would lose most of its effectiveness if it were robbed of its superbly informative footnotes!

My footnotes (unlike many in Wilder's book) are almost never of historical import, but do support the understanding of those who may be bewildered by long displayed formulas. To the rescue I often include a very *local* footnote *inside* the display to explain a potentially puzzling *spot* —and I do so *on the spot!* Consider the display below as an example.

$$x \leq k, \text{ where } k \text{ is a natural number, implies } x \leq \lceil x \rceil \leq^{8} k \tag{1}$$

The style of exposition that I prefer is informal and conversational and is expected to serve well not only the readers who have the guidance of an instructor but also those readers who wish to learn discrete mathematics on their own. I use several devices to promote understanding, such as frequent "pauses" that anticipate questions and encourage the reader to rethink an issue that might be misunderstood if *read* but *not* studied and reflected upon. All pauses start with "**Pause.**" and end with "◄".

Apropos quotes and punctuation, we follow the "logical approach" (as Gries and Schneider (1994) call it) where punctuation is put *inside* the quotation marks if and *only if* it is a logical part of the quoted text; *never* otherwise. So we would *never* write

> The relation "is a member of" is fundamental in set theory. It is denoted by "∈."

No. "." is *not* part of the symbol! We should write instead

> The relation "is a member of" is fundamental in set theory. It is denoted by "∈".

Another feature of the above reference that I have adopted is the logical use of the em-dash "—" as a *parenthesis*. As such we have a *left version* and a *right version* to avoid ambiguities. The left version is contiguous with the *following* word but *not* with the preceding word. The right version is the reverse of this. For example, "discrete mathematics is *easy* —as long as one studies— *and* is useful towards preparing you for courses in algorithms and logic".

I have included numerous remarks, examples, and embedded exercises (the latter in addition to the end-of-chapter exercises) that reflect on a preceding definition or theorem.

Influenced by my teaching —where I love emphasising things— but originally by the books of Bourbaki, I use in my books the stylised "winding road ahead" warning,  , that I first saw in Bourbaki (1966).

It delimits a passage that is too *important* to skim over.

 delimits non-elementary passages that I could not resist including.

There are 202 end-of-chapter exercises and several embedded ones in the text. Many have hints and thus I refrained from (subjectively) flagging them for "level of difficulty".

[8] $\lceil x \rceil$ is the *smallest* natural number $\geq x$.

After all, as one of my mentors, Allan Borodin, used to say to us (when I was a graduate student at the University of Toronto), "Attempt *all* exercises. The ones you can do, don't do; do the ones you *cannot do*".

Toronto, Canada George Tourlakis
February 2023

Acknowledgments I wish to thank all those who taught me, including my parents. In particular, I thank the many mentors I have had at the University of Toronto as a graduate student[9] and all those in my prehistory, in chronological order Andreas Katsaros, Yiannis Ioannidis, and Pan. Ladopoulos —all three of whom taught me geometry.

[9] I will avoid a name listing since every permutation unintentionally —but inevitably— implies a ranking.

Contents

Some Elementary Informal Set Theory

1

Overview

Set theory is due to Georg Cantor. "Elementary" in the title above does not apply to the body of his work, since he went into considerable technical depth and mathematical sophistication in this, his new theory. It applies however to *our* coverage in this volume as we are going to restrict ourselves to elementary topics *only*.

Cantor's Set Theory contains quite a few contradictions widely referred to in the literature as *paradoxes*[1] or *antinomies*,[2] some of considerable consequence. The next section is about the least technical, hence the most elementary of all to describe, and most fundamental of these antinomies contained in Cantorian set theory and was discovered by Bertrand Russell.

What caused these contradictions or *inconsistencies* as logicians call them?
The reason is that Cantor's set theory was not based on axioms nor rigid rules of reasoning — a state of affairs for a theory that we loosely characterise as "informal".

At the opposite end of "informal" we have the *formal* theories that are based on the *form* of the mathematical statements under consideration and utilise axioms *and* the rules of *mathematical logic* to formulate proofs.

As such the latter theories are "safer" to develop; they do not lead to *obvious* contradictions.

One *cannot* fault Cantor for not using logic in arguing his theorems —that process for "doing mathematics" was not yet invented when he built his theory— but then, *a fortiori*, mathematical logic was not invented in Euclid's time either, *and yet* he did use axioms that

[1] From the Greek παράδοξο. Παρά means "against" while δοκώ means "I believe" or "I know". A *paradox* thus is against one's belief or knowledge.

[2] From the Greek Αντινομία. Αντί also means "against" and νόμος means "the law". So an *antinomy* is against the (mathematical or logical) law.

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0_1

stated how his building blocks, *points*, *lines* and *planes* interacted with each other and how they behaved.

Incidentally, Euclidean Geometry (provably) *is* free of *contradiction —or, as we say differently, it is a* consistent *theory.*

The problem with Cantor's set theory is that anything goes regarding what sets *are*. A set is just a *synonym* of the dictionary terms "aggregate", "collection", "class", etc. Definition-by-dictionary of synonyms as it were.

Moreover, Cantorian set theory does not deal with the most fundamental question: "How are sets *formed*? This question has a remarkably simple answer (Russell's) that can be credited for the particular *choice of the axioms of modern axiomatic formal ZF Set Theory*.[3]

We sample in the next section the kind of logical "trouble" this extremely informal approach entails (Russell's paradox).

It must be stated at the outset that to have a set theory that has no obvious inconsistencies does not necessitate to work formally within the axiomatic method. Our first chapter, based on Russell's approach allows us to do "safe" set theory —a nickname we gave to *informal* set theory that does not have any of the known inconsistencies of Cantor's set theory— within an *informal* (non axiomatic) setting.

Following Russell we will *ask and answer* how sets are built *first*, and then derive from our answer some *principles* that will guide (and protect!) the theory's development —and, in particular, will guide us in "safely" building "large" sets; indeed any sets!

1.1 Russell's "Paradox"

Cantor's *naïve* (this adjective is not derogatory but is synonymous in the literature with *informal* and *non axiomatic*) set theory was plagued by *paradoxes*, the most famous of which (and the *least* "technical") was pointed out by Bertrand Russell and was thus nicknamed "Russell's paradox".

Cantor's theory is the theory of collections (synonymously, sets) of objects, as we mentioned above, terms that were not defined and moreover it was not indicated how these sets were built.[4]

[3] "ZF" for Zermelo and Fraenkel, the designers of the most commonly used axiomatic set theory.

[4] This is not a problem *in itself*. Euclid too did not say *what* points and lines *were*; but his axioms did characterise their nature and interrelationships: For example, he started from these (among a few others) *a priori truths* (axioms): *a unique line passes through two distinct points*; also, *on any plane, a unique line l can be drawn parallel to another line k on the plane if we want l to pass through a given point A that is not on k.*

The point is:

This theory studies operations on sets, properties of sets, and aims to use set theory as the foundation *of all mathematics*. Naturally, mathematicians "do" set theory of *mathematical object collections* —not collections of birds and other beasts. We have learnt some elementary aspects of set theory in high school. We will learn more from this book.

1. **Variables**. Like any theory, informal or not, informal set theory —a "safe"[5] variety of which we will develop here— uses *variables* just as algebra does. There is only *one type* of variable that varies over set and over atomic objects too, the latter being objects that have no set structure. For example integers. We use the names $A, B, C, \ldots$ and $a, b, c, \ldots$ for such variables, sometimes with primes (e.g., A'') or subscripts (e.g., x_{23}), or both (e.g., x''''_{22}, Y'_{42}).
2. **Notation**. *Sets given by listing*. For example, $\{1, 2\}$ is a set that contains precisely the objects 1 and 2, while $\{1, \{5, 6\}\}$ is a set that contains precisely the objects 1 and $\{5, 6\}$. The braces { and } are used to delimit at the left and right the indicated collection/set of objects by outright listing.
3. **Notation**. *Sets given by "defining property"*. But what if we cannot (or will not) explicitly list all the members of a set? Then we may define what objects x get in the set/collection by having them to *pass an entrance requirement*, $P(x)$:

   An object x gets in the set *iff* (*if and only if*) $P(x)$ is true of said object.

   Let us parse "iff":

   a. The *IF*: So, IF $P(x)$ is true, then x gets in the set (it passed the "admission requirement").
   b. The *ONLY IF*: So, IF x gets in the set, then the ***only*** *way for this to happen* is for it to pass the "admission requirement"; that is, $P(x)$ is true.

   In other words, "iff" (as we probably learnt in high school or some previous university course such as calculus) is the same thing as "is equivalent":

   "x is in the set" is equivalent to "$P(x)$ is true".

You cannot leave out *both* what the *nature* of your objects of study is *and how* they behave/interrelate and get away with it! Euclid omitted the former but provided the latter, so all worked out.

[5] Safe from contradictions, that is.

We denote the collection/set[6] defined by the entrance condition $P(x)$ by

$$\{x : P(x)\} \tag{1}$$

but also as

$$\{x \mid P(x)\} \tag{1'}$$

reading it "the set of all x *such that* (this "such that" is the ":" or "|") $P(x)$ is true [or holds]"

4. "$x \in A$" is the assertion that "object x is in the set A". Of course, this assertion may be true or false or "it depends", just like the assertions of algebra $2 = 2$, $3 = 2$ and $x = y$ are so (respectively).
5. $x \notin A$ is the negation of the assertion $x \in A$.
6. **Properties**

- Sets are *named* by letters of the Latin alphabet (cf. **Variables**, above). Naming is pervasive in mathematics as in, e.g., "let $x = 5$" in algebra.
  So we can write "let $A = \{1, 2\}$" and let "$c = \{1, \{5, 6\}\}$" to give the names A and c to the two example sets above, ostensibly because we are going to discuss these sets, and refer to them often, and it is cumbersome to keep writing things like $\{1, \{5, 6\}\}$. Names are *not permanent*;[7] they are *local* to a discussion (argument).
- **Equality of sets** (repetition and permutation do not matter!)
  Two sets A and B are equal iff they have the same members. Thus order and multiplicity do not matter! E.g., $\{1\} = \{1, 1, 1\}$, $\{1, 2, 1\} = \{2, 1, 1, 1, 1, 2\}$.
- The fundamental equivalence pertaining to definition of sets by "defining property": So, if we name the set in (1) above, S, that is,

if we say "let $S = \{x : P(x)\}$", then "$x \in S$ iff $P(x)$ is true" (1)

Incidentally, we almost *never say* "is true" unless we want to emphasise this fact. We would say instead: "$x \in S$ iff $P(x)$".
Equipped with the knowledge of the previous bullet, we see that the symbol $\{x : P(x)\}$ defines a *unique* set/collection: Well, say A and B are so defined, that is, $A = \{x : P(x)\}$ and $B = \{x : P(x)\}$. Thus

$$x \in A \overset{A=\{x:P(x)\}}{\text{iff}} P(x) \overset{B=\{x:P(x)\}}{\text{iff}} x \in B$$

[6] We have not yet reached Russell's result, so keeping an open mind and humouring Cantor we still allow ourselves to call said collection a "set".

[7] OK, there *are* exceptions: $\emptyset$ is the permanent name for the *empty set* —the set with no elements at all— and for that set only; $\mathbb{N}$ is the permanent name of the set of all *natural numbers*.

thus

$$x \in A \text{ iff } x \in B$$

and therefore $A = B$ by the way equality of sets/collections is defined.

Let us pursue, as Russell did, the point made in the boxed statement (last bullet) above. Take $P(x)$ to be specifically the assertion $x \notin x$. He then gave a name to

$$\{x : x \notin x\}$$

say, R. But then, by the referred to statement above,

$$x \in R \text{ iff } x \notin x \tag{2}$$

If we now *believe*,[8] as *Cantor*, the father of set theory who did not question and went ahead with it, that every $P(x)$ defines a *set*, *then* R *is a set*.

What is wrong with that?

Well, if R is a set then this object has the proper *type* to be substituted into the *variable of type "math object"*, namely, x, throughout the equivalence (2) above. But this yields the contradiction

$$R \in R \text{ iff } R \notin R \tag{3}$$

This contradiction is called the Russell's Paradox.

This and similar paradoxes motivated mathematicians to develop formal symbolic logic and to invent axiomatic set theory[9] as a means to avoid paradoxes like the above.

Other mathematicians who did not want to use mathematical logic and axiomatic theories found a way to do set theory *informally*, yet *safely*.

What they did was to ask *and* answer "how are sets formed?"[10]

We will look into the details of the founding this "safe" set theory in the next chapter.

[8] Informal mathematics often relies on "I know so" or "I believe" or "it is 'obviously' true". Some people call "proofs" like this —that is, baseless arguments— "proofs by intimidation". Nowadays, with the ubiquitousness of the qualifier "fake", one could also call them "fake proofs".

[9] There are many flavours or axiomatisations of set theory, the most frequently used being the "ZF" set theory, due to Zermelo and Fraenkel.

[10] Actually, axiomatic set theory —in particular, its axioms are— is built upon the answers this group came up with. This story is told at an advanced level in Tourlakis (2003b).

Safe Set Theory

2

Overview

This chapter introduces Russell's idea that *sets* are built by stages. This avoids the obvious contradictions of the naïve set theory of Cantor's that stem from situations where it allows some collections, such as $\{x : x \notin x\}$, to be "self contradictory sets" as mathematicians referred to them back then when set theory was new and contradictions were first discovered (cf. Wilder 1963).

Once we introduce the what and the how of Russell's Principles of set formation by stages and demonstrate their application by examples, we let the chapter proceed with the development of the elementary theory of sets. This theory —as introduced here— recognises that we have *two types* of collections, *sets* and *non sets*, the latter called *proper classes* in the modern literature, and we have tools to tell them apart. The "self contradictory sets" of naïve set theory go away in this setting.

We begin in this chapter the development of elementary set theory by introducing the usual operations on sets such as $\cup$, $\cap$, $\times$ that create new *sets* from given ones.

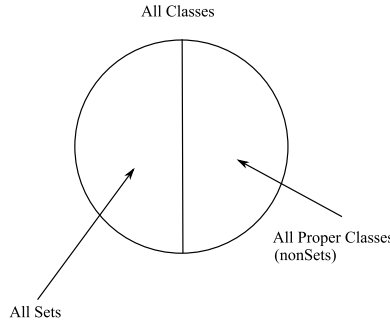
2.0.1 Definition (**Classes and sets**) We call *all* collections *classes*.

Definitions by defining property "Let $\mathbb{A} = \{x : P(x)\}$" always define a *class*, but as we saw (Sect. 1.1), sometimes —e.g., when "$P(x)$" is specifically "$x \notin x$"— that $\mathbb{A}$ is *not* a *set*.

Classes that are not sets are called *proper classes*.

The above is captured by the following picture:

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0_2



We will normally use what is known as "blackboard bold" notation and capital latin letters to denote classes by names such as $\mathbb{A}, \mathbb{B}, \mathbb{X}$. If we determine that some class $\mathbb{A}$ *is* a set, we would rather write it as A, but we make an exception for the following *sets*: Mathematicians use notation and results from set theory in their everyday practice. We call the sets that mathematicians use the "real sets" of our mathematical *intuition*, like the set of natural numbers, $\mathbb{N}$ (also denoted by ω), integers $\mathbb{Z}$, rationals $\mathbb{Q}$ and reals $\mathbb{R}$. □

In forming the class $\{x : P(x)\}$ for any property $P(x)$ we say that we apply *comprehension*. It was Frege and Cantor who believed (explicitly or implicitly) that comprehension was *safe* —i.e., always produced what they understood to be a "*set*".

But as we saw in Sect. 1.1 Russell proved that this was not the case.

> Mind you, Cantor never said what a "set" *really* is. He just relied on dictionary-derived *synonyms* —which, alas, does not settle it[a]— such as "collection" and "aggregate".
>
> Nevertheless, *very precisely*, Russell proved that *whatever you might want to call* "collections" (or "sets" for that matter) of objects that are *namable* by "$\{x : P(x)\}$"-type names it is a *mathematical fact* that there is a *choice* of at least one "$P(x)$" that *names* a "collection" that *cannot possibly* be of the *same type* as *any* of *those collections that you* just believe *you are "defining" via names such as* "$\{x : P(x)\}$"!
>
> [a] "Natural language" is neither a substitute nor an aid for the precision of mathematics.

It is a widely held tenet that set theory, using as primitives the notions of *set*, *atom* (an object that is not sub-divisible; not a collection of objects), and the relation *belongs to* ($\in$), is sufficiently strong to serve as the foundation of all mathematics. Mathematicians use notation and results from set theory in their everyday practice.

In Definition 2.0.1 we said that $\{x : P(x)\}$ always defines a class, say, $\mathbb{A}$.

Is there a converse in this observation? That is, if $\mathbb{A}$ names a class, is there *always* a "property" $P(x)$ —*whose expression does not use the letter* $\mathbb{A}$— such that $\mathbb{A} = \{x : P(x)\}$?

If $P(x)$ can refer to the letter $\mathbb{A}$ then, yes: $\mathbb{A} = \{x : x \in \mathbb{A}\}$ since this simply says "$x \in \mathbb{A}$ iff $x \in \mathbb{A}$" (cf. (1) on p. 4).

If on the other hand we heed the restriction in italics immediately above, then this converse is false. Here is why:

The term "property" is in our context, mathematically speaking, a "*formula* (of logic) in which the *only* set theory *symbol* that *need* occur is $\in$".[1]

We will later learn that we can only have *enumerably many* such formulas (properties) —meaning that we can enumerate all of them in a *straight* (infinitely long) *line* where unique positive integers are associated with —they *index*— each formula that we place on said line, and no such integer repeats in our straight line.

On the other hand we will also learn that if we consider *all the* sets *of integers*, then we *cannot* enumerate them in a similar way on a straight line. We have "more" such sets *of integers* S than we have properties $T(x)$ to *name* them without cheating[2] as $S = \{x : T(x)\}$.

2.1 The "Real Sets"

So, how can we tell, or indeed *guarantee*, that a certain class is a *set*?

Russell proposed the following "recovery" from his Paradox:

Make sure that sets are built by stages, where at stage 0 all atoms are available. Atoms are also called *urelements* in the literature from the German *Urelemente*, which in analogy with the word "*urtext*" —meaning *the earliest text*— would mean that they are the "earliest" mathematical objects available. Witness that they are available at stage 0!

We may then collect atoms to form all sorts of "first level" *sets*. We may also proceed to collect any mix of atoms and first-level sets to build new collections —second-level sets— *and so on*. Much of what set theory does is attempting to remove the ambiguity from this "and so on". See below, **Principles** 0–2.

Thus, at the beginning we have all the level-0, or type-0, objects available to us. For example, atoms such as $1, 2, 13, \sqrt{2}, \pi$ are available. At the next level we can include any number of such atoms (from none at all, to all) to build a set, that is, a new mathematical object. Allowing the usual notation, i.e., listing of what is included within braces, we may cite a few examples of level-1 sets:

L1-1. $\{1\}$.
L1-2. $\{1, 1\}$.
L1-3. $\{1, \sqrt{2}\}$.

---

[1] The multitude of symbols we use in set theory, "$\emptyset, \cap, \subseteq, \cup, \bigcup$" are all *derived* symbols —"macros" if you will— that are expressed using variables and "$\in$" only.

[2] "Cheating" would be to write $S = \{x : x \in S\}$. You see, the *informal* name "S" is not a permissible symbol to use in writing down set or atom "properties". It is neither a symbol of logic, nor is it the permissible symbol $\in$.

L1-4. $\{\sqrt{2}, 1\}$.
L1-5. $\{\pi, \sqrt{3}, \pi\}$.

We already can identify a few level-2 objects, using what (we already know) *is* available:

L2-1. $\{\{\sqrt{2}, 1\}\}$.
L2-2. $\{\{0\}, \pi\}$.


Note how the level of nesting of { }-brackets matches the level or stage of the formation of these objects!


2.1.1 Definition (**Class and set *equality***) This definition applies to any classes, hence, in particular, to any *sets* as well.

Two classes $\mathbb{A}$ and $\mathbb{B}$ are *equal* —written $\mathbb{A} = \mathbb{B}$— means

$$x \in \mathbb{A} \text{ iff } x \in \mathbb{B} \tag{1}$$

That is, an object is in $\mathbb{A}$ iff it is also in $\mathbb{B}$.

$\mathbb{A}$ is a *subclass* of $\mathbb{B}$ —written $\mathbb{A} \subseteq \mathbb{B}$— means that every element of the first class occurs also in the second, or

$$\text{If } x \in \mathbb{A}, \text{ then } x \in \mathbb{B}$$

If $\mathbb{A}$ is a set and $\mathbb{A} \subseteq \mathbb{B}$, then we say it is a *subset* of $\mathbb{B}$.

If we have $\mathbb{A} \subseteq \mathbb{B}$ but $\mathbb{A} \neq \mathbb{B}$, then we write $\mathbb{A} \subsetneqq \mathbb{B}$ (some of the literature uses $\mathbb{A} \subsetneq \mathbb{B}$ or even $\mathbb{A} \subset \mathbb{B}$ instead) and say that $\mathbb{A}$ is a *proper subclass* of $\mathbb{B}$.

Caution. In the terminology "*proper subclass*" the "proper" refers to the fact that $\mathbb{A}$ is *not all* of $\mathbb{B}$. It does *not* say that $\mathbb{A}$ is not a set! It *may* be a set and then we say that it is *proper subset* of $\mathbb{B}$. □

2.1.2 Remark

(1) Thus our definition of how classes (or sets) *compare* with respect to equality is *chosen* to be 2.1.1 —yes, this is *our choice* about what is *the* important factor for two classes to be equal; other choices are possible but not taken in the standard set theory literature. For example, $\{1\} = \{1, 1, 1\}$. Why? Because any object I see in the class to the left of "=" I also see in the class to the right, and vice versa. Similarly, $\{1, 2\} = \{2, 1\}$, for I see just "1" and "2" in the left class and I note these objects are also in the right class, and vice versa.

These two observations related to the representation of classes by listing, obtained from Definition 2.1.1, are often stated as "in a class or set depicted by listing its elements within braces, neither the order (of listing) the elements nor their multiplicity matter". Thus one will usually write $\{1, 2\}$ rather than $\{1, 2, 2, 1, 1\}$.

(2) If n is an integer-valued variable, then what do we understand by the statement "$2n$ is even"? The normal understanding is that "no matter what the value of n is, $2n$ is even", or "for all values of n, $2n$ is even".
When we get into our logic topic later on we will see that we *can* write "for all values of n, $2n$ is even" with less English as "$(\forall n)(2n$ is even)". The expression "$(\forall n)$" says "for all (values of) n".
Mathematicians often prefer to have statements like "$2n$ is even" with the "for all" *implied*.[3] You can write a whole math book without writing $\forall$ even once, and without overdoing the English.

(3) Definition 2.1.1 is called "extensionality" because it is the *extension* —that is, what members are in the two classes— that determines equality; not the *intention* —i.e., *how* the members of the two classes were selected.
For example, the two classes $\{x : x^2 - 2x + 1 = 0\}$ and $\{1\}$ are equal. Both contain just "1".

(4) Definition 2.1.1, more economically, could be stated

$$\textbf{if} \text{ we have "} x \in \mathbb{A} \text{ iff } x \in \mathbb{B}\text{", } \textbf{then} \text{ we have } \mathbb{A} = \mathbb{B}$$

The converse follows from logic needing no help from set theory concepts.

How? Well, in

$$x \in \mathbb{A} \qquad (\dagger)$$

$\mathbb{A}$ is a *name* of a mathematical object. Therefore, if the *name* $\mathbb{B}$ stands for the *same object* (i.e., $\mathbb{A} = \mathbb{B}$) then $x \in \mathbb{B}$ means exactly the same thing as $(\dagger)$.

But see also the  -passage below. □ 

 Is Definition 2.1.1 a "definition" or is it an "axiom"?

It depends:

- In a formal approach to set theory, said theory is an *extension* of predicate logic, obtained by adding the theory-specific symbol "$\in$" and adding a number of *set theory-specific*

[3] An exception occurs in Induction that we will study later, where you *fix* an n (but keep it as a variable of an *unspecified fixed value*, not as 5 or 42) and assume the "induction hypothesis" $P(n)$. But do not worry about this now!

axioms, one of which is extensionality.[4] The axioms governing the *behaviour* of equality "=" in logic are inherited by *any* theory that we base *on logic*, that is, a theory whose theorems we are proving syntactically *using logic*.

- Thus in such theories we do *not* "redefine" or "amend" what equality is and what its axiomatically postulated properties —*in logic*— are.[5]
- Here is an analogy from Peano Arithmetic (PA) —an axiomatic theory of natural numbers based on logic. It contains among others the axiom "$x + 1 = y + 1$ implies $x = y$". This axiom evidently is *not* a "definition of =" between numbers, but rather is an *axiom* about the *behaviour of the function* "+1".[6] in the *presence of equality*
  Another property of the successor is captured by the PA axiom "$x + 1 \neq 0$", and again it clearly does not "define" equality or its negation "$\neq$" —it is rather about the successor's behaviour around "=" and "0".
  Entirely analogously, *extensionality* is not about logic's "=", but rather is about how *sets*[7] *behave* around "=".
  An axiom about *sets*!
- However, our exposition of the elements of "safe set theory" is *not* axiomatic —so we do not rely on preexisting axioms for "=" from logic— thus we will side with the excellent informal but mathematically rigorous discussion of the foundations of set theory in Wilder (1963, p. 58) and take extensionality as a *definition* with no harmful side-effects. This choice is convenient as at once we "define" equality for *all* classes —that may or may not be *sets*.

2.1.3 Remark Since "iff" between two statements S_1 and S_2 means that we have *both* directions

$$\text{If } S_1, \text{ then } S_2$$

and

$$\text{If } S_2, \text{ then } S_1$$

we have that "$\mathbb{A} = \mathbb{B}$" is logically the same as (equivalent to) "$\mathbb{A} \subseteq \mathbb{B}$ and $\mathbb{B} \subseteq \mathbb{A}$". □

[4] In formal set theory if one *ever* speaks of classes (e.g., Levy 1979; Tourlakis 2003b) then one does so informally and only for convenience. Non set classes have no status within the theory. Within the theory we have only sets and atoms, and the axioms are about sets and atoms *only*.

[5] One such postulated property of "=" *in logic* is one of the usual *axioms of equality*, namely, "$x = x$".

[6] Called the *successor* function

[7] A symbolic formulation within logic of the relationship between "$A = B$" (*for sets*) and "$x \in A$ iff $x \in B$" is the "*axiom* of extensionality" of axiomatic set theory.

2.1.4 Example In the context of the "$\mathbb{A} = \{x : P(x)\}$" notation we should remark that notation-by-listing can be simulated by notation-by-defining-property: For example, $\{a\} = \{x : x = a\}$ —here "$P(x)$" is $x = a$.

Also $\{A, B\} = \{x : x = A \text{ or } x = B\}$. Let us verify the latter: Say $x \in$ lhs.[8] Then $x = A$ or $x = B$. Thus x must be A or B. But then the entrance requirement of the rhs[9] is met, so $x \in$ rhs.

Conversely, say $x \in$ rhs. Then the entrance requirement is met so we have (at least) one of $x = A$ or $x = B$. Trivially, in the first case $x \in$ lhs and ditto for the second case. □

We now postulate the principles of formation of sets!

Sets and atoms are the *mathematical objects* of our (safe) set theory.

Sets are formed by stages. At stage 0 we acknowledge the *presence* of atoms. *They are given outright, they are not built.*

Principle 0. At *any* stage Σ we *may* build a *set*, collecting together other *mathematical objects* (sets or atoms) *provided* these (mathematical) objects that we put into our set *were available at stages* before Σ.

Principle 1. *Every set is built at some stage.*

So *there is no set* in our approach that "just happens".

Principle 2. If Σ is a stage of set construction, then *there is* a stage Φ *after* it.

2.1.5 Remark (**Assumed properties of stages**) The reader would be surprised by this remark: Do we need to say more about stages? The concept of building something by stages is *intuitively clear*: At stage 0 we do this; at stage 1 we do that; at stage 3 we do something else, etc.

Note however that this impatient observation is based on stages that are (*named* by) *natural numbers*. Natural numbers have nice and well understood order properties. For example you cannot have $n < n$ if n is a natural number. In fact you cannot have an increasing chain that starts with n and ends with n. So, we do not have $n < n$. On the other hand we have that $n < m < k$ implies $n < k$.

But there are far too many sets. *More than natural numbers*, which is easy to readily agree with since we also have real numbers, a much "larger" set that contains the natural numbers.

Ergo, we need many more stages of set formation than just (those named by) natural numbers.

[8] Left Hand Side.

[9] Right Hand Side.

So we *postulate* for our stages "reasonable" and "intuitively desirable" properties below, which imitate the order properties of natural numbers, *without* attempting *to identify* stages with such numbers as this would be unnecessarily restrictive as we noted above.

Below we depict stages by the letters Σ or T with or without primes or subscripts and *postulate* as true a few intuitively pleasing properties they will have with respect to "before" and "after" relation.

We accept that the stages of set formation ordered by "before" (or "after") share the following properties with the natural numbers, the latter ordered by "less than".

Namely, let us write $\Sigma <_s \Sigma'$ for "stage Σ is *before* stage Σ'". Then we have

1. $\Sigma <_s \Sigma$ is false. That is "before" and "after" mean what *we expect them to*. No event or stage can occur before (or after) itself.
2. If $\Sigma <_s \Sigma' <_s \Sigma''$, then $\Sigma <_s \Sigma''$. No surprises here either: the expected transitivity of before and after relations.
3. If Σ, Σ' are stages, then we have one of $\Sigma <_s \Sigma'$, $\Sigma = \Sigma'$, $\Sigma' <_s \Sigma$. *We expect to be able to tell* if a stage is before another (or after, or are the same), else how will we be able to assert that a class that we just built was built *after* all its members?
4. If Σ is any stage, then there is a stage Σ' after it: $\Sigma <_s \Sigma'$ (this repeats Principle 2).

Principle 2 (equivalently, 4. above) makes it clear that we have infinitely many stages of set formation in our toolbox. Indeed, starting with any Σ, by repeated application of said Principle we can build an infinite ascending sequence

$$\Sigma <_s \Sigma' <_s \Sigma'' <_s \dots \tag{1}$$

All members in (1) are distinct, else one, say Σ_a, repeats. We then have

$$\Sigma_a <_s T <_s T' <_s \dots <_s \Sigma_a$$

By repeated application of 2. we get $\Sigma_a <_s \Sigma_a$, which contradicts 1. □

2.1.6 Remark If some set is definable ("buildable") at some stage Σ, then it is also definable at any later stage as well, as **Principle** 0 makes clear.

The informal set-formation-by-stages will guide us to build, safely, all the sets we may need in order to do mathematics. □

In axiomatic set theory ZFC[10] —just as in "small" tasks where we use natural numbers as "stages"— one *defines* stages beyond natural numbers to be certain "infinite numbers" called *ordinals*. See, for example, Tourlakis (2003b).

2.2 What Caused Russell's Paradox

How would the set-building-by-stages doctrine avoid Russell's paradox?

Recall that *à la Cantor* we get a paradox (contradiction) because we insisted to believe that all classes are sets, that is, following Cantor we "believed" Russell's "R" was a *set*.

Principles 0–2 allow us to know a priori that R is a proper class. No contradiction!

How so?

OK, is $x \in x$ true or false? Is there *any* mathematical object x —say, A— for which the following *is* true?

$$A \in A? \tag{1}$$

Well, for atom A, (1) is false since atoms have no set structure, that is, they are not collections of objects. An atom A *cannot contain anything*, in particular it cannot contain A.

What if A is a set and $A \in A$? Then in order to build A, the *set*, we have to wait until *after* its member, A is built (Principle 0 says "*provided*"). So, we need (the left) A to be built *before* we can build (the right) A in (1) as a set. In short, since the left and right A are the same, we want A build before A is built. Preposterous!

So (1) is *false*. A being *arbitrary*, we demonstrated that

$$x \in x \text{ is false (for all } x) \tag{2}$$

thus $x \notin x$ is true (for all x), therefore the R of Sect. 1.1 is $\mathbb{U}$, the universe of *all sets and atoms*; *the class of everything*.

"Everything" with restrictions in the *modern literature*. Our classes are allowed to contain only *atoms* and *sets*. *Not* proper classes.

Of course, Cantorian set theory had no such restriction since it did not distinguish between set and non-set classes to begin with.

[10] As founded by Zermelo and Fraenkel, with the axiom of Choice.

Thus

$$R = \mathbb{U}$$

Here is now an *exact* reason why $\mathbb{U}$ is *not* a set. Well, *assume for a moment that it is*.

Then

- $\mathbb{U} \in \mathbb{U}$ since the rhs contains all sets *and* we *temporarily assumed* the lhs to be a set.
- But we just saw that the above is false taking x to be $\mathbb{U}$ in (2) above.

2.2.1 Remark The immediate reactions of the mathematical community to Russell's paradox was to blame "size": "R is too big to be a set". Well, *define* "too big"!

They did not. The discussion of the panic that ensued is outlined in Wilder (1963) in a very illuminating manner. He points out (loc. cit.) that even the phrase "all sets" was viewed with suspicion, not only the dangerous act of *collecting* "all sets"!

But why did Russell bother to define his R? Why did he not use

$$\mathbb{U} = \{x : x = x\}$$

to collect all sets and prove *directly*, as we did above, that $\mathbb{U}$ *cannot* be a set, thus demonstrating in this alternative way that not all "defining properties" lead to sets?

Because his idea that sets should be build by stages was suggested *later*. Incidentally, the "too big" $\mathbb{U}$ —if any collection qualifies for the label "too big" surely the one that contains everything does!— was also discovered (without showing $R = \mathbb{U}$) to *not* be a set in a roundabout longish manner that I cannot reproduce this early in our development.

I promise to come back to this "paradox of the *powerset*" —to be defined later. You see, $\mathbb{U}$ being an omni-container contains its powerset as an element *and* as a subset. A so-called cardinality argument then derives a contradiction to the claim that $\mathbb{U}$ is a set. □

So $\mathbb{U}$, and R, are *proper* classes. Thus, the fact that R is not a set is neither a surprise, nor paradoxical. It is just a proper class as we just have recognised.

2.3 Some Useful Sets

2.3.1 Example (**Pair**) By Principle 0, if A and B are sets or atoms, then let A be available at stage Σ and B at stage Σ'. Without loss of generality say Σ' is not later than Σ —recall postulate 3. about the relative positions of two stages.

Let then pick a stage Σ'' *after* Σ (Principle 2). This will be after both (postulate 2. on p.14) Σ, Σ'.

At stage Σ'' we can build

$$\{A, B\} \tag{1}$$

as a *set* (cf. Principle 0).

We call (1) the (unordered) *pair set*.

Pause. Why "unordered"? See Remark 2.1.2, item 1.◀ □

We have just proved a theorem above:

2.3.2 Theorem *If A, B are sets or atoms, then $\{A, B\}$ is a set.*

2.3.3 Exercise Without referring to stages in your proof, prove that if A is a set or atom, then $\{A\}$ is a set.

Incidentally, a set that contains exactly one element is called a *singleton*. □

2.3.4 Remark A very short digression into Boolean Logic —for now. It will be convenient to use *truth tables* to handle many simple situations that we will encounter where "logical connectives" such as "not", "and", "or", "implies" and "is equivalent" enter into our arguments.

We will put on record here how to handle things such as "S_1 and S_2", "S_1 implies S_2", etc., where S_1 and S_2 stand for two arbitrary statements of mathematics. In the process we will introduce the *mathematical symbols* for "and", "implies", etc.

The symbol translation table from English to symbol (and back) is:

NOT	$\neg$
AND	$\wedge$
OR	$\vee$
IMPLIES (IF…,THEN)	$\rightarrow$
IS EQUIVALENT	$\equiv$

The truth table below has a simple reading. For *all possible* truth values —true/false, in short **t/f**— of the "simpler" statements S_1 and S_2 we indicate the computed truth value of the compound (or "more complex)" statement that we obtain when we apply one or the other Boolean connective of the previous table.

S_1	S_2	$\neg S_1$	$S_1 \wedge S_2$	$S_1 \vee S_2$	$S_1 \rightarrow S_2$	$S_1 \equiv S_2$	$S_2 \rightarrow S_1$
f	**f**	**t**	**f**	**f**	**t**	**t**	**t**
f	**t**	**t**	**f**	**t**	**t**	**f**	**f**
t	**f**	**f**	**f**	**t**	**f**	**f**	**t**
t	**t**	**f**	**t**	**t**	**t**	**t**	**t**

Comment. All the computations of truth values satisfy our intuition, except perhaps that for "$\rightarrow$":

$\neg$ flips the truth value as it should, $\wedge$ is eminently consistent with common sense as it applies to "and", $\vee$ is the "inclusive or" of the mathematician, and $\equiv$ is just equality on the set $\{\mathbf{f}, \mathbf{t}\}$, as it should be.

The "uneasiness" with this so-called "classical" $\rightarrow$ is that there is no *causality* from left to right. The only "easy to understand" entry is for $\mathbf{t} \rightarrow \mathbf{f}$. The outcome should be false, that is, indicating a "bad implication": You see, we have a *true* hypothesis but a *false* conclusion while, intuitively, a "good" implication ought to *preserve truth*. This implication must be "broken", so we entered **f**.

But what I just said about the case $\mathbf{t} \rightarrow \mathbf{f}$ indicates that $\rightarrow$ is meant to preserve truth from left to right. But that it *precisely does* as per table!

Here is the full picture for $\rightarrow$:

- Row one is the "no counterexample" case. That is, I claim that truth *was* preserved since *there was no truth* (left of $\rightarrow$) to preserve anyway! You have no counterexample to what I said. For that you *need* a **t** to the left and a **f** to the right of $\rightarrow$.
- In the second row we are good! We got **t** without lifting a finger!
- In the last row truth *is* preserved left to right!
- As for row three, we made our case already.

Practical considerations. Thus

1. If you want to demonstrate that $S_1 \vee S_2$ is true, for any component statements S_1, S_2, then show that *at least one* of the S_1 and S_2 is true.
2. If you want to demonstrate that $S_1 \wedge S_2$ is true, then show that *both* of the S_1 and S_2 are true.

   Note, incidentally, if we *know* that $S_1 \wedge S_2$ is true, then the truth table guarantees that each of S_1 and S_2 *must* be true.

3. If now you want to show the implication $S_1 \rightarrow S_2$ is true, then the only real work is to show that *if we assume* S_1 is true, *then* S_2 *is true too*.
   If S_1 *is known to be false, then* no work is required *to prove the implication because of the first two lines of the truth table*!
4. If you want to show $S_1 \equiv S_2$, then —because the last three columns show that this is equivalent to (same truth values as) $\Big(S_1 \rightarrow S_2\Big) \wedge \Big(S_2 \rightarrow S_1\Big)$— you just prove *each* of the two implications $S_1 \rightarrow S_2$ and $S_2 \rightarrow S_1$. □ 

From the truth table we see that we have one unary (takes one argument) and four binary (they take two arguments each) Boolean connectives. We can cascade the operations the connectives indicate to obtain more complex expressions, such as $(S_1 \vee S_2) \wedge S_3$, $(S_1 \wedge S_2) \vee S_3$, $(S_1 \to S_2) \to S_3$.

Do we always have to carry as many brackets as in the examples immediately above?

> Well, as a rule never remove brackets that you or someone else that understands logic cannot restore correctly.

By *agreeing* on the "strengths" or "priorities" of the connectives we often can get away with fewer brackets iff *we have an algorithm using which we can restore the ones we remove to their* original *positions*. The usual agreement is that the unary "$\neg$" is strongest (has highest priority) and the binary connectives follow (see below) from left to right in order of decreasing priority.

$$\neg, \wedge, \vee, \to, \equiv \tag{$\dagger$}$$

Equipped with these priorities we can reinsert brackets *correctly* if anyone has removed them *correctly*.

2.3.5 Example

1. Consider $(S_1 \vee S_2) \wedge S_3$. We cannot remove the brackets we see in this example, for if we did, then the strengths of the connectives would suggest we reinsert them this way

   $$S_1 \vee (S_2 \wedge S_3) \tag{$\ddagger$}$$

   because, in the contest between $\vee$ and $\wedge$ to win over S_2, $\wedge$ wins in ($\ddagger$) while $\vee$ wins in the original (as the brackets override the priorities).
   But ($\ddagger$) is not correct. How do we determine *incorrect*ness? By finding truth values for the S_i —can you find them?— that lead to distinct results in the original versus ($\ddagger$).
2. $(S_1 \wedge S_2) \vee S_3$. This simplifies to $S_1 \wedge S_2 \vee S_3$ in a reversible manner, since the priority of $\wedge$ (vs. $\vee$) allows us to reinsert the missing brackets.
3. $(S_1 \to S_2) \to S_3$. Can we simplify this, by, say, removing the brackets? This example amplifies the fact that the priorities are chosen by *agreement*.
   So if we did remove the brackets, how would we reinsert them? Well, the standard agreement when the *same* connective fights to win an S_i, as in

   $$S_1 \to S_2 \to S_3 \tag{¶}$$

   is to let the one to the right always win. That is, if we have a chain connected using the same connective throughout we insert brackets from right to left. Thus here we would say that brackets would have to be inserted this way

$$S_1 \to (S_2 \to S_3) \qquad (\|)$$

So? Is the above different from the formula in the beginning of 3?
Yes! Find truth values for the S_i so that the overall truth values of the formula in the beginning of 3 and ($\|$) are different. Thus, the bracket removal we contemplated in the 2nd sentence of 3 is incorrect. □

An important variant of $\to$ and $\equiv$. Pay attention to this point since almost *everybody* gets it wrong! In the literature and in the interest of creating a usable shorthand many practitioners of mathematical writing use notation like

$$S_1 \to S_2 \to S_3 \qquad (1)$$

attempting to convey the meaning

$$(S_1 \to S_2) \wedge (S_2 \to S_3) \qquad (2)$$

Alas, (2) is not the same as (1)![11]

Back to $\to$-chains like (1) versus chains like (2): Take S_1 to be **t** (true), S_2 to be **f** and S_3 to be **t**. Then (1) is true because in a chain using the same Boolean connective *we put brackets from right to left*: (1) is $S_1 \to (S_2 \to S_3)$ and evaluates to **t**, while (2) evaluates clearly to false (**f**) since $S_1 \to S_2 = \mathbf{f}$ and $S_2 \to S_3 = \mathbf{t}$.

So we need a special symbol to denote (2) correctly. We need a *conjunctional implies*! Most people use $\Longrightarrow$ for that:

$$S_1 \Longrightarrow S_2 \Longrightarrow S_3 \qquad (3)$$

that means, *by definition*, (2) above. Incidentally, a conjunctional implication "$\Longrightarrow$" is, we say, an implication *used conjunctionally*.

Similarly, $\equiv$ is *not* conjunctional, it is associative. That is,

$$S_1 \equiv S_2 \equiv S_3 \qquad (4)$$

means *equivalently*, as one can check from the truth tables,

[11] Logic does not have the sole privilege of being abused. So does plain arithmetic, from High School onwards: One often writes $a < b < c$ but they mean $a < b \wedge b < c$! This is wrong!

An amusing example from PL/1 —"Programming Language One"— an old programming language that incorporates Algol *and* Cobol (!) *and* SNOBOL (!) among others is based on the flexibility of this language in its handling of different data types. It *converts* from one data type to the other readily, without error messages. In particular, the logical "true" constant (what we call "**t**" in this book) is essentially —I am avoiding tedious details that are immaterial here— the number "1". Thus it allows, say, $6 > 5 > 3$ as a condition. PL/1 evaluates expressions from left to right and $6 > 5$ is evaluated first and returns 1 (true). Then $1 > 3$ is evaluated and returns false.

Try this in your familiar programming language and see what happens!

$$(S_1 \equiv S_2) \equiv S_3 \tag{4'}$$

or

$$S_1 \equiv (S_2 \equiv S_3) \tag{4''}$$

where the brackets indicate "which $\equiv$ applies first".

On the many occasions that we we may want to chain two or more equivalences with the intention that the chain means that *all* the equivalences are true, we use a conjunctional "$\equiv$" denoted by $\Longleftrightarrow$.

Thus

$$S_1 \Longleftrightarrow S_2 \Longleftrightarrow S_3 \tag{5}$$

means that all equivalences $S_i \equiv S_{i+1}$ —for $i = 1, 2$— are true. "$\Longleftrightarrow$" is the conjunctional "$\equiv$".

Note that the notation $\Longleftrightarrow$ is not offered just for the sake of notation.

The two notations do have *distinct* meanings. For example, if the truth values of S_1, S_2 and S_3 are **f**, **f** and **t**, respectively, then (4) computed either as (4″) or the equivalent (4′) yields the value **t**.

On the other hand, evaluating (5), that is, (5′) below for the same truth values of the S_i yields the value **f**.

$$(S_1 \equiv S_2) \wedge (S_2 \equiv S_3) \tag{5'}$$

So how do we denote (5) correctly without repeating the consecutive S_2's and omitting the implied "$\wedge$"? This way:

$$S_1 \Longleftrightarrow S_2 \Longleftrightarrow S_3 \tag{4}$$

By definition, "$\Longleftrightarrow$" is conjunctional: It applies to two statements —S_i and S_{i+1}— only and implies an $\wedge$ before the adjoining next similar equivalence.

2.3.6 Theorem *(***The subclass theorem***) Let $\mathbb{A} \subseteq B$ (B a set). Then $\mathbb{A}$ is a set.*

Proof Well, B being a set there is a stage Σ when it is built (Principle 1). By Principle 0, *all members of B* are available or built before stage Σ.

But by $\mathbb{A} \subseteq B$, all the members of $\mathbb{A}$ are among those of B.

Hey! By Principle 0 we can build $\mathbb{A}$ at stage Σ, so *it is a set*. □

Some corollaries are useful:

2.3.7 Corollary *(***Modified comprehension I***) If for all x we have*

$$P(x) \rightarrow x \in A \tag{1}$$

for some set A, then $\mathbb{B} = \{x : P(x)\}$ is a set.

Proof I will show that $\mathbb{B} \subseteq A$, that is,

$$x \in \mathbb{B} \to x \in A$$

Indeed (see 3. under **Practical considerations** in Remark 2.3.4), let $x \in \mathbb{B}$. Then $P(x)$ is true, hence $x \in A$ by (1). Now invoke Theorem 2.3.6. □

2.3.8 Corollary *(***Modified comprehension II***) If A is a set, then so is $\mathbb{B} = \{x : x \in A \wedge P(x)\}$ for any property $P(x)$.*

Proof The defining property here is "$x \in A \wedge P(x)$". This implies $x \in A$ —by 2. in Remark 2.3.4— that is, we have

$$(x \in A \wedge P(x)) \to x \in A$$

Now invoke Corollary 2.3.7. □

2.3.9 Remark (***The* empty set**) The class $\mathbb{E} = \{x : x \neq x\}$ has no members at all; it is empty. Why? Because

$$x \in \mathbb{E} \equiv x \neq x$$

but the condition $x \neq x$ is always false, therefore so is the statement

$$x \in \mathbb{E} \tag{1}$$

Nothing is permitted to enter $\mathbb{E}$.

Is the class $\mathbb{E}$ a set?

Well, take $A = \{1\}$. This is a set as the atom 1 is given at stage 0, and thus we can construct the *set* A at stage 1.

Note that, by (1) and 3. in Remark 2.3.4 we have that

$$x \in \mathbb{E} \to x \in \{1\}$$

is true (for all x). That is, $\mathbb{E} \subseteq \{1\}$.

By Theorem 2.3.6, $\mathbb{E}$ *is a set*.

But is it unique so we can justify the use of the definite article "the"? Yes. The specification of the empty set is a class with no members. So if D is another empty set, then we will have $x \in D$ always being false. But then

$$x \in \mathbb{E} \equiv x \in D \text{ (both sides of } \equiv \text{ are false)}$$

and we have $\mathbb{E} = D$ by Definition 2.1.1.

The *unique* empty set is denoted by the symbol $\emptyset$ in the literature. □

2.4 Operations on Classes and Sets

The reader probably has seen before (perhaps in calculus) the operations on sets denoted by $\cap$, $\cup$, $-$ and others. We will look into them in this section.

2.4.1 Definition (**Intersection of two classes**) We define for any classes $\mathbb{A}$ and $\mathbb{B}$

$$\mathbb{A} \cap \mathbb{B} \overset{Def}{=} \left\{x : x \in \mathbb{A} \wedge x \in \mathbb{B}\right\}$$

We call the operator $\cap$ *intersection* and the result $\mathbb{A} \cap \mathbb{B}$ the intersection of $\mathbb{A}$ and $\mathbb{B}$.

If $\mathbb{A} \cap \mathbb{B} = \emptyset$ —which happens precisely when the two classes have no common elements— we call the classes *disjoint*.

It is meaningless to have $-$ *operate on atoms.*[12] □

We have the easy theorem below:

2.4.2 Theorem *If B is a set, as its notation suggests, then $\mathbb{A} \cap B$ is a set.*

Proof I will prove $\mathbb{A} \cap B \subseteq B$ which will rest the case by Theorem 2.3.6. So, I want

$$x \in \mathbb{A} \cap B \to x \in B$$

To this end, let then $x \in \mathbb{A} \cap B$ (cf. 3. in 2.3.4). This says that $x \in \mathbb{A} \wedge x \in B$ is true, so $x \in B$ is true. □

2.4.3 Corollary *For sets A and B, $A \cap B$ is a set.*

2.4.4 Definition (**Union of two classes**) We define for any classes $\mathbb{A}$ and $\mathbb{B}$

$$\mathbb{A} \cup \mathbb{B} \overset{Def}{=} \left\{x : x \in \mathbb{A} \vee x \in \mathbb{B}\right\}$$

We call the operator $\cup$ *union* and the result $\mathbb{A} \cup \mathbb{B}$ the union of $\mathbb{A}$ and $\mathbb{B}$.

It is meaningless to have $\cup$ *operate on atoms.* □

[12] The definition expects $\cap$ to *operate on classes*. As we know, atoms (by definition) *have no set/class structure* thus no class and no set is an atom.

2.4.5 Theorem *For any* sets *A and B, $A \cup B$ is a set.*

Proof By assumption say A is built at stage Σ while B is built at stage Σ'. Without loss of generality (in short, "wlg") say Σ is no later than Σ', that is, $\Sigma \leq \Sigma'$.

By Principle 2 I can pick a state $\Sigma'' > \Sigma'$, thus

$$\Sigma'' > \Sigma' \tag{1}$$

and

$$\Sigma'' > \Sigma \tag{2}$$

Let us pick any item $x \in A \cup B$:

I have two (not necessarily mutually exclusive) cases (by Definition 2.4.4):

- $x \in A$. Then x was available or built[13] at a stage $< \Sigma$,

$$\text{hence, by (2), } \underline{x \text{ is available } \textit{before } \Sigma''} \tag{3}$$

- $x \in B$. Then x was available or built at a stage $< \Sigma'$,

$$\text{hence, by (1), } \underline{x \text{ is available } \textit{before } \Sigma''} \tag{4}$$

In either case, (3) or (4), the arbitrary x from $A \cup B$ is built before Σ'', so we can collect all those x-values at stage Σ'' in order to form a *set*: $A \cup B$. □

2.4.6 Definition (**Difference of two classes**) We define for any classes $\mathbb{A}$ and $\mathbb{B}$

$$\mathbb{A} - \mathbb{B} \stackrel{Def}{=} \left\{ x : x \in \mathbb{A} \wedge x \notin \mathbb{B} \right\}$$

We call the operator "$-$" (set-theoretic) *difference* and the result $\mathbb{A} - \mathbb{B}$ the difference of $\mathbb{A}$ and $\mathbb{B}$, in that order.

It is meaningless to have $-$ operate on atoms. □

2.4.7 Theorem *For any* set *A and class $\mathbb{B}$, $A - \mathbb{B}$ is a set.*

Proof The reader is asked to verify that $A - \mathbb{B} \subseteq A$. We are done by Theorem 2.3.6. □

Notation. The definitions of $\cap$ and $-$ suggest a shorter notation for the rhs for $\mathbb{A} \cap \mathbb{B}$ and $\mathbb{A} - \mathbb{B}$. That is, respectively, it is common to write instead

[13] As x may be an atom, we allow the *possibility* that it was available *with no building involved*, hence we said "available or built". For A and B though we are told they are *sets*, so they *were built* at some stage, by Principle 1!

$$\left\{x \in \mathbb{A} : x \in \mathbb{B}\right\}$$

and

$$\left\{x \in \mathbb{A} : x \notin \mathbb{B}\right\}$$

2.4.8 Exercise Demonstrate —using Definition 2.4.1— that for any $\mathbb{A}$ and $\mathbb{B}$ we have $\mathbb{A} \cap \mathbb{B} = \mathbb{B} \cap \mathbb{A}$.

Hint. There are two parts in the proof:

1. Assume that $x \in \mathbb{A} \cap \mathbb{B}$. Prove that $x \in \mathbb{B} \cap \mathbb{A}$.
2. Assume that $x \in \mathbb{B} \cap \mathbb{A}$. Prove that $x \in \mathbb{A} \cap \mathbb{B}$.

For 1. and 2. use Remark 2.3.4, practical considerations, 3. □

2.4.9 Exercise Demonstrate —using Definition 2.4.4— that for any $\mathbb{A}$ and $\mathbb{B}$ we have $\mathbb{A} \cup \mathbb{B} = \mathbb{B} \cup \mathbb{A}$. □

2.4.10 Exercise By picking two particular very small sets A and B show that $A - B = B - A$ *is not true* for all sets A and B.

Is it true of all classes? □

Let us generalise unions and intersections next. First a definition:

2.4.11 Definition (**Family of sets**) A class $\mathbb{F}$ is called a *family of sets* iff it contains no atoms. The letter F is here used generically, and a family may be given any name, usually capital. □

2.4.12 Example Thus, $\emptyset$ is a family of sets; the empty family.

So are $\{\{2\}, \{2, \{3\}\}\}$ and $\mathbb{V}$, the latter given by

$$\mathbb{V} \stackrel{Def}{=} \left\{x : x \text{ is a set}\right\}$$

Incidentally, as $\mathbb{V}$ contains *all* sets (but no atoms!) it is a proper class! Why? Well, if it is a set, then it is *equal* to one of the x-values that we are collecting, thus $\mathbb{V} \in \mathbb{V}$. But we saw that this statement is false *for sets*!

2.4.13 Exercise Is $\mathbb{A} \in \mathbb{A}$ also false for proper classes $\mathbb{A}$? Why? □

Here are some classes that are *not* families: $\{1\}$, $\{2, \{\{2\}\}\}$ and $\mathbb{U}$, the latter being the universe of all objects —sets *and* atoms— and equals Russell's "R" as we saw in Sect. 2.2. These all are disqualified from being "families" as they contain atoms. □

2.4.14 Definition (**Intersection and union of families**) Let $\mathbb{F}$ be a family of sets. Then

(i) the symbol $\bigcap \mathbb{F}$ denotes the class that contains *all the objects* that *are common to all* $A \in \mathbb{F}$.
In symbols the definition reads:

$$\bigcap \mathbb{F} \stackrel{Def}{=} \left\{ x : \text{for all } A,\ A \in \mathbb{F} \rightarrow x \in A \right\} \tag{1}$$

(ii) the symbol $\bigcup \mathbb{F}$ denotes the class that contains *all the objects* that *are found among the various* $A \in \mathbb{F}$. That is, imagine that the members of *each* $A \in \mathbb{F}$ are "emptied" into a single —originally empty— container $\{\ldots\}$. The class we get this way is what we denote by $\bigcup \mathbb{F}$.
In symbols the definition reads (and arguably is clearer):

$$\bigcup \mathbb{F} \stackrel{Def}{=} \left\{ x : \text{for some } A,\ A \in \mathbb{F} \wedge x \in A \right\} \tag{2}$$

□

2.4.15 Example Let $\mathbb{F} = \{\{1\}, \{1, \{2\}\}\}$. Then emptying all the contents of the members of $\mathbb{F}$ into some (originally) empty container we get

$$\{1, 1, \{2\}\} \tag{3}$$

This is $\bigcup \mathbb{F}$.

Would we get the same answer from the mathematical definition (2)? Of course:

1 *is* in some member of $\mathbb{F}$, indeed in both of the members $\{1\}$ and $\{1, \{2\}\}$, and in order to emphasise this I wrote two copies of 1 in (3) —it is emptied/contributed twice. Then $\{2\}$ is the member that only $\{1, \{2\}\}$ of $\mathbb{F}$ contributes.

What is $\bigcap \mathbb{F}$? Well, only 1 is common between the two sets —$\{1\}$ and $\{1, \{2\}\}$— that are in $\mathbb{F}$. So, $\bigcap \mathbb{F} = \{1\}$. □

2.4.16 Exercise

1. Prove that $\bigcup \left\{ A, B \right\} = A \cup B$.
2. Prove that $\bigcap \left\{ A, B \right\} = A \cap B$.

Hint. In each of part 1. and 2. show that lhs $\subseteq$ rhs and rhs $\subseteq$ lhs (cf. Remark 2.3.4, **practical considerations**, 3. and 4.). For that analyse membership, i.e., "assume $x \in$ lhs and prove $x \in$ rhs", and conversely (cf. Definition 2.1.1 and Remark 2.1.3.) □

2.4.17 Theorem *If the* set F *is a family of sets, then* $\bigcup F$ *is a set.*

Proof Let F be built at stage Σ. Now,

$$x \in \bigcup F \equiv x \in \overset{\text{some}}{\overset{\downarrow}{A}} \in F$$

Thus x is available or built before A which is built before stage Σ since that is when F was built. x being arbitrary, all members of $\bigcup F$ are available/built before Σ, so we can build $\bigcup F$ as a set at stage Σ. □

2.4.18 Theorem *If the class* $\mathbb{F} \neq \emptyset$ *is a family of sets, then* $\bigcap \mathbb{F}$ *is a set.*

Proof By assumption there *is* some set in $\mathbb{F}$. Fix one such and call it D.

First note that

$$x \in \bigcap \mathbb{F} \rightarrow x \in D \qquad (*)$$

Why? Because (i) of Definition 2.4.14 says that

$$x \in \bigcap \mathbb{F} \equiv \text{for all } A \in \mathbb{F} \text{ we have } x \in A$$

Well, D *is* one of those "A" sets in $\mathbb{F}$, so if $x \in \bigcap \mathbb{F}$ then $x \in D$. We established $(*)$ and thus we established

$$\bigcap \mathbb{F} \subseteq D$$

by Definition 2.1.1. We are done by Theorem 2.3.6. □

2.4.19 Remark What if $\mathbb{F} = \emptyset$? Does it affect Theorem 2.4.18? Yes, *drastically*!

In Definition 2.4.14 we read

$$\bigcap \mathbb{F} \overset{Def}{=} \left\{ x : \text{for all } A,\ A \in \mathbb{F} \rightarrow x \in A \right\} \qquad (**)$$

However, as the hypothesis (i.e., lhs) of the implication in $(**)$ is **false**, the implication itself is **true**. Thus the entrance condition "for all A, $A \in \mathbb{F} \rightarrow x \in A$" is true for *all* x and thus allows *all* objects x to get into $\bigcap \mathbb{F}$.

Therefore $\bigcap \mathbb{F} = \mathbb{U}$, the universe of *all* objects which we saw (cf. Sect. 2.2) is a proper class. □

2.4.20 Exercise What is $\bigcup F$ if $F = \emptyset$? Set or proper class? Can you "compute" which class it is precisely? □

2.4.21 Remark (**More notation**)
Suppose the family of sets Q is a *set* of sets A_i, for $i = 1, 2, \ldots, n$ where $n \geq 3$.

$$Q = \{A_1, A_2, \ldots, A_n\}$$

Then we have a few alternative notations for $\bigcap Q$:

(a)

$$A_1 \cap A_2 \cap \ldots \cap A_n$$

or, more elegantly,

(b)

$$\bigcap_{i=1}^{n} A_i$$

or also

(c)

$$\bigcap\nolimits_{i=1}^{n} A_i$$

Similarly for $\bigcup Q$:

(i)

$$A_1 \cup A_2 \cup \ldots \cup A_n$$

or, more elegantly,

(ii)

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

or also

(iii)

$$\bigcup\nolimits_{i=1}^{n} A_i$$

If the family has so many elements that all the natural numbers are needed to index the sets in the set family Q we will write

$$\bigcap_{i=0}^{\infty} A_i$$

or

$$\bigcap\nolimits_{i=0}^{\infty} A_i$$

or

$$\bigcap_{i\geq 0} A_i$$

or

$$\bigcap_{i\geq 0} A_i$$

for $\bigcap Q$ and

$$\bigcup_{i=0}^{\infty} A_i$$

or

$$\bigcup_{i=0}^{\infty} A_i$$

or

$$\bigcup_{i\geq 0} A_i$$

or

$$\bigcup_{i\geq 0} A_i$$

for $\bigcup Q$ □ 

2.4.22 Example Thus, for example, $A \cup B \cup C \cup D$ can be seen —just changing the notation— as $A_1 \cup A_2 \cup A_3 \cup A_4$, therefore it means, $\bigcup\{A_1, A_2, A_3, A_4\}$, or $\bigcup\{A, B, C, D\}$.

Same comment for $\cap$. □

Pause. How come for the case for $n = 2$ we *proved*[14] $A \cup B = \bigcup\{A, B\}$ (2.4.16) but *here* we say ($n \geq 3$) that something like the content of the previous remark and example are just *notation* (*definitions*)?

Well, we had *independent* definitions (and associated theorems re set status for each, Theorems 2.4.5 and 2.4.17) for $A \cup B$ and $\bigcup\{A, B\}$ so it makes sense to compare the two definitions *after the fact* and see if we can *prove* that they say the same thing. For $n \geq 3$ we opted to *not* give a definition for $A_1 \cup \ldots \cup A_n$ that is independent of $\bigcup\{A_1 \cup \ldots \cup A_n\}$, rather we gave the definition of the former in terms of the latter. No independent definitions, no theorem to compare the two!◄

2.5 The Powerset

2.5.1 Definition For any set A the symbol $\mathscr{P}(A)$ —pronounced the *powerset* of A— is defined to be the class

[14] Well, *you* proved! Same thing :-)

$$\mathscr{P}(A) \overset{Def}{=} \left\{x : x \subseteq A\right\}$$

Thus we collect *all* the subsets x of A to form $\mathscr{P}(A)$.

The literature most frequently uses the symbol 2^A in place for of $\mathscr{P}(A)$. □

(1) The term "power*set*" is slightly premature, but it is apt. Under the conditions of the definition —A a set— 2^A is also a *set* as we prove immediately below.

(2) We said "*all* the subsets x of A" in the definition. This is correct. As we know from Theorem 2.3.6, if $\mathbb{X} \subseteq Y$ and Y is a set, then so is $\mathbb{X}$.

2.5.2 Theorem *For any set A, its powerset $\mathscr{P}(A)$ is a set.*

Proof Let A be built at stage Σ. Then each of its members y are given or built *before* Σ.

Thus, since *every* subset x of A is a set of y-values, *every such subset x can be built at stage* Σ.

But then, just take any $\Sigma' > \Sigma$. Since all x-values (such that $x \subseteq A$) are built *before* Σ', at stage Σ' we can collect them all and build the *set* 2^A. □

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

$$\mathscr{P}(A) = \left\{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{3, 2\}, \{1, 2, 3\}\right\}$$

Thus the powerset of A has 8 elements.

We will later see that if A has n elements, for any $n \geq 0$, then 2^A has 2^n elements. This observation is at the root of the notation "2^A". □

2.5.4 Remark For any set A it is trivial (verify!) that we have $\emptyset \subseteq A$ and $A \subseteq A$. Thus, for any A, $\{\emptyset, A\} \subseteq 2^A$. □

2.6 The Ordered Pair and Finite Sequences

To introduce the concepts of cartesian product —so that, for example, plane analytic geometry can be developed within set theory— we need an object "(A, B)" that is *like* the set pair (2.3.1) in that it contains *two* objects, A and B ($A = B$ is a possibility), but in (A, B) order *and* length (in (A, B) it is two) matter!

We want $(A, B) = (A', B')$ to imply *$A = A'$ and $B = B'$. Moreover, note that (A, A) is* not *$\{A\}$! It is still an ordered* pair *but it so happens that the first and second component, of the ordered pair, are equal in this example.*

So, are we going to accept a new type of object in set theory? *Not at all*! We will build (A, B) so that it is a set!

2.6.1 Definition (**Ordered pair**) *By definition*, (A, B) is the abbreviation (short name) given below:

$$(A, B) \stackrel{Def}{=} \Big\{A, \{A, B\}\Big\} \tag{1}$$

We call "(A, B)" an *ordered pair*, and A its first *component*, while B is its second component. □

2.6.2 Remark

1. Note that $A \neq \{A, B\}$ and $A \neq \{A, A\}$, because in either case we would otherwise get $A \in A$, which is false for *sets or atoms* A. Thus (A, B) *does* contain exactly two members or *has length two*: A and $\{A, B\}$.
   Pause. We have *not* said in Definition 2.6.1 that A and B are sets or atoms. So what right do we have in the paragraph above to so declare?◀
2. What about the desired property that

$$(A, B) = (X, Y) \rightarrow A = X \wedge B = Y \tag{2}$$

   Well, *assume the lhs* of "$\rightarrow$" in (2) and prove the rhs, "$A = X \wedge B = Y$". From our truth table we know that we do the latter by proving *each* of $A = X$ and $B = Y$ true (separately).

   The lhs that we assume translates to

$$\Big\{A, \{A, B\}\Big\} = \Big\{X, \{X, Y\}\Big\} \tag{3}$$

   By the remark 1. above there are *two* distinct members in each of the two sets that we equate in (3).
   So since (3) is true (by assumption) we have (by definition of set equality) one of:

   a. $A = \{X, Y\}$ and $\{A, B\} = X$, that is, *1st listed element in lhs of "=" equals the 2nd listed in rhs; and 2nd listed element in lhs of "=" equals the 1st listed in rhs.*
   b. $A = X$ and $\{A, B\} = \{X, Y\}$.

   Now case (a) above *cannot hold*, for it leads to $A = \{\{A, B\}, Y\}$. This in turn leads to

$$\{A, B\} \in A$$

   and thus the set $\{A, B\}$ is built *before* one of its members A, which contradicts Principle 0.

Let us then work with case (b).

We have

$$\{A, B\} = \{A, Y\} \tag{4}$$

Well, all the members on the lhs must also be on the rhs. I note that A is. We have two cases.

- What if B is also equal to A? Then we have $\{B\} = \{A, Y\}$ and thus $Y \in \{B\}$ (why?). Hence $Y = B$. We showed so far $A = X$ (listed in case (b)) and $B = Y$ (proved here); great!
- Here B is *not* equal to A. But B must be in the rhs of (4), so the only way for that is $B = Y$. *All Done*! □

Worth noting as a *theorem* what we have just proved above:

2.6.3 Theorem *If* $(A, B) = (X, Y)$*, then* $A = X$ *and* $B = Y$.

But is (A, B) a set? (atom it is not, of course!) Yes!

2.6.4 Theorem (A, B) *is a set.*

Proof $(A, B) = \Big\{A, \{A, B\}\Big\}$. By Example 2.3.1, $\{A, B\}$ is set. Applying Example 2.3.1 once more, $\Big\{A, \{A, B\}\Big\}$ is a set. □

2.6.5 Example So, $(1, 2) = \{1, \{1, 2\}\}$, $(1, 1) = \{1, \{1\}\}$, and $(\{a\}, \{b\}) = \{\{a\}, \{\{a\}, \{b\}\}\}$. □

2.6.6 Remark We can extend the ordered pair to ordered *triple*, ordered *quadruple*, and beyond!

We take this approach in these notes:

$$(A, B, C) \stackrel{Def}{=} \Big((A, B), C\Big) \tag{1}$$

$$(A, B, C, D) \stackrel{Def}{=} \Big((A, B, C), D\Big) \tag{2}$$

$$(A, B, C, D, E) \stackrel{Def}{=} \Big((A, B, C, D), E\Big) \tag{3}$$

etc.

So suppose we defined what an n-tuple is, for *some fixed unspecified* n, and denote it by $(A_1, A_2, \ldots, A_n)$ for convenience. Then we *define*

$$(A_1, A_2, \ldots, A_n, A_{n+1}) \stackrel{Def}{=} \Big((A_1, A_2, \ldots, A_n), A_{n+1}\Big) \qquad (*)$$

This is an "*inductive*" or "*recursive*" definition, defining a concept ($n+1$-tuple) in terms of *a smaller instance of itself*, namely, in terms of the concept for an n-tuple, and in terms of the case $n = 2$ that we dealt with by *direct* definition (*not* in terms of the concept itself) in Definition 2.6.1.

Suffice it to say this "case of $n+1$ in terms of case of n" provides just *shorthand notation* to take the mystery out of the "etc." above. We *condense*/*codify* infinitely many definitions (1), (2), (3), … into just *two*:

- Definition 2.6.1
  and
- $(*)$

The reader has probably seen such recursive definitions before (likely in calculus and/or high school).

The most frequent example that occurs is to define, for any natural number n and any real number $a > 0$, what a^n means. One goes like this:

$$\begin{aligned} a^0 &\stackrel{Def}{=} 1 \\ a^{n+1} &\stackrel{Def}{=} a \cdot a^n \end{aligned} \qquad (1)$$

The pair of *definitions* above condenses infinitely many definitions such as

$$\begin{aligned} &a^0 = 1 \\ &a^1 = a \cdot a^0 = a \\ &a^2 = a \cdot a^1 = a \cdot a \\ &a^3 = a \cdot a^2 = a \cdot a \cdot a \\ &a^4 = a \cdot a^3 = a \cdot a \cdot a \cdot a \\ &\vdots \end{aligned}$$

into just two!

We will study *inductive definitions* and *induction* soon!

2.6.7 Exercise What would happen if we defined (in (1)) $a^0 \stackrel{Def}{=} 42$?

Caution. Should we not? Why not? Because then $a^1 = a \times a^0 = a \times 42$. *Hardly* the intended and expected value for a^1!

A correct answer to these two questions does not *prove* that $a^0 = 1$! This expression is *not* provable by logic using some axioms I forgot to mention. It is just a judicious *renaming* of "1" as "a^0". □

Before we exit this remark note that $(A, B, C) = (A', B', C')$ implies $A = A', B = B', C = C'$ because it implies

$$C = C' \text{ and } (A, B) = (A', B')$$

That is, (A, B, C) is an *ordered* triple (3-tuple).

We can also prove that $(A_1, A_2, \ldots, A_n, A_{n+1})$ is an *ordered* $n+1$-tuple, i.e.,

$$(A_1, A_2, \ldots, A_{n+1}) = (A'_1, A'_2, \ldots, A'_{n+1}) \to A_1 = A'_1 \wedge \ldots \wedge A_{n+1} = A'_{n+1} \qquad (2)$$

if we have followed (proved) the "etc." all the way to the case of $(A_1, A_2, \ldots, A_n)$ —for $k = 1, 2, 3, \ldots, n$. Then, by $(*)$, the case for $k = n+1$ —(2) above— is a straightforward application of the case for Theorem 2.6.3 where $X = (A_1, \ldots, A_n)$ and $Y = A_{n+1}$.

We will do the "etc."-argument *elegantly* once we learn induction! □

2.6.8 Definition (**Finite sequences**) An n-tuple for $n \geq 1$ is called a *finite sequence* of *length* n, where we extend the concept to a *one element sequence* —*by definition*— to be

$$(A) \stackrel{Def}{=} A$$

□

Note that now we can redefine all sequences of lengths $n \geq 1$ using again $(*)$ above, but this time with starting condition that of Definition 2.6.8. Indeed, for $n = 2$ we rediscover (A_1, A_2):

$$\text{the "new" 2-tuple pair: } (A_1, A_2) \stackrel{\text{by } (*)}{=} \Big((A_1), A_2\Big) \stackrel{\text{by } 2.6.8}{=} \Big(A_1, A_2\Big)$$

The big brackets are applications of the ordered pair defined in Definition 2.6.1, just as it was in the general definition $(*)$.

2.7 The Cartesian Product

We are ready to define classes of *ordered* pairs.

2.7.1 Definition (**Cartesian product of classes**) Let $\mathbb{A}$ and $\mathbb{B}$ be classes. Then we define

$$\mathbb{A} \times \mathbb{B} \stackrel{Def}{=} \Big\{(x, y) : x \in \mathbb{A} \wedge y \in \mathbb{B}\Big\}$$

called the *Cartesian product* of $\mathbb{A}$ and $\mathbb{B}$ *in that order*. The definition requires both sides of $\times$ to be classes. It makes no sense if one or both are atoms. □

2.7.2 Theorem *If A and B are sets, then so is $A \times B$.*

Proof By Definitions 2.7.1 and 2.6.1

$$A \times B = \left\{ \{x, \{x, y\}\} : x \in A \wedge y \in B \right\} \tag{1}$$

So, for each $\{x, \{x, y\}\} \in A \times B$ we have $x \in A$ and $\{x, y\} \subseteq A \cup B$, or $x \in A$ and $\{x, y\} \in 2^{A \cup B}$. Thus $\{x, \{x, y\}\} \subseteq A \cup 2^{A \cup B}$ and hence (changing notation) $(x, y) \in 2^{A \cup 2^{A \cup B}}$.

We have established that

$$A \times B \subseteq 2^{A \cup 2^{A \cup B}}$$

thus $A \times B$ is a set by Theorems 2.3.6, 2.4.5 and 2.5.2. □

2.7.3 Definition Mindful of the Remark 2.6.6 where $\big((A, B), C\big)$, $\big((A, B, C), D\big)$, etc. were defined, we define here $A_1 \times \ldots \times A_n$ for any $n \geq 3$ as follows:

$$\begin{aligned}
&A \times B \times C && \stackrel{Def}{=} (A \times B) \times C \\
&A \times B \times C \times D && \stackrel{Def}{=} (A \times B \times C) \times D \\
&\vdots \\
&A_1 \times A_2 \times \ldots \times A_n \times A_{n+1} && \stackrel{Def}{=} (A_1 \times A_2 \times \ldots \times A_n) \times A_{n+1} \\
&\vdots
\end{aligned}$$

$$\text{We may write } \bigtimes_{i=1}^{n} A_i \text{ for } A_1 \times A_2 \times \ldots \times A_n$$

If $A_1 = \ldots = A_n = B$ we may write B^n for $A_1 \times A_2 \times \ldots \times A_n$. □

2.7.4 Remark Thus, what we learnt in Definition 2.7.3 is, in other words,

$$\bigtimes_{i=1}^{n} A_i \stackrel{Def}{=} \left\{ (x_1, \ldots, x_n) : x_i \in A_i, \text{ for } i = 1, 2, \ldots, n \right\}$$

and

$$B^n \stackrel{Def}{=} \left\{ (x_1, \ldots, x_n) : x_i \in B \right\}$$

□

2.7.5 Theorem *If A_i, for $i = 1, 2, \ldots, n$ is a set, then so is $\bigtimes_{i=1}^{n} A_i$.*

Proof $A \times B$ is a set by Theorem 2.7.2. By Definition 2.7.3, *and in this order*, we verify that so is $A \times B \times C$ and $A \times B \times C \times D$ and ... and $A_1 \times A_2 \times \ldots \times A_n$ and ... □

If we had inductive definitions available already, then Definition 2.7.3 would simply read

$$A_1 \times A_2 \stackrel{Def}{=} \left\{(x_1, x_2) : x_1 \in A_1 \wedge x_2 \in A_2\right\}$$

and, for $n \geq 2$,

$$A_1 \times A_2 \times \ldots \times A_n \times A_{n+1} \stackrel{Def}{=} (A_1 \times A_2 \times \ldots \times A_n) \times A_{n+1}$$

Correspondingly, the proof of 2.7.5 would be far more elegant, via induction.

2.7.1 Strings or Expressions Over an Alphabet

2.7.6 Definition (**Strings over an Alphabet**) A *string* x (or *expression* or *word* or *vector*) *over an alphabet* A is just an n-tuple, all of whose components come from the same set, A.

That is, we say that "x *is a string of length* n over *the alphabet* A" meaning $x \in A^n$, for some $n > 0$ —or we simply say "x is over A". □

Traditionally, strings are written down in "string notation" without separating commas or spaces, ommiting enclosing brackets. So if $A = \{a, b\}$ we will write $aababa$ rather than (a, a, b, a, b, a).

Thus all strings over A that we spoke of already are members of $\bigcup_{i>0} A^i$.

What is the advantage of the notation $aababa$ over that of (a, a, b, a, b, a)? Well, it is more *natural*!

We write words (strings) like this "words" instead of like this "(w,o,r,d,s)" and a formula $(\exists x)x = y$ is written like this "$(\exists x)x = y$" rather than like this "$\Big((, \exists, x,), x, =, y\Big)$". However we must be careful with the bracket-less and comma-less notation: Let us start with the alphabet $A = \{1, 11\}$. Which string is denoted by "111"? Unfortunately, we can answer this in many different ways, so the notation is ill-defined. It is *ambiguous* as we say. In n-tuple notation we depict the possible meanings of "111" below:

(1, 1, 1) (length 3) or (11, 1) (length 2) or (1, 11) (length 2).

We avoid this ambiguity in notation if we choose our alphabet members so that each is *a symbol of length one*. Thus, in application in assembly programming where one uses integers base-16, rather than employing the digits

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15$$

which are mathematically necessary and sufficient for the job one avoids ambiguities by renaming the last 6 digits

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f$$

Thus the decimal-notation number 11 is denoted by "b" base-16, since 11 translates (in decimal notation) as $1 \times 16 + 1 = 17$.

We can exercise the remedy of length-1-symbols easily in practice —just as in the example above— because we normally deal with finite alphabets.

Concatenation of the strings $(a_1, \ldots, a_m)$ and $(b_1, \ldots, b_n)$ in that order, denoted as

$$(a_1, \ldots, a_m) * (b_1, \ldots, b_n)$$

is the string of length $m + n$

$$(a_1, \ldots, a_m, b_1, \ldots, b_n)$$

Clearly, concatenation as defined above is *associative*, that is, for any strings x, y and z we have $(x * y) * z = x * (y * z)$.

It is convenient to include an *empty vector* or *empty string* —also known as the *null string*— "( )" as a vector with no components and define that it is the *only member of* A^0. It has zero length.

The symbols prevalent in the literature used to denote the empty string are ϵ or λ. We will choose λ since ϵ might be confused with "$\in$".

Note that the empty string is an *ordered* empty set, so cannot be identified, nor confused, with the empty *unordered* set, $\emptyset$.

At the intuitive level, and given how concatenation was defined, we see that $x * \lambda = \lambda * x = x$ for any string x.

The set of *all strings of non zero length* over A is denoted by A^+. If we want to include λ we must include $A^0 = \{\lambda\}$. That is, all strings over A, including λ form the set

$$\bigcup_{i=0}^{\infty} A^i \tag{1}$$

We use the symbols by A^* for the set in (1) and A^+ for the set $\bigcup_{i=1}^{\infty} A^i$.

Thus A^* relates to A^+ by the relation

$$A^* = A^+ \cup \{\lambda\}$$

A^* is often called the *Kleene closure* or *Kleene star* of A.

A string A is a *prefix* of a string B if there is a string C such that $B = A * C$. It is a *suffix* of B if for some D, we have $B = D * A$. The prefix (suffix) is *proper* if it is not equal to B.

Just as we use *implied multiplication*, ab for $a \times b$ or $a \cdot b$, we also use *implied concatenation*, xy for $x * y$ —leaving it up to the context to fend off ambiguities.

2.7.7 Definition (**Languages**) A *language*, L, over an alphabet A is just a subset of A^*. □

The "interesting" languages are those that are *finitely definable*. *Automata and language theory* studies the properties of such finitely definable languages and of the "machinery" that effects these finite definitions. The language of Logic is also finitely definable.

2.7.8 Definition (**Concatenation of Languages**) If L and M are two languages over an alphabet A, then the symbol $L * M$ or simply (implied concatenation) LM means the set $\{xy : x \in L \wedge y \in M\}$. □


One can learn to live with $*$ as both a unary (one-argument) operation, A^*, and as a binary one, $L * M$, much the same way we can see no ambiguity in uses of minus as $-x$ and $y - z$.


2.8 Exercises

1. An argument towards showing that $\mathbb{U}$, the class of *all* sets and atoms, is a set might go like this:
   Let Σ be a stage *after* all atoms and all member sets of $\mathbb{U}$ were built. At stage Σ we can build $\mathbb{U}$ as a *set*.
   Do you accept the preceding argument? Why?
2. Let a be a set, and consider the class $b = \{x \in a : x \notin x\}$.
   Show that, despite similarities with the Russell class R, b *is* a set.
   Moreover, show that $b \notin a$.
3. Show that R (the Russell class)$= \mathbb{U}$.
4. Show that if a class $\mathbb{A}$ satisfies $\mathbb{A} \subseteq \mathbb{X}$ for all $\mathbb{X}$, then $\mathbb{A} = \emptyset$.
5. Without using set-formation-by-stages Principles, show that $\emptyset \neq \{\emptyset\}$.
6. Without using set-formation-by-stages Principles, show that $\emptyset \notin \emptyset$.
7. Without using set-formation-by-stages Principles, show that $1 \notin 1$.
8. Let us prove that $\{A\}$ —where A is a set— is a set. **Argument**. Well, $\{A\} \subseteq \{A, B\}$ and $\{A, B\}$ has been proved to be a set. We conclude by the subclass theorem. □
   What *exactly* is wrong with this argument?
   Hint. What exactly are we *given*?
9. Now prove that $\{A\}$ is a set *correctly*! Do *not* argue via Principles 0, 1, 2.
10. Prove that the class $\{\{x\} : x = x\}$ which includes only one-element sets $\{x\}$ —but includes *all of them*— is a proper class. Incidentally, the literature calls one-element sets *singletons*.
11. Prove that the class $\{\{x\} : x \text{ is an atom}\}$ is a set.
12. How about the class $\{x : x \text{ is an atom}\}$? Set or proper class?

13. ZF (Zermelo-Fraenkel) axiomatic set theory contains the following axiom, here expressed in terms of sets:

$$\emptyset \neq S \rightarrow (\exists x)\Big(x \in S \wedge \neg(\exists z)(z \in S \wedge z \in x)\Big)$$

Prove that this axiom is true if we accept the Principles of set-formation-by-stages. *Hint*. Pick an $x \in S$ that is *not* built *later* than *any* $z \in S$.

14. Suppose that A and B have intuitively n and m members respectively. That is, $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_m\}$ where all the a_i and b_j are pairwise distinct. Prove that $A \times B$ has nm members.

15. What is $\bigcup \emptyset$ (and why)?

16. What is $\bigcap \emptyset$ (and why)?

17. Show that
(1) $\mathbb{A} \cup \mathbb{B} = \mathbb{B} \cup \mathbb{A}$ and
(2) $\mathbb{A} \cup (\mathbb{B} \cup \mathbb{C}) = \mathbb{A} \cup \mathbb{B} \cup \mathbb{C}$ (ensure that you translate the left hand side correctly).

18. Show that
(1) $\mathbb{A} \cap \mathbb{B} = \mathbb{B} \cap \mathbb{A}$ and
(2) $\mathbb{A} \cap (\mathbb{B} \cap \mathbb{C}) = \mathbb{A} \cap \mathbb{B} \cap \mathbb{C}$ (ensure that you translate the left hand side correctly).

19. Show that $\mathbb{A} \cup (\mathbb{A} \cap \mathbb{B}) = \mathbb{A}$.

20. Show that $\mathbb{A} \cap (\mathbb{A} \cup \mathbb{B}) = \mathbb{A}$.

21. For any set A, show that $\mathbb{U} - A$ is a proper class.

22. Show for any classes $\mathbb{A}, \mathbb{B}$, that $\mathbb{A} - \mathbb{B} = \mathbb{A} - \mathbb{A} \cap \mathbb{B}$.

23. For any classes $\mathbb{A}, \mathbb{B}$ show that $\mathbb{A} \cup \mathbb{B} = \mathbb{A}$ iff $\mathbb{B} \subseteq \mathbb{A}$.

24. For any classes $\mathbb{A}, \mathbb{B}$ show that $\mathbb{A} \cap \mathbb{B} = \mathbb{A}$ iff $\mathbb{A} \subseteq \mathbb{B}$.

25. For any classes $\mathbb{A}, \mathbb{B}$ show that $\mathbb{A} - (\mathbb{A} - \mathbb{B}) = \mathbb{B}$ iff $\mathbb{B} \subseteq \mathbb{A}$.

26. (1) Express $\mathbb{A} \cap \mathbb{B}$ using class difference as the only operation.
(2) Express $\mathbb{A} \cup \mathbb{B}$ using class difference as the only operations.

27. (*Generalized de Morgan's laws*). Prove for any class $\mathbb{A}$ and indexed family $(\mathbb{B}_i)_{i \in \mathbb{F}}$, that

(1) $$\mathbb{A} - \bigcup_{i \in \mathbb{F}} \mathbb{B}_i = \bigcap_{i \in \mathbb{F}} (\mathbb{A} - \mathbb{B}_i)$$

(2) $$\mathbb{A} - \bigcap_{i \in \mathbb{F}} \mathbb{B}_i = \bigcup_{i \in \mathbb{F}} (\mathbb{A} - \mathbb{B}_i)$$

28. (*Distributive laws for* $\cup, \cap$). For any classes $\mathbb{A}, \mathbb{B}, \mathbb{D}$ show

(1) $$\mathbb{A} \cap (\mathbb{B} \cup \mathbb{D}) = (\mathbb{A} \cap \mathbb{B}) \cup (\mathbb{A} \cap \mathbb{D})$$

(2) $$\mathbb{A} \cup (\mathbb{B} \cap \mathbb{D}) = (\mathbb{A} \cup \mathbb{B}) \cap (\mathbb{A} \cup \mathbb{D})$$

29. (*Generalized distributive laws for* $\cup, \cap$). Prove for any class $\mathbb{A}$ and indexed family $(\mathbb{B}_i)_{i\in\mathbb{F}}$, that

(1) $$\mathbb{A} \cap \bigcup_{i\in\mathbb{F}} \mathbb{B}_i = \bigcup_{i\in\mathbb{F}} (\mathbb{A} \cap \mathbb{B}_i)$$

(2) $$\mathbb{A} \cup \bigcap_{i\in\mathbb{F}} \mathbb{B}_i = \bigcap_{i\in\mathbb{F}} (\mathbb{A} \cup \mathbb{B}_i)$$

30. Show that the Principles of set formation by stages disallow the truth of $a \in a$.

31. Show that the axiom of *foundation* (Exercise 2.8.13) disallows the truth of $a \in a$.

32. Show that the Principles of set formation by stages disallow the truth of $a \in b \in c \in \cdots \in a$.

33. Show that the axiom of *foundation* (Exercise 2.8.13) disallows the truth of $a \in b \in c \in \cdots \in a$.

34. Show that $\mathbb{V} = \{x : x \text{ is a set}\}$ is a proper class.

35. Show that for any class (not just set) $\mathbb{A}$, $\mathbb{A} \in \mathbb{A}$ is false.

36. Somebody once said (cf. Wilder 1963) "Consider the class $\mathbb{A}$ of *all abstract ideas*. But that **is** an abstract idea, *so it is a member of itself*."
Discuss.

37. (1) Show that $\mathbb{A}$ ="the class of *all* sets that contain at least one element" can be defined by a defining property.
(2) Show that $\mathbb{A}$ is a proper class.

38. Expand (i.e., show the set in by-listing notation) $2^{\{1,2,3\}}$.

39. This exercise will reappear after we covered Induction over $\mathbb{N}$.
Attach the intuitive meaning to the statement that the set A has n distinct elements.
Show that if A has n elements then $\mathscr{P}(A)$ has 2^n elements.
Hint. Imagine that you arranged the members of A in a straight line in any fixed order you please. So they occupy position 1, position 2, position 3, . . ., position n in an array. Now *any subset* of A can be marked off by a checkmark against each of its members in the above mentioned array. No checkmark against an A-member means it is *not* in the subset under consideration.
Well, we can have *as many subsets as we can have ways* to mark *some* entries of the array and leave the *rest* unmarked! How *many such marking schemes* do we have?

40. Show (*without* the use of the Principles of set formation) that $\big\{\{a\}, \{a, b\}\big\} = \big\{\{a'\}, \{a', b'\}\big\}$ implies $a = a'$ and $b = b'$.

41. For any sets x, y show that $x \cup \{x\} = y \cup \{y\} \to x = y$.
Hint: Use principles of set formation, or even foundation (2.8.13).

42. For any $\mathbb{A}, \mathbb{B}$ show that $\emptyset = \mathbb{A} \times \mathbb{B}$ iff $\mathbb{A} = \emptyset$ or $\mathbb{B} = \emptyset$.

43. (*Distributive law for* $\times$) Show for any $\mathbb{A}, \mathbb{B}$ and $\mathbb{D}$ that

$$\mathbb{D} \times (\mathbb{A} \cup \mathbb{B}) = (\mathbb{D} \times \mathbb{A}) \cup (\mathbb{D} \times \mathbb{B})$$

Relations and Functions

3

Overview

The topic of relations and functions is central in all mathematics and computing. In the former, whether it is calculus, algebra or theory of computation, one deals with relations (notably equivalence relations, order) and all sorts of functions while in the latter one *computes* relations and functions (among other related endeavours[1]), in that, one writes programs that given an input to a relation they compute the response (true or false) or given an input to a function they compute a response which is some object (number, graph, tree, matrix, other) or *nothing*, in case there is no response for said input (for example, there is no response to input "x, y" if what we are computing is $\frac{x}{y}$ but $y = 0$).

We are taking mostly an "extensional" point of view of relations and functions in this course, as is customary in set theory, that is, we view them as sets of (input, output) ordered pairs. It is also possible to take an intentional point of view, especially in theory of computation and some specific areas of mathematics, viewing relations and functions as *methods* to compute outputs from given inputs.

The topics in this chapter include an introduction to equivalence and order relations, finite and infinite sets, to uncountable sets and diagonalisation, and contain the proof of the nontrivial Cantor-Bernstein theorem.

[1] Cf. Tourlakis (2022).

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0_3

3.1 Relations

3.1.1 Definition (Binary relation) A *binary relation* is a class $\mathbb{R}^2$ of ordered pairs.

The statements $(x, y) \in \mathbb{R}$, $x\mathbb{R}y$ and $\mathbb{R}(x, y)$ are equivalent. $x\mathbb{R}y$ is the *infix notation* —imitating notation such as $A \subset B$, $x < y$, $x = y$ and has notational advantages. □

3.1.2 Remark $\mathbb{R}$ contains just pairs (x, y), that is, just sets $\{x, \{x, y\}\}$, in other words, it is a family of sets. □

3.1.3 Example Examples of relations:

(i) $\emptyset$
(ii) $\{(1, 1)\}$
(iii) $\{(1, 1), (1, 2)\}$
(iv) $\mathbb{N}^2$, that is $\{(x, y) : x \in \mathbb{N} \wedge y \in \mathbb{N}\}$. This is a set by the fact that $\mathbb{N}$ is (Why?) and thus so is $\mathbb{N} \times \mathbb{N}$ by 2.7.2.
(v) $<$ on $\mathbb{N}$, that is $\{(x, y) : x < y \wedge x \in \mathbb{N} \wedge y \in \mathbb{N}\}$. This is a set since $< \subseteq \mathbb{N}^2$.
(vi) $\in$, that is,

$$\{(x, y) : x \in y \wedge x \in \mathbb{U} \wedge y \in \mathbb{V}\} \qquad (*)$$

This is a proper class (nonSet). Why? Well, if $\in$ *is* a set, then it is built at some stage Σ. Now examine the arbitrary (x, y) in $\in$. This is $\{x, \{x, y\}\}$ so it is built before Σ, but then so is its member x (available before Σ). Thus we can collect *all* such x into a *set* at stage Σ. But this "set" contains *all* $x \in \mathbb{U}$ due to the middle conjunct in the entrance condition in $(*)$.[3] That is, this "set" is $\mathbb{U}$. This is absurd! □

Here is another way to argue that the relation $\in$ is not a set: If it is, so is $\bigcup \in$. Any $(x, y) \in \in$ is of the form $\{x, \{x, y\}\}$. Thus all x for which there is a y such that $x \in y$ are in $\bigcup \in$. As we said in the footnote, taking $y = \{x\}$ makes clear that "$x \in y$" does not restrict the x's we can get. *We get them all*: thus they form the proper class $\mathbb{U}$. I argued $\mathbb{U} \subseteq \bigcup \in$, thus $\bigcup \in$ cannot be a set. So, neither can $\in$ (2.4.17).

So, a binary relation $\mathbb{R}$ is a table of pairs:

1. Thus one way to view $\mathbb{R}$ is as a device that for inputs x, valued $a, a', \ldots, u, \ldots$ one gets the outputs y, valued $b, b', \ldots, u', \ldots$ respectively. It is all right that a given input may yield multiple outputs (e.g., case (iii) in the previous example).

---

[2] I write "$\mathbb{R}$" or "R" for a relation, generically, but $\mathbb{P}$, $\mathbb{Q}$, $\mathbb{S}$ and $\mathbb{T}$ are available to use as well.

[3] Hmm. Doesn't the first conjunct "$x \in y$" constrain and reduce the number of x-values? No: *For every* x out there take $y = \{x\}$ thus the conjunct $x \in y$ is fulfilled for all x-values, as I just showed how to find a y that works.

2. Another point of view is to see both x and y as inputs and the outputs are true or false (**t** or **f**) according as (x, y) is in the table —that is, $x\mathbb{R}y$ is true— or not. For example, (a, b) is in the table (that is, $a\mathbb{R}b$) hence if the relation receives it as input, then it outputs **t**.

input: x	output: y
a	b
a'	b'
$\vdots$	$\vdots$
u	u'
$\vdots$	$\vdots$

Most of the time we will take the point of view in 1 above. This point of view compels us to define *domain* and *range* of a relation $\mathbb{R}$, that is, the class of all inputs that *cause* an output and the class of all caused outputs respectively.

3.1.4 Definition (Domain and range) For any relation $\mathbb{R}$ we define *domain*, in symbols "dom" by

$$\mathrm{dom}(\mathbb{R}) \stackrel{Def}{=} \{x : (\exists y)x\mathbb{R}y\}$$

where we have introduced the notation "$(\exists y)$" as short for "there exists some y such that", or "for some y,"

Range, in symbols "ran", is defined also in the obvious way:

$$\mathrm{ran}(\mathbb{R}) \stackrel{Def}{=} \{x : (\exists y)y\mathbb{R}x\}$$

Notation 1. For a relation $\mathbb{P}$, the symbol $(a)\mathbb{P}$ means the class of all outputs caused by a:

$$(a)\mathbb{P} \stackrel{Def}{=} \{x : a\mathbb{P}x\}$$

If $(a)\mathbb{P} \neq \emptyset$ and therefore $a \in \mathrm{dom}(\mathbb{P})$ we may also write $(a)\mathbb{P}\downarrow$ and say "$(a)\mathbb{P}$ is defined". Otherwise —$(a)\mathbb{P} = \emptyset$— we write $(a)\mathbb{P}\uparrow$ and say "$(a)\mathbb{P}$ is undefined".

We sometimes want to restrict a relation $\mathbb{S}$ to a class $\mathbb{A}$. There are two main ways to want to do this:

Notation 2. Restrict *both* inputs and outputs to be in $\mathbb{A}$: This is the way we restrict *relations* to obtain a *relational restriction*. We obtain

$$\mathbb{S} \mid \mathbb{A} \stackrel{Def}{=} \mathbb{S} \cap \mathbb{A}^2$$

Notation 3. For functions (to be introduced shortly) one prefers to restrict *only* inputs of $\mathbb{S}$ to be in $\mathbb{A}$: We obtain a *functional restriction*

$$\mathbb{S} \upharpoonright \mathbb{A} \stackrel{Def}{=} \mathbb{S} \cap \Big(\mathbb{A} \times \mathrm{ran}(\mathbb{S})\Big)$$

Notation 4. "Notation 1" above becomes $(a)\Big(\mathbb{S} \mid \mathbb{A}\Big)$ and $(a)\Big(\mathbb{S} \upharpoonright \mathbb{A}\Big)$ in the context of Notations 2 and 3 (note the brackets to help readability). □

We settle the following, before other things:

3.1.5 Theorem *For a* set *relation R, both $dom(R)$ and $ran(R)$ are sets.*

Proof For domain we collect all the x such that $x\,R\,y$, for some y, that is, all the x such that

$$\{x, \{x, y\}\} \in R \tag{1}$$

for some y. Since R is a family of sets, we have that $\bigcup R$ is a set. But then each x in the set $\{x, \{x, y\}\}$ in (1) is in $\bigcup R$. But the set of these x is $\mathrm{dom}(R)$ (3.1.4). Thus $\mathrm{dom}(R) \subseteq \bigcup R$. This settles the domain case.

Now $\bigcup R$ may contain atoms as some of the x in the $\{x, \{x, y\}\}$ maybe indeed be such. Let then A be the set of all atoms in $\bigcup R$ and define

$$S \stackrel{Def}{=} \Big(\bigcup R\Big) - A$$

We know that S is a *set* (cf. 2.4.7).

So, S is a *family of sets* that contains all $\{x, y\}$ as $\bigcup R$ does and no $\{x, y\}$ is an atom. Thus, $\bigcup S$ contains all the y. That is, $\mathrm{ran}(R) \subseteq \bigcup S$, and that settles the range case. □

3.1.6 Definition In practice we often have an *a priori decision* about what are *in principle* "legal" inputs for a relation $\mathbb{R}$, and where its outputs *go*. Thus we have two classes, $\mathbb{A}$ and $\mathbb{B}$ for the class of legal inputs and possible outputs respectively. Clearly we have $\mathbb{R} \subseteq \mathbb{A} \times \mathbb{B}$.

We call $\mathbb{A}$ and $\mathbb{B}$ *left field* and *right field* respectively, and instead of $\mathbb{R} \subseteq \mathbb{A} \times \mathbb{B}$ we often write

$$\mathbb{R} : \mathbb{A} \to \mathbb{B}$$

and also

$$\mathbb{A} \stackrel{\mathbb{R}}{\longrightarrow} \mathbb{B}$$

pronounced "$\mathbb{R}$ is a relation *from* $\mathbb{A}$ *to* $\mathbb{B}$".

The term *field* —without left/right qualifiers— for $\mathbb{R} : \mathbb{A} \to \mathbb{B}$ refers to $\mathbb{A} \cup \mathbb{B}$.

If $\mathbb{A} = \mathbb{B}$ then we have

$$\mathbb{R} : \mathbb{A} \to \mathbb{A}$$

but rather than pronouncing this as "$\mathbb{R}$ is a relation *from* $\mathbb{A}$ *to* $\mathbb{A}$" we *prefer*[4] to say "$\mathbb{R}$ is on $\mathbb{A}$". □

3.1.7 Remark Trivially, for any $\mathbb{R} : \mathbb{A} \to \mathbb{B}$, we have $\text{dom}(\mathbb{R}) \subseteq \mathbb{A}$ and $\text{ran}(\mathbb{R}) \subseteq \mathbb{B}$ (give a quick proof of each of these inclusions).

Also, for any relation $\mathbb{P}$ with no *a priori* specified left/right fields, $\mathbb{P}$ is a relation from $\text{dom}(\mathbb{P}) \to \text{ran}(\mathbb{P})$. Naturally, we say that $\text{dom}(\mathbb{P}) \cup \text{ran}(\mathbb{P})$ is *the* field of $\mathbb{P}$. □

3.1.8 Example As an example, consider the *divisibility relation* on all integers (their set denoted by $\mathbb{Z}$) that is usually named "|":

$$x|y \text{ means } x \text{ divides } y \text{ with 0 remainder}$$

thus, for $x = 0$ and all y, the division is *illegal*, therefore

> *The input* $x = 0$ *to the relation* "|" ***produces no output***, *in other words*, "***for input*** $x = 0$ *the relation is undefined*."

We walk away with two things from this example:

1. It **does** make sense for some relations to *a priori* choose left and right fields, here

$$| : \mathbb{Z} \to \mathbb{Z}$$

   You would not have divisibility on *real numbers*!
2. $\text{dom}(|)$ is the set of all inputs that produce some output. Thus, it is *not* the case *for all relations* that their domain is the same as the left field *chosen*! Note the case in this example! And, incidentally, ignore the term "codomain" that may appear —*erroneously*, instead of the correct "right field"— in some of the elementary discrete mathematics literature! □

3.1.9 Example Next consider the relation $<$ with left/right fields restricted to $\mathbb{N}$. Then $\text{dom}(<) = \mathbb{N}$, but $\text{ran}(<) \subsetneq \mathbb{N}$. Indeed, $0 \in \mathbb{N} - \text{ran}(<)$. □

Let us extract some terminology from the above examples:

[4] Both ways of saying it are correct.

3.1.10 Definition Given

$$\mathbb{R} : \mathbb{A} \to \mathbb{B}$$

If $\text{dom}(\mathbb{R}) = \mathbb{A}$, then we call $\mathbb{R}$ *total* or totally defined. If $\text{dom}(\mathbb{R}) \subsetneqq \mathbb{A}$, then we say that $\mathbb{R}$ is *nontotal*.

If $\text{ran}(\mathbb{R}) = \mathbb{B}$, then we call $\mathbb{R}$ *onto*. If $\text{ran}(\mathbb{R}) \subsetneqq \mathbb{B}$, then we say that $\mathbb{R}$ is *not onto*. □

So, | above is nontotal, and $<$ is not onto.

3.1.11 Example Let $A = \{1, 2\}$.

- The relation $\{(1, 1)\}$ on A is neither total nor onto.
- The relation $\{(1, 1), (1, 2)\}$ on A is onto but not total.
- The relation $\{(1, 1), (2, 1)\}$ on A is total but not onto.
- The relation $\{(1, 1), (2, 2)\}$ on A is total *and* onto. □

3.1.12 Definition The relation Δ_A on the set A is given by

$$\Delta_A \stackrel{Def}{=} \{(x, x) : x \in A\}$$

We call it the *diagonal* ("Δ" for "diagonal") or *identity* relation on A.

Consistent with the second terminology, we may also use the symbol $\mathbf{1}_A$ for this relation. □

3.1.13 Definition A relation R (not *a priori* restricted to have *predetermined* left or right fields) is

1. *Transitive*: Iff $x\,R\,y \wedge y\,R\,z$ implies $x\,R\,z$.
2. *Symmetric*: Iff $x\,R\,y$ implies $y\,R\,x$.
3. *Antisymmetric*: Iff $x\,R\,y \wedge y\,R\,x$ implies $x = y$.
4. *Irreflexive*: Iff $x\,R\,y$ implies $x \neq y$.

Now assume R is *on a set* A. Then we call it reflexive iff $\Delta_A \subseteq R$. □

3.1.14 Example

(i) *Transitive* examples: $\emptyset$, $\{(1, 1)\}$, $\{(1, 2), (2, 3), (1, 3)\}$, $<$, $\leq$, $=$, $\mathbb{N}^2$.
(ii) *Symmetric* examples: $\emptyset$, $\{(1, 1)\}$, $\{(1, 2), (2, 1)\}$, $=$, $\mathbb{N}^2$.
(iii) *Antisymmetric* examples: $\emptyset$, $\{(1, 1)\}$, $=$, $\leq$, $\subseteq$.
(iv) *Irreflexive* examples: $\emptyset$, $\{(1, 2)\}$, $<$, $\subsetneqq$, the relation "$\neq$" on $\mathbb{N}$.

(v) *Reflexive* examples: $\mathbf{1}_A$ on A, $\{(1,1)\}$ on $\{1\}$, $\{(1,2),(2,1),(1,1),(2,2)\}$ on $\{1,2\}$, $=$ on $\mathbb{N}$, $\leq$ on $\mathbb{N}$. □

3.1.15 Exercise Show that R is symmetric iff $xRy \equiv yRx$. □

We can compose relations:

3.1.16 Definition (Relational Composition) Let R and S be (set) relations. Then, their composition, *in that order*, denoted by $R \circ S$ is defined for all x and y by:

$$xR \circ Sy \stackrel{Def}{\equiv} (\exists z)\Big(xRz \wedge zSy\Big)$$

It is customary to abuse notation and write "$xRzSy$" for "$xRz \wedge zSy$" just as one writes $x < y < z$ for $x < y \wedge y < z$.

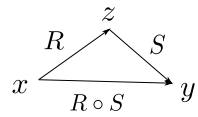
The definition, *unchanged*, applies to any *class* relations $\mathbb{R}$ and $\mathbb{S}$ as well. □

3.1.17 Example Here is whence the emphasis "*in that order*" above. Say, $R = \{(1,2)\}$ and $S = \{(2,1)\}$. Thus, $R \circ S = \{(1,1)\}$ while $S \circ R = \{(2,2)\}$. Thus, $R \circ S \neq S \circ R$ *in general*. □

3.1.18 Example For any R, we diagrammatically indicate xRy by

$$x \xrightarrow{R} y$$

Thus, the situation where we have that $xR \circ Sy$ means, for some z, $xRzSy$ and is depicted as:



□ 

3.1.19 Theorem *The composition of two (set) relations R and S in that order is also a set.*

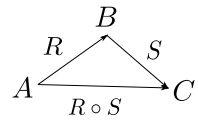
Proof Trivially, $R \circ S \subseteq \mathrm{dom}(R) \times \mathrm{ran}(S)$.

Indeed, if $xR \circ Sy$ then $xRzSy$, for some z. Hence the x is in $\mathrm{dom}(R)$ (by xRz) and the y is in $\mathrm{ran}(S)$ (by zSy). Moreover, we proved in 3.1.5 that $\mathrm{dom}(R)$ and $\mathrm{ran}(S)$ are sets. Thus so is $\mathrm{dom}(R) \times \mathrm{ran}(S)$ (2.7.2). □

3.1.20 Corollary *If we have $R : A \to B$ and $S : B \to C$, then $R \circ S : A \to C$.*

Proof This is a trivial modification of the argument above. □

The result of the corollary is depicted diagrammatically as



3.1.21 Theorem (Associativity of composition) *For any relations $\mathbb{R}$, $\mathbb{S}$ and $\mathbb{T}$, we have*

$$(\mathbb{R} \circ \mathbb{S}) \circ \mathbb{T} = \mathbb{R} \circ (\mathbb{S} \circ \mathbb{T})$$

We state and prove this central result for any class relations.

Proof We have two directions:

$\to$: Fix x and y and let $x(\mathbb{R} \circ \mathbb{S}) \circ \mathbb{T} y$.

Then, for some z, we have $x(\mathbb{R} \circ \mathbb{S}) z \mathbb{T} y$ and hence for some w, the above becomes

$$x \mathbb{R} w \mathbb{S} z \mathbb{T} y \tag{1}$$

But $w \mathbb{S} z \mathbb{T} y$ means $w \mathbb{S} \circ \mathbb{T} y$, hence we rewrite (1) as

$$x \mathbb{R} w (\mathbb{S} \circ \mathbb{T}) y$$

Finally, the above says $x \mathbb{R} \circ (\mathbb{S} \circ \mathbb{T}) y$.

$\leftarrow$: Fix x and y and let $x \mathbb{R} \circ (\mathbb{S} \circ \mathbb{T}) y$.

Then, for some z, we have $x \mathbb{R} z (\mathbb{S} \circ \mathbb{T}) y$ and hence for some u, the above becomes

$$x \mathbb{R} z \mathbb{S} u \mathbb{T} y \tag{2}$$

But $x \mathbb{R} z \mathbb{S} u$ means $x \mathbb{R} \circ \mathbb{S} u$, hence we rewrite (2) as

$$x (\mathbb{R} \circ \mathbb{S}) u \mathbb{T} y$$

Finally, the above says $x(\mathbb{R} \circ \mathbb{S}) \circ \mathbb{T} y$. □

The following is almost unnecessary, but offered for emphasis:

3.1.22 Corollary *If R, S and T are (set) relations, all on some set A,*[5] *then "$R \circ S \circ T$" has a meaning that is* independent of how brackets are inserted.

[5] Recall that "R is on a set A" means $R \subseteq A^2$, which is the same as $R : A \to A$.

The corollary allows us to just omit brackets in a chain of compositions, even longer than the above. It also leads to the definition of relational exponentiation, below:

3.1.23 Definition (Powers of a binary relation) Let R be a (set) relation. We define R^n, for $n > 0$, as

$$\underbrace{R \circ R \circ \cdots \circ R}_{n\ R} \tag{1}$$

Note that the resulting relation in (1) is independent of how brackets are inserted (3.1.22).

If moreover we have defined R to be on a set A, then we also define the 0-th power: R^0 stands for Δ_A or $\mathbf{1}_A$. □

3.1.24 Exercise Let R be a relation on A. Then for all $n \geq 0$, R^n is a set.

Hint. You do not need to do induction. A "and so on" argument will be all right. □

3.1.25 Example Let $R = \{(1, 2), (2, 3)\}$. What is R^2?

Well, when can we have xR^2y? Precisely if/when we can find x, y, z that satisfy $xRzRy$. The values $x = 1$, $y = 3$ and $z = 2$ are the *only ones* that satisfy $xRzRy$.

Thus $1R^23$, or $(1, 3) \in R^2$. We conclude $R^2 = \{(1, 3)\}$ due to the "only ones" above. □

3.1.26 Exercise Show that if for a a relation R we know that $R^2 \subseteq R$, then R is transitive and conversely. □

3.2 Transitive Closure

3.2.1 Definition (Transitive closure of R) *A transitive closure* of a relation R —if it exists— is a $\subseteq$-*smallest* transitive T that contains R as a subset.

More precisely,

1. T is transitive, and $R \subseteq T$.
2. If S is also transitive and $R \subseteq S$, then $T \subseteq S$. This makes the term "$\subseteq$-*smallest*" precise.

□

Note that we hedged twice in the definition, because at this point we do not know yet:

- If every relation has a transitive closure; hence the "if it exists".
- We do not know if it is unique, hence the emphasised indefinite article "*A*".

3.2.2 Remark Uniqueness can be settled immediately *from the definition above*: Suppose T and T' fulfil Definition 3.2.1, that is,

1. $R \subseteq T$
   and
2. $R \subseteq T'$

since both are closures. But now think of T as a closure and T' as the "S" of 3.2.1 (it includes R all right!)

Hence $T \subseteq T'$.

Now reverse the role playing and think of T' as a closure, while T plays the role of "S". We get $T' \subseteq T$. Hence, $T = T'$. □

3.2.3 Definition The unique transitive closure, *if it exists*, is denoted by R^+. □

3.2.4 Exercise If R is transitive, then R^+ exists. In fact, $R^+ = R$. □

The above exercise is hardly exciting, but learning that R^+ exists for *every* R and also learning how to "compute" R^+ *is* exciting. We do this next.

3.2.5 Lemma *Given a (set) relation R. Then $\bigcup_{n=1}^{\infty} R^n$ is a transitive (set) relation.*

Proof We have two things to do.

1. $\bigcup_{n=1}^{\infty} R^n$ is a set.
2. $\bigcup_{n=1}^{\infty} R^n$ is a transitive relation.

Proof of **1.** Note that all positive powers of R, R^{n+1}, for $n \geq 0$, are sets. Indeed, they *all are subsets of the same set*!
Here is why:
Firstly, $R \subseteq \mathrm{dom}(R) \times \mathrm{ran}(R)$ by Definition 3.1.4.
Let now $n > 0$: We have

$$R^{n+1} = \overbrace{R \circ R \circ \ldots \circ R}^{n+1} = \overbrace{R \circ R \circ \ldots \circ R}^{n} \circ R = R^n \circ R$$

similarly, observing that

$$\overbrace{R \circ R \circ \ldots \circ R}^{n+1} = R \circ \overbrace{R \circ \ldots \circ R}^{n} = R \circ R^n$$

we have $R^{n+1} = R \circ R^n$. Thus, we established

$$R^{n+1} = R \circ R^n \tag{1}$$

and

$$R^{n+1} = R^n \circ R \tag{2}$$

Applying 3.1.19 to (1) we get

$$R^{n+1} \subseteq \mathrm{dom}(R) \times \ldots \tag{1'}$$

and applying 3.1.19 to (2) we get

$$R^{n+1} \subseteq \ldots \times \mathrm{ran}(R) \tag{2'}$$

Thus

$$R^{n+1} \subseteq \mathrm{dom}(R) \times \mathrm{ran}(R)$$

for $n \geq 0$.
Therefore,

$$X \in \mathbb{F} = \{R^i : i = 1, 2, 3, \ldots\} \Rightarrow X \subseteq \mathrm{dom}(R) \times \mathrm{ran}(R) \Rightarrow X \in 2^{\mathrm{dom}(R) \times \mathrm{ran}(R)} \tag{3}$$

Thus $\mathbb{F}$ —being a subclass of $2^{\mathrm{dom}(R) \times \mathrm{ran}(R)}$— is a set and so is

$$\bigcup \mathbb{F} \overset{2.4.21}{=} \bigcup_{i=1}^{\infty} R^i$$

Proof of **2.** Next, let

$$x \bigcup_{i=1}^{\infty} R^i \; y \bigcup_{i=1}^{\infty} R^i \; z$$

Thus for some n and m we have

$$x \; R^n \; y \; R^m \; z$$

this says the same thing as

$$x \; \overbrace{R \circ R \circ \cdots R}^{n} \; y \; \overbrace{R \circ R \circ \cdots R}^{m} \; z$$

or

$$x \; \overbrace{R \circ R \circ \cdots R}^{n} \circ \overbrace{R \circ R \circ \cdots R}^{m} \; z$$

or

$$x \; \overbrace{R \circ R \circ \cdots R}^{n+m} \; z$$

that is,

$$x \bigcup_{i=1}^{\infty} R^i z \qquad \square$$

3.2.6 Remark Why all this work for Part 1 of the proof above? Why not just *use* 2.4.21 right away? Because 2.4.21 offers *only notation* once we *know* that

$$\mathbb{F} = \{A_0, A_1, A_2, A_3, \ldots\} \tag{3}$$

is a set! Cf. "Suppose the family of sets Q is a set of sets", the opening statement in the passage 2.4.21 on *notation* states.

Here we do *not know* (yet) if every family of sets like (3) is indeed a set —but in *this* case it turns out that we *do not care* because *every* member of $\mathbb{F} = \{R^i : i = 1, 2, 3, \ldots\}$ is included (as a subset) in $\text{dom}(R) \times \text{ran}(R)$ (a set), which allows us to sidestep the issue!

Whether *every* family of *sets* like $\mathbb{F}$ in (3) is a set will be answered affirmatively in 3.3.6. For now note that we cannot recklessly say that after *any* sequence of construction by stages steps there is a stage after all those stages. Why? Well, take *all* the objects in set theory. Each is given outright (atom; stage 0) or is constructed at some stage (set). If we could *prove* there is a stage after all these stages then we could also *prove* that $\mathbb{U}$ is a set, a claim we *refuted* with two methods so far! $\square$

Since $R \subseteq \bigcup_{i=1}^{\infty} R^i$ due to $R = R^1$, all that remains to show is that $\bigcup_{i=1}^{\infty} R^i$ is a transitive closure of R is to show that

3.2.7 Lemma *If $R \subseteq S$ and S is transitive, then $\bigcup_{i=1}^{\infty} R^i \subseteq S$.*

Proof I will just show that for all $n \geq 1$, $R^n \subseteq S$. OK, $R \subseteq S$ is our assumption, thus $R^1 \subseteq S$ is true.

For $R^2 \subseteq S$ let xR^2y, thus (for some z), $xRzRy$ hence $xSzSy$. As S is transitive, the latter gives xSy. Done.

For $R^3 \subseteq S$ let xR^3y, thus (for some z), xR^2zRy hence $xSzSy$. As S is transitive, the latter gives xSy. Done.

You see the pattern: *Assume* now that we proved up to *some fixed but unspecified* n that (1) below holds and we want to prove for $n + 1$ that $R^{n+1} \subseteq S$ as well using the *same value* for n, as in our *assumption* above.

$$\text{So, we have } R^n \subseteq S. \tag{1}$$

Thus,

$$x R^{n+1} y \iff x R^n \circ Ry \iff x R^n z R y \text{ (some } z) \overset{(1)}{\Longrightarrow} x S z S y \Longrightarrow x S y \text{ (S transitive)}$$

□

We have proved:

3.2.8 Theorem (The transitive closure exists) *For any relation R, its transitive closure R^+ exists and is unique. Indeed we have that $R^+ = \bigcup_{i=1}^{\infty} R^i$.*

An interesting corollary that will lend a computational flavour to 3.2.8 is the following.

3.2.9 Corollary *If R is on the set $S = \{a_1, a_2, \ldots, a_n\}$ where for all i, j in S we have $i \neq j$ implies $a_i \neq a_j$, then $R^+ = \bigcup_{i=1}^{n} R^i$.*

Proof By 3.2.8, all we have to do is prove

$$\bigcup_{i=1}^{\infty} R^i \subseteq \bigcup_{i=1}^{n} R^i \tag{1}$$

since the $\supseteq$ part is obvious.

So let $x \bigcup_{i=1}^{\infty} R^i y$. This means that

$$x R^q y, \text{ for some } q \geq 1 \tag{2}$$

Thus, I have two cases for (2):

Case 1. $q \leq n$. Then $x \bigcup_{i=1}^{n} R^i y$ since $R^q \subseteq \bigcup_{i=1}^{n} R^i$, R^q being one of the "R^i" with i in the $1 \leq i \leq n$ range.

Case 2. $q > n$. In this case I will show that there is also a $k \leq n$ such that $x R^k y$, which sends me back to the "easy **Case 1**".
Well, if there is *one* $q > n$ that satisfies (2) there are probably more. Let us choose our q to be *the smallest* $> n$ that gives us (2).

Wait! Why is there a *smallest* q such that

$$x R^q y \text{ and } q > n \,? \tag{3}$$

Because among those "q" that fit (3)[6] imagine we fix attention to *one* such.
Now, if it is not the smallest such, then go down to the *next smaller* one that still satisfies (3), call it q'.
Now go down to the next smaller, $q'' > n$, if q' is not smallest.

[6] There is at least one, else we would *not* be in **Case 2**.

Continue like this. Can I do this forever? That is, can we have the following being an infinitely long sequence of distinct numbers?

$$n < \ldots < q^{(k)}[7] < \ldots < q''' < \ldots < q'' < q' < q \tag{4}$$

If yes, then I will have an infinite "descending" chain of distinct natural numbers between q and n. *Absurd!*[8]

Back to the proof. So let the q we are working with be the smallest that satisfies (3). Then we have the configuration

$$x\,R z_1 R z_2 R z_3 \ldots z_{i-1} R \boxed{z_i R z_{i+1} \ldots} z_r R z_{r+1} \ldots z_{q-1} R y \tag{5}$$

The above accounts for q copies of R as needed for

$$R^q = \overbrace{R \circ \ldots R}^{q\ R}$$

Now the sequence

$$z_1, z_2, z_3 \ldots z_i, z_{i+1}, \ldots z_r, z_{r+1}, \ldots, z_{q-1}, y$$

in (5) above contains $q > n$ members. As they all come from A, *not all are distinct*. So let $z_i = z_r$ (the z_r could be as late in the sequence as y, i.e., equal to y).
Now omit the boxed part in (5). We obtain

$$\begin{array}{c} x\,R z_1 R z_2 R z_3 \ldots z_{i-1} R z_r R z_{r+1} \ldots z_{q-1} R y \\ \| \\ z_i \end{array} \tag{6}$$

which contains *at least* one *"R" fewer* than the sequence (5) does —the entry "$z_i R z_{i+1}$" (and everything else in the "…" part in the box) being removed. That is, (6) states

$$x\,R^{q'}\,y$$

with $q' < q$. Since the q in (3) was *smallest* $> n$, we *must have* $q' \leq n$ which sends us to **Case 1** and we are done. □

[7] By "$q^{(k)}$" I mean q with k primes.

[8] Including n and q there are exactly $q - n + 1$ distinct numbers in (4).

3.2.1 Computing the Transitive Closure

The result from 3.2.9 permits the computation of the transitive closure of relations on finite sets. We will give a general definition of "finite" later but for this subsection we mean sets like S in 3.2.9. We will introduce a matrix representation of relations R on finite sets (in the sense agreed to for this subsection) the matrix rows and columns being *indexed* by the entries of the set S. Since matrices like their coordinates to be pairs of natural numbers, much will be gained in clarity and nothing lost in mathematical generality if the names of the entries of finite sets like S of 3.2.9 are *natural numbers* rather than a generic letter "a" *indexed* by natural numbers. Our S-sets therefore are precisely the

$$S = \{1, 2, 3, \ldots, n\} \tag{1}$$

A relation R on a set S (as in (1)) can be represented by a matrix.

3.2.10 Definition (Matrices and the Adjacency matrix) A *matrix* is the term used in mathematics for the programming term "two dimensional array". An "$m \times k$" matrix has m rows and k columns. The address or location of an item in a matrix M, just as in programming, is given by two *coordinates*, i (row number) and j (column number) with this notation $M(i, j)$. Two matrices M and N are equal iff

- They are both $k \times r$, for some k and r,
  and
- for all i, j satisfying $1 \leq i \leq k$ and $1 \leq j \leq r$ we have $M(i, j) = N(i, j)$.

A special case of matrices are the so-called adjacency matrices. Given a relation R on a finite set S. Its *adjacency matrix* M_R —or just M if R is understood— is an $n \times n$ matrix of 0-1 entries given by

$$M_R(i, j) \stackrel{Def}{=} \begin{cases} 1 & \text{if } iRj \\ 0 & \text{othw} \end{cases}$$

For computational purposes the entries "1" and "0" are taken to be "Boolean" with respect to addition, that is, their arithmetic is not the normal one on natural numbers but is governed by the "addition table" below.

Table 3.1 Addition table

x	y	x + y
0	0	0
0	1	1
1	0	1
1	1	1

Table 3.2 Multiplication table

x	y	x × y
0	0	0
0	1	0
1	0	0
1	1	1

Multiplication is the same one as for numbers:

□

3.2.11 Example (Matrix addition) Addition of two $n \times n$ matrices is done by adding all the entries with the same coordinates in the two matrices. That is, if M and N are two $n \times n$ matrices, then $(M + N)(i, j) \stackrel{Def}{=} M(i, j) + N(i, j)$, for all i, j.

Thus

- $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$
- $\begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

□

3.2.12 Example (Matrix multiplication) Multiplication of two $n \times n$ matrices is done according to the following formula:

Say M and N are two $n \times n$ matrices. Then

$$(M \times N)(i, j) \stackrel{Def}{=} \sum_{k=1}^{n} M(i, k) \times N(k, j), \text{ for all } i, j$$

 The notation $\sum_{k=1}^{n} f(k)$ means take all $f(k)$, for $1 \leq k \leq n$, and add them. 

Thus $\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ because

1. Address (1, 1) of the product holds $0 \times 0 + 1 \times 1 = 1$
2. (1, 2) holds $0 \times 1 + 1 \times 1 = 1$
3. (2, 1) holds $1 \times 0 + 1 \times 1 = 1$
4. (2, 2) holds $1 \times 1 + 1 \times 1 = 1 + 1 = 1$

□

Incidentally, an $n \times n$ matrix is called a "square" matrix.

3.2.13 Example (The Identity Matrix) For each $n > 0$ we have an $n \times n$ *identity matrix* I_n —or I, if the n is understood from the context— whose entries are as follows:

1. The *diagonal entries* are all equal to 1: That is, $I_n(i, i) = 1$, for all $1 \leq i \leq n$.
2. All non diagonal entries are zero: That is, $I_n(i, j) = 0$, for all $1 \leq j, i \leq n$ such that $i \neq j$.

□

3.2.14 Example Given two relations R and Q on $A = \{1, 2, \ldots, n-1, n\}$ for some n.

We can calculate the entries of $M_{R\circ Q}$ in terms of the entries of M_R and M_Q that we have *outright*, after all, that *is* how R and Q are "*given*" to a computer: via M_R and M_Q.

Indeed, pick i and j. By definition of "$\circ$",

$$i\,R \circ Q\,j \text{ iff, for some } k, \text{ it is } i\,R\,k \text{ and } k\,Q\,j \tag{1}$$

By 3.2.10, (1) is equivalent to

$$i\,R \circ Q\,j \text{ iff, for some } k, \text{ it is } M_R(i,k) \times M_Q(k,j) = 1 \tag{1'}$$

The above in turn can be written without the "wordy" part "for some k" as follows (see also Tables 3.1 and 3.2)

$$i\,R \circ Q\,j \text{ iff } \left(\sum_{k=1}^{n} M_R(i,k) \times M_Q(k,j)\right) = 1 \tag{1''}$$

Indeed, the part $\sum_{k=1}^{n} M_R(i,k) \times M_Q(k,j)$ is 1 iff, *for at least one k-value*, ("for some k" as we also say) we have $M_R(i,k) \times M_Q(k,j) = 1$.

One last observation and we are done:
By 3.2.12 we have $(M_R \times M_Q)(i,j) = \sum_{k=1}^{n} M_R(i,k) \times M_Q(k,j)$. Factoring in (1'') we have now

$$i\,R \circ Q\,j \text{ iff } (M_R \times M_Q)(i,j) = 1 \tag{1'''}$$

In other words,

$$M_{R\circ Q}(i,j) = (M_R \times M_Q)(i,j), \text{ for all } i, j \tag{2}$$

and thus

$$M_{R\circ Q} = M_R \times M_Q \tag{3}$$

Pause. Elaborate the "*In other words*".◀ □

3.2.15 Remark In particular, if R is on $A = \{1, 2, \ldots n\}$, then $M_{R^2} = M_R \times M_R$ or as we write usually,

1.
$$M_{R^2} = M_R^2 \tag{4}$$

2.
$$M_{R^3} = M_{R^2 \circ R} \overset{(3)}{=} M_{R^2} \times M_R \overset{(4)}{=} M_R^2 \times M_R = M_R^3$$

3. Suppose now that we have had enough perseverance to progress sufficiently far to a number m that we will *not* disclose, and obtained
$$M_{R^m} = M_R^m \tag{5}$$

4. To show that we can obtain identities like (5) as far as we like we show how to stand on the shoulders of (5) and obtain (6) below for $m + 1$ ("m" *is still the fixed undisclosed number we used in* (5))
$$M_{R^{m+1}} = M_{R^m \circ R} \overset{(3)}{=} M_{R^m} \times M_R \overset{(5)}{=} M_R^m \times M_R = M_R^{m+1} \tag{6}$$

□

3.2.16 Exercise Give an example of two 2×2 matrices M and N such that $M \times N \neq N \times M$. □

3.2.17 Exercise Prove that for any $n \times n$ adjacency matrix M we have

$$M \times I_n = I_n \times M = M$$

□

We have just seen that a computation of R^+ *can be based on (Boolean) matrix multiplication* (due to 3.2.9 and the results immediately above). We want to compute

$$R \cup R^2 \cup R^3 \cup \cdots \cup R^n$$

as

$$M_R + M_R^2 + M_R^3 + \cdots + M_R^n$$

Here is then the most obvious algorithm to do so:

3.2.18 Example (A crude algorithm for computing R^+) Let R be on $A = \{1, 2, \ldots, n\}$. We can compute M_{R^+} and hence R^+ as follows:

$$
\begin{array}{lll}
T & \leftarrow & I_n \\
\text{for } k = 1 \text{ to } n & & \text{do} \\
T & \leftarrow M_R + T \times M_R & \\
 & & \text{end}
\end{array}
$$

In fact, on the k-th iteration of the loop ($1 \leq k \leq n$) T holds

$$M_R + M_R^2 + \cdots + M_R^k$$

since the successive contents of T at the end of the k-the iteration are

$k = 1$, $T = M_R + I_n \times M_R = M_R + M_R \overset{Why?}{=} M_R$

$k = 2$, $T = M_R + T \times M_R = M_R + M_R^2$

$k = 3$, $T = M_R + T \times M_R = M_R + (M_R + M_R^2) \times M_R = M_R + M_R^2 + M_R^3$

Guess and Postulate the pattern for $k = m$ below:

$k = m$, $T = M_R + M_R^2 + \cdots + M_R^m$, thus (we are right! The pattern is preserved below!)

$k = m + 1$,

$$
\begin{aligned}
T =& M_R + (M_R + M_R^2 + \cdots + M_R^m) \times M_R \\
=& M_R + M_R^2 + M_R^3 + \cdots + M_R^{m+1}
\end{aligned}
$$

thus we validated the form of T also for iteration $k = m + 1$ and therefore our "guess" of the form of T at iteration $k = m$ is correct for all m-values. In particular,

It follows that when the program exits ($k = n$ iterations) we have

$$T = M_R + M_R^2 + M_R^3 + \cdots + M_R^n$$

and therfore

$$T = M_{R^+}$$

If we ignore a multiplicative constant, the algorithm's run time is n^3 Boolean additions and multiplications every time it goes through a loop iteration. That is, it is Kn^4 such operations overall (over all n passages through the loop), for all large n where K is a constant that in the *analysis of algorithms* domain we "normally" (read "most of the time") do not care to specify exactly.

This "do not care" attitude has led to the so-called "big-O" notation that we will develop in some detail in Section 7.1. Put simply, for now, for two non negative expressions $f(n)$

and $g(n)$, we can express the English "the expression $f(n)$ is bounded (or *majorised*) by a constant times $g(n)$, for all large n" —or also the less verbose "$f(n) \leq K \times g(n)$, for all $n \geq N_0$"— by the *very brief notation*

$$f(n) = O(g(n))$$

Thus, overall the run time is dominated by n times n^3 (the program loops n times), that is, the algorithm's run time is $O(n^4)$ (Boolean operations) in big-O notation. □

3.2.2 The Special Cases of Reflexive Relations on Finite Sets

We can compute R^+ on a finite set A faster if we know that R is reflexive on A. This better algorithm is based on the following theorem and its corollaries.

3.2.19 Theorem *If R is reflexive on $A = \{1, 2, 3, \ldots, n\}$ and $m \geq n$, then*

$$\bigcup_{i=1}^{m} R^i = R^{m-1} \tag{1}$$

Proof Our proof relies on the techniques in 3.2.9 (ibid. Case 2 of the proof). Towards (1) we have two directions.

$\supseteq$-direction. We want $\bigcup_{i=1}^{m} R^i \supseteq R^{m-1}$, but this is trivial.

$\subseteq$-direction. We want $\bigcup_{i=1}^{m} R^i \subseteq R^{m-1}$. This needs some work. So let $x \bigcup_{i=1}^{m} R^i y$.

Then $x R^i y$, for *some* i among $1 \leq i \leq m$.

We have three cases for the above:

1. $i = m - 1$ works for the assumption (left hand side of $\subseteq$). But then $x R^{m-1} y$ which is our conclusion.
2. $i = k < m - 1$ works for the assumption.

   From $x R^k y$ and $y R y$ (reflexivity) we get $x R^k y \overbrace{R y \ldots y R y}^{m-1-k \ R}$ which trivially implies $x R^{m-1} y$.
3. $i = m$ works for the assumption. We must reduce i so we end up in one of the above two cases. We have

$$\overbrace{x R a_1 R a_2 R a_3 \cdots a_{m-1} R y}^{m \ R} \tag{2}$$

We will partly use the technique of proof of 3.2.9, Case 2. Now we named $m+1$ points in (2) but we have only $n < m+1$ distinct ones in A. So two names in (2) must name the same point. We have cases:

Case 1. $a_s = a_t$ for some $1 \leq s, t \leq m-1$. Argue as in (proof of) 3.2.9, Case 2 to remove at least one R from (2) thus end up with $x R^q y$, where $q \leq m-1$.

Case 2. $x = a_r$ for some $1 \leq r \leq m-1$. Argue as in (proof of) 3.2.9, Case 2 to remove at least one R from (2) thus end up with $x R^q y$, where $q \leq m-1$.

Case 3. $y = a_r$ for some $1 \leq r \leq m-1$. Argue as in (proof of) 3.2.9, Case 2 to remove at least one R from (2) thus end up with $x R^q y$, where $q \leq m-1$.

Case 4. $x = y$. By reflexivity we have $x R y$, that is, $x R^1 y$. But $1 \leq m-1$. □

3.2.20 Corollary *If R is reflexive on $A = \{1, 2, \ldots, n\}$ and $m \geq n$, then $R^+ = R^{m-1}$.*

Proof We know that

$$R^+ \overset{3.2.9}{=} \bigcup_{i=1}^{n} R^i \overset{3.2.8}{=} \bigcup_{i=1}^{\infty} R^i$$

Thus adding powers R^i (via the union operation "$\cup$") *beyond the n-th* does not alter the expression $\bigcup_{i=1}^{n} R^i$. So, if $m \geq n$, then

$$R^+ = \bigcup_{i=1}^{n} R^i = \bigcup_{i=1}^{m} R^i \overset{3.2.19}{=} R^{m-1}$$

□

We can now compute faster: Let R be reflexive on $A = \{1, 2, \ldots, n\}$ and let p be *smallest*[9] such that

$$n - 1 \leq 2^p \tag{3}$$

That is $p = \lceil \log_2(n-1) \rceil$.[10] Set $m = 2^p + 1$. Thus $n \leq m$ and by 3.2.20

$$R^+ = R^{m-1} = R^{2^p} \tag{4}$$

We can now compute with jumps, by starting with M_R and using repeated *squaring* (p times)

[9] As the expression 2^p increases without bound and $n-1$ is fixed, there are infinitely many p for which $n - 1 \leq 2^p$ is true. We can thus pick the smallest p that works.

[10] Let x be a real number and t an integer such that $t - 1 < x \leq t$. Then we call t the *ceiling* of x and write $t = \lceil x \rceil$.

$$
\begin{array}{ll}
T & \leftarrow M_R \\
\text{for } k = 1 \text{ to } p \text{ do} & \\
T & \leftarrow T^2 \\
& \text{end}
\end{array}
$$

By (4) T, at the end of the algorithm above, holds the adjacency matrix $M_R^{2^p}$ of R^+.

The computation of T takes[11] a *constant* times $p \cdot n^3 = n^3 \cdot \lceil \log_2(n-1) \rceil$ *Boolean* $+, \times$ operations to conclude the above indicated computation. In big-O notation that is $O(n^3 \cdot \lceil \log_2(n-1) \rceil)$ Boolean operations.

3.2.3 Warshall's Algorithm

There is an even faster way to compute R^+ due to Warshall. The algorithm relies on the visualisation that $x R^+ y$ means that there is a path from x to y that passes through points (members) of A which are connected by arrows labelled "R" and all (arrows) point in the direction from the "start" x toward the "end" y.

$$x \xrightarrow{R} a_1 \xrightarrow{R} a_2 \xrightarrow{R} a_3 \xrightarrow{R} \cdots \xrightarrow{R} a_j \xrightarrow{R} a_{j+1} \xrightarrow{R} \cdots \xrightarrow{R} a_{r-1} \xrightarrow{R} y \qquad (1)$$

Thus $M_{R^+}(x, y) = 1$ iff x and y are connected by a path as depicted in (1) above, that is, $x R^r y$ holds. What the algorithm below does is whenever it detects (1) and (1′) below — manifested as $T(x, y) = 1$ and $T(y, z) = 1$— it adds an "edge" from x to z, that is, makes $T(x, z) = 1$ too.

$$y \xrightarrow{R} b_1 \xrightarrow{R} b_2 \xrightarrow{R} b_3 \xrightarrow{R} \cdots \xrightarrow{R} b_j \xrightarrow{R} b_{j+1} \xrightarrow{R} \cdots \xrightarrow{R} b_{q-1} \xrightarrow{R} z \qquad (1')$$

The algorithm is simple:

$$
\begin{array}{lllll}
T & \leftarrow & M_R & & \\
\text{for } j = 1 & \text{to} & n & & \\
\text{for } i = 1 & \text{to} & n & & \\
\text{for} & k = 1 & \text{to} & n & \\
& T(i,k) & \leftarrow T(i,k) + T(i,j) \times T(j,k) & &
\end{array}
$$

The command in the last loop says "if there is a path from i to j and one from j to k then acknowledge a path (edge) from i to k by making $T(i, k)$ equal to 1."

This is the correct behaviour for the algorithm but the \$1M question is:
Does the algorithm add *all* the edges (paths) needed for R^+?

[11] "It takes" in the analysis of an algorithm's run time is *rarely* exact. Usually, as is the case here, "it takes" is short for "it takes up to"; an upper bound on steps/time.

Is it *possible* that $T(i,k)$, for *some* i,k, will *incorrectly stay* 0 because, when we come to perform $T(i,j) \times T(j,k)$, the entry $T(j,k)$ is not 1 *yet*?

No, not possible.

We will prove the correctness of Warshall's algorithm employing a "trick". Well, not a trick really, but the *methodology* of "dynamic programming" taught in courses on algorithms but also appearing in the proof of *Kleene's theorem* that expresses sets recognised by *finite automata* (FA) as *regular expressions* (cf. for example, Tourlakis (2012)). Namely, we add notation to help the *reasoning* about the correctness of the program above.

We add a superscript to T on the right and left of "$\leftarrow$" in the innermost loop:

$$T^{(j)}(i,k) \leftarrow T^{(j-1)}(i,k) + T^{(j-1)}(i,j) \times T^{(j-1)}(j,k)$$

The meaning of $T^{(q)}(x,y)$ is that this entry is 1 precisely if there is a "path" from x to y (such as (1)) that does not use *intermediate* points a_r that are *outside* the set $\{1,2,\ldots,q\}$.

Correspondingly, the initialisation $T \leftarrow M_R$ should be viewed as $T^{(0)} \leftarrow M_R$ since all paths depicted by M_R are *direct* (single "edges" (arrows) labelled by R) —no intermediate points on any of them.

Thus, not only the initialisation is correct but also the innermost loop behaves correctly:

The right hand side (before the execution of the assignment instruction inside the loop) holds recorded paths —if such were recorded— $i \to k$, $i \to j$ and $j \to k$ that have *no inner points outside* $\{1,2,\ldots,j-1\}$ the "record" being a 1 or 0 in the corresponding matrix entries $T^{(j-1)}(i,k)$, $T^{(j-1)}(i,j)$, and $T^{(j-1)}(j,k)$ according as the foregoing paths were detected or not.

The left hand side $T^{(j)}(i,k)$ records paths $i \to k$ that *either* have no inner points outside $\{1,2,\ldots,j-1\}$ (term $T^{(j-1}(i,k)$ to the right of "$\leftarrow$") OR by virtue of the concatenation of $i \to j$ and $j \to k$ they have inner points from 1 *up to and including* j; justifying us to place a "j" superscript on T to the left of "$\leftarrow$".

Given the semantics of

$$T^{(j)}(i,k) \leftarrow T^{(j-1)}(i,k) + T^{(j-1)}(i,j) \times T^{(j-1)}(j,k) \tag{2}$$

as noted in the above two paragraphs, and since $T^{(0)}$ on line 1 is initialised correctly as already noted in the preceding boxed remark, assignment (2) is *correct* for *all* j (and all i,k). In particular, for $j=n$, we have $T^{(n)}(i,k)=1$ iff there is a path from i to k that *uses no internal nodes outside* $\{1,2,\ldots,n\}$. In short, "$T^{(n)}(i,k)=1$ iff there is a path from i to k" is correct, period; *without* the preceding italicised qualification.

Do we need to use the superscripts in $T^{(q)}$ and to introduce *new matrices* $T^{(j)}$, one for each $j=1,2,\ldots,n$, beyond their use notionally in the justification of correctness?

No.

We can record the $T^{(q)}$ entries into T —that is, we store $T^{(q)}$ into T at each step that the former is updated— without altering the analysis above: Namely, if the $T^{(j-1)}(i,k)$, $T^{(j-1)}(i,j)$ and $T^{(j-1)}(j,k)$ in (2) have already been stored in T, then if paths (1) and (1′) have already been recorded as $T^{(j-1)}(i,j)=1$ and $T^{(j-1)}(j,k)=1$, according to the previous analysis, then they are stored as $T(i,j)=1$ and $T(j,k)=1$ in the suggested algorithm, without superscripts. Thus using the "$T(i,k) \leftarrow T(i,k)+T(i,j)\times T(j,k)$" in the innermost loop —no "(j)" superscript on the leftmost T!— the algorithm above correctly updates the left hand side $T(i,k)$ since the right hand side is assumed correct and the assignment statement can be viewed as having two steps: One, obtaining $T^{(j)}(i,k)$ as an "intermediate step" and Two, copying this result into the left hand side of "$\leftarrow$" as $T(i,k)$.

The latter entry is yes/no (1/0) without reference to inner nodes.

Before we turn to the timing assessment of the algorithm we give it the form that is prevalent in the literature.

```
T     ←   M_R
for   j = 1 to      n
  for i = 1 to      n
      if  T(i, j)   = 1   then
      for k = 1     to    n
          T(i,k) ←T(i,k)+T(j,k)
```

By inspection of the above program and the presence of the three nested loops it is trivial that the algorithm's run time is bounded by $O(n^3)$ Boolean $+$ operations (no $\times$!).

3.3 Equivalence Relations

Equivalence relations must be *on some set* A, since we require reflexivity. They play a significant role in many branches of mathematics and even in computer science. For example, the minimisation process of finite automata (a topic that we will not cover) relies on the concept of equivalence relations.

3.3.1 Definition A relation R on A is an equivalence relation, provided it is all of

1. Reflexive
2. Symmetric
3. Transitive □

3.3.2 Example The following are equivalence relations

- $\{(1, 1)\}$ on $A = \{1\}$.
- $=$ (or $\mathbf{1}_A$ or Δ_A) on A.
- Let $A = \{1, 2, 3\}$. Then $R = \{(1, 2), (1, 3), (2, 3), (2, 1), (3, 1), (3, 2), (1, 1), (2, 2), (3, 3)\}$ is an equivalence relation on A.
- $\mathbb{N}^2$ is an equivalence relation on $\mathbb{N}$. □

Here is a longish, more sophisticated example, that is central in number theory. We will have another instalment of it after a few definitions and results.

3.3.3 Example (Congruences) Fix an $m \geq 2$. We define the relation $\equiv_m$ on $\mathbb{Z}$ by

$$x \equiv_m y \text{ iff } m \mid (x - y)$$

Recall that "|" is the "divides with zero remainder" relation.

A notation that is very widespread in the literature is to split the symbol "$\equiv_m$" into two and write

$$x \equiv y \pmod{m} \text{ instead of } x \equiv_m y$$

"$x \equiv y \pmod{m}$" and $x \equiv_m y$ are read "x is *congruent* to y *modulo* m (or just 'mod m')". Thus "$\equiv_m$" is the congruence (mod m) short symbol, while "$\equiv \ldots \pmod{m}$" is the long two-piece symbol. *We will be using the short symbol.*

We verify the required properties for $\equiv_m$ to be an equivalence relation.

1. Reflexivity: Indeed, $m \mid (x - x)$, hence $x \equiv_m x$.
2. Symmetry: Clearly, if $m \mid (x - y)$, then $m \mid (y - x)$. I translate: If $x \equiv_m y$, then $y \equiv_m x$.
3. Transitivity: Let $m \mid (x - y)$ and $m \mid (y - z)$. The first says that, for some k, $x - y = km$. Similarly the second says, for some n, $y - z = nm$. Thus, adding these two equations I get $x - z = (k + n)m$, that is, $m \mid (x - z)$. I translate: If $x \equiv_m y$ and $y \equiv_m z$, then also $x \equiv_m z$. □

3.3.4 Definition (Equivalence classes) Given an equivalence relation R on A. The *equivalence class* of an element $x \in A$ is $\{y \in A : xRy\}$. We use the symbol $[x]_R$, or just $[x]$ if R is understood, for the equivalence class.

3.3.5 Remark Suppose an equivalence relation R on A is given.

By reflexivity, xRx, for any x. Thus $x \in [x]_R$, hence all equivalence classes are nonempty. □

Be careful to distinguish the brackets $\{\ldots\}$ from these $[\ldots]$. It is *not* a priori obvious that $x \in [x]_R$ *until you look at the definition* 3.3.4! $[x]_R \neq \{x\}$.

The symbol A/R denotes the *quotient class* of A with respect to R, that is,

$$A/R \stackrel{Def}{=} \{[x]_P : x \in A\}$$

□

This is the time to introduce "**Principle 3**"[12] of set formation.

3.3.6 Remark (Principle 3) Suppose that the *class* $\mathbb{F}$ *is indexed* by some (or all) members of a *set* A. Then $\mathbb{F}$ is a set.

Being *indexed* by (some) members of a set A *means* that —for *every* $x \in \mathbb{F}$— we have attached to it as "*label(s)*" (each often depicted as a subscript or superscript) some member(s) of A.

We *must* ensure that once a label is used it is *not used again* for another $y \in \mathbb{F}$.

Thus, if $\mathbb{F} = \{a, b, c\}$, then $\{a_1, b_{13,19,0}, c_{42}\}$ is a valid labelling with members from $\mathbb{N}$.[13] $\{a_{1,13}, b_{13}, c_{19}\}$ is not correctly labelled (same label, 13, twice), while the labelling $\{a_{1,42}, b_{13}, C\}$ is also invalid (C was not labelled):

In sum, we can label an object in $\mathbb{F}$ with many labels, but we *may not use the same label twice* to label two objects of $\mathbb{F}$ and *we may not leave any object of* $\mathbb{F}$ *unlabelled.*

Note that in 3.3.4 we have labelled every $X \in A/R$ by a member of A by virtue of the fact that any X is an $[a]_R$ We can use a or any (or all) $x \in [a]_R$ to label X.

Two things:

1. The presence of a valid (correct) labelling from a *set* A ensures that the *labelled class* is a *set* as it —*intuitively*!— *has no more members* than the *set* of labels (I can spend many —or even all— of available labels on *one* set of $\mathbb{F}$, but I *may not* reuse a label, so I have *at least as many labels as there are members in* $\mathbb{F}$).
   Thus $\mathbb{F}$ is as "small" as a *set*, and thus is a set itself. Some people call Principle 3 the *size limitation doctrine*.[14]
2. Why can't I use the Principles 0–2 to argue that $\mathbb{F}$, labelled by A, is a set? Well, because these principles are notorious in not telling me when a stage exists after *infinitely many stages of construction* that I might have if, say, I were to build one set for each natural number:

[12] This is the last Principle, I promise!

[13] b has three labels attached to it.

[14] Practitioners on the foundations of set theory felt that paradoxes occurred in connection with enormous classes.

$$A_0, A_1, \ldots, A_n, \ldots$$

Suppose the nature of *each* A_i —for each $i \geq 0$— is such that each A_{i+1} is built at stage Σ_{i+1} that is astronomically later than the stage Σ_i at which A_i was built.
Thus we get an infinite sequence of stages, wildly apart! How can I *justify* —just on the basis of Principles 0-2— the *existence* of a stage Σ that is *after all* the Σ_i, in order to build the class $\{A_0, A_1, \ldots, A_n, \ldots, \}$ as a *set*? □

We can now state the obvious:

3.3.7 Theorem *A/R is a set for any set A and equivalence relation R on A.*

Proof A provides labels for all members of A/R. Now invoke Principle 3. □

Now that we have had an excuse to introduce Principle 3 early, and applied it to the easy example above let us do the following exercise:

3.3.8 Exercise Show that it was *not* necessary to apply the new Principle to prove 3.3.7.
Specifically show that the Lemma follows by Principles 0–2 implicitly via 2.3.6.
Hint. You will need, of course, to find a superset of A/R, that is, a class X that demonstrably is a set, and satisfies $A/R \subseteq X$. □

3.3.9 Lemma *Let P be an equivalence relation on A. Then $[x] = [y]$ iff $x\,P\,y$ —where we have omitted the subscript $_P$ from the $[\ldots]$-notation.*

Proof ($\rightarrow$) part. By reflexivity, $x \in [x]$ (3.3.5). The assumption then yields $x \in [y]$ and therefore $y\,P\,x$ by 3.3.4. Symmetry gives us $x\,P\,y$ now.

($\leftarrow$) part. Let $z \in [x]$. Then $x\,P\,z$. The assumption yields $y\,P\,x$ (by symmetry), thus, transitivity yields $y\,P\,z$. That is, $z \in [y]$, proving

$$[x] \subseteq [y]$$

By swapping letters we have proved above that $y\,P\,x$ implies $[y] \subseteq [x]$. Now (by symmetry) our original assumption, namely $x\,P\,y$, implies $y\,P\,x$, hence also $[y] \subseteq [x]$. All in all, $[x] = [y]$. □

3.3.10 Lemma *Let R be an equivalence relation on A. Then*

(i) $[x] \neq \emptyset$, *for all* $x \in A$.
(ii) $[x] \cap [y] \neq \emptyset$ *implies* $[x] = [y]$, *for all* x, y *in* A.
(iii) $\bigcup_{x \in A} [x] = A$.

Proof

(*i*) 3.3.5.
(*ii*) Let $z \in [x] \cap [y]$. Then $x R z$ and $y R z$, therefore $x R z$ and $z R y$ (the latter by symmetry) hence $x R y$ (transitivity). Thus, $[x] = [y]$ by Lemma 3.3.9.
(*iii*) The $\subseteq$-part is obvious from $[x] \subseteq A$. The $\supseteq$-part follows from $\bigcup_{x \in A} \{x\} = A$ and $\{x\} \subseteq [x]$. □

The properties (*i*)–(*iii*) are characteristic of the notion of a *partition of a set*.

3.3.11 Definition (Partitions) Let F be a family of subsets of A. It is a *partition of* A iff all of the following hold:

(*i*) For all $X \in F$ we have that $X \neq \emptyset$.
(*ii*) If $\{X, Y\} \subseteq F$ and $X \cap Y \neq \emptyset$, then $X = Y$.
(*iii*) $\bigcup F = A$. □

3.3.12 Remark Often a partition F is given as an indexed family of sets denoted by $(F_a)_{a \in I}$, where I is the indexing set.

Less informatively we may write $(F_a)_{a \in I}$ as

$$\{F_a, F_b, F_c, \ldots\}$$

where the F_a are the $X, Y, \ldots$ of the definition above. □

There is a natural affinity between equivalence relations and partitions on a set A. In fact,

3.3.13 Theorem *Given a partition F on a set A. This leads to the definition of an equivalence relation P whose equivalence classes are precisely the sets of the partition, that is $F = A/P$.*

Proof First we define P:

$$x P y \overset{Def}{\text{ iff }} (\exists X \in F)\{x, y\} \subseteq X \tag{1}$$

Observe that

(i) P is reflexive: Take any $x \in A$. By 3.3.11(iii), there is an $X \in F$ such that $x \in X$, hence $\{x, x\} \subseteq X$. Thus $x P x$.
(ii) P is, trivially, symmetric since there is no order in $\{x, y\}$ to make a difference in definition (1).

(iii) P is transitive: Indeed, let $x\,P\,y\,P\,z$. Then $\{x, y\} \subseteq X$ and $\{y, z\} \subseteq Y$ for some X, Y in F.
Thus, $y \in X \cap Y$ hence $X = Y$ by 3.3.11(ii). Hence $\{x, z\} \subseteq X$, therefore $x\,P\,z$.

So P is an equivalence relation. Let us compare its equivalence classes with the various $X \in F$.

Now $[x]_P$ (denoted without the subscript $_P$ in the remaining proof) is

$$\{y : x\,P\,y\} \tag{2}$$

Let us compare $[x]$ with the unique $X \in F$ that contains x —why unique? By 3.3.11(ii). Thus,

$$y \in [x] \overset{(2)}{\Longleftrightarrow} x\,P\,y \overset{(1)}{\Longleftrightarrow} x \in X \wedge y \in X \overset{x \in X \text{ is } \mathbf{t}}{\Longleftrightarrow} y \in X$$

Thus $[x] = X$. □

3.3.14 Example (Another look at congruences) Euclid's theorem for the division of integers states, where $\mathbb{Z}$ is the set of all integers, negative, positive and 0:

If $a \in \mathbb{Z}$ and $0 < m \in \mathbb{Z}$, then *there are unique* q and r such that

$$a = mq + r \text{ and } 0 \leq r < m \tag{1}$$

There are many proofs, but here is one: The set

$$T = \{x : 0 \leq x = a - mz, \text{ for some z}\}$$

is not empty. For example, if $a > 0$, then take $z = 0$ to obtain $x = a > 0$ in T. If $a = 0$, then take $z = 0$ to obtain $x = 0$. Finally, if $a < 0$, then take $z = -2|a|$[15] to obtain $x = -|a| + 2m|a| = |a|(2m - 1) > 0$. Since $m \geq 1$ we have $2m \geq 2$.

Let then r be the *smallest* $x \geq 0$ in T. If there is one x that works (as we just showed), then possibly there are more. But we *cannot* have an infinite descending sequence of *nonnegative* integers

$$\ldots < x''' < x'' < x' < x$$

thus, in particular, we cannot have such a sequence *in* T.

There are just $x + 1$ numbers from 0 to x inclusive! *So a smallest* $x \in T$ *that works exists*. The *corresponding* "z" to the smallest $x = r$ let us call q. So we have

$$a = mq + r$$

Can $r \geq m$? If so, them write $r = k + m$, where $k = r - m \geq 0$ and $k < r$ (recall that $m > 0$). I got

[15] Absolute value.

$$a = m(q+1) + k$$

As $k < r$ I have contradicted the minimality of r.

This proves that $r < m$ (that $r \geq 0$ is trivial; why?)

We have proved *existence of at least one pair* q and r that works for (1). How about uniqueness? Well, the worst thing that can happen is to have two representations (1). Here is another one:

$$a = mq' + r' \text{ and } 0 \leq r' < m \tag{2}$$

As both r and r' are $< m$, their "distance" (absolute difference) is also $< m$.

Now, from (1) and (2) we get

$$m|q - q'| = |r - r'| \tag{3}$$

This cannot be unless $q = q'$ (*in which case* $r = r'$, *therefore uniqueness is proved*).

Wait: Why "it cannot be" if $q \neq q'$? Because then $|q - q'| \geq 1$ thus the lhs of "=" in (3) is $\geq m$ but the rhs is $< m$.

We now take a deep breath!

Now, back to congruences! The above was just a preamble!

Fix an $m > 1$[16] and consider the congruences $x \equiv_m y$. What are the equivalence classes?

Better question is what representative members are convenient to use for each such class? Given that $a \equiv_m r$ by (1), and using Lemma 3.3.9 we have $[a]_m = [r]_m$.

r is a far better representative than a for the class $[a]_m$ as it is "normalised".

Thus, we have just m equivalence classes $[0], [1], \ldots, [m-1]$.

Wait! Are they distinct? Yes! Since $[i] = [j]$ is the same as $i \equiv_m j$ (3.3.9) and, since $0 < |i - j| < m$, m cannot divide $i - j$ with 0 remainder, we cannot have $[i] = [j]$.

OK. How about missing some? We are not, for any a is uniquely expressible as $a = m \cdot q + r$, where $0 \leq r < m$. Since $m \mid (a - r)$, we have $a \equiv_m r$, i.e., (by 3.3.4) $a \in [r]$. □

3.3.15 Example (A practical example) Say, I chose $m = 5$. Where does $a = -110987$ belong? I.e., in which $[\ldots]_5$ class out of $[0]_5, [1]_5, [2]_5, [3]_5, [4]_5$?

Well, let's do primary-school-learnt long division of $-a$ divided by 5 and find quotient q and remainder r. We find, in this case, $q = 22197$ and $r = 2$. These satisfy

$$-a = 22197 \times 5 + 2$$

[16] Congruences modulo $m = 1$ are trivial and not worth considering.

Thus,

$$a = -22197 \times 5 - 2 \tag{1}$$

(1) can be rephrased as

$$a \equiv_5 -2 \tag{2}$$

But easily we check that $-2 \equiv_5 3$ (since $3 - (-2) = 5$). Thus,

$$a \in [-2]_5 = [3]_5 \qquad \square$$

3.3.16 Exercise Can you now *easily* write the same a above as

$$a = Q \times 5 + R, \text{ with } 0 \leq R < 5?$$

Show all your work. $\square$

3.4 Partial Orders

This section introduces one of the most important kind of binary relations in set theory and mathematics in general: The *partial order* relations.

3.4.1 Definition (Converse or inverse relation of $\mathbb{P}$) For any relation $\mathbb{P}$, the symbol $\mathbb{P}^{-1}$ stands for the *converse* or *inverse* relation of $\mathbb{P}$ and is defined as

$$\mathbb{P}^{-1} \stackrel{Def}{=} \{(x, y) : y\mathbb{P}x\} \tag{1}$$

Equivalently to (1), we may define $x\mathbb{P}^{-1}y$ iff $y\mathbb{P}x$. $\square$

3.4.2 Definition ("$(a)\mathbb{P}$" notation) For any relation $\mathbb{P}$ we write "$(a)\mathbb{P}$" to indicate the *class* —might fail to be a set— of *all outputs* of $\mathbb{P}$ *on (caused by) input* a. That is,

$$(a)\mathbb{P} \stackrel{Def}{=} \{y : a\,\mathbb{P}\,y\}$$

If $(a)\mathbb{P} = \emptyset$, then $\mathbb{P}$ is *undefined* at a —that is, $a \notin \text{dom}(\mathbb{P})$. This undefinedness statement is often denoted simply by "$(a)\mathbb{P}\uparrow$" and is naturally read as "$\mathbb{P}$ is *undefined* at a".

If $(a)\mathbb{P} \neq \emptyset$, then $\mathbb{P}$ is *defined* at a —that is, $a \in \text{dom}(\mathbb{P})$. This definedness statement is often denoted simply by "$(a)\mathbb{P}\downarrow$" and is naturally read as "$\mathbb{P}$ is *defined* at a". $\square$

3.4.3 Exercise Give an example of a specific relation $\mathbb{P}$ and *one* specific object (set or atom) a such that $(a)\mathbb{P}$ is a proper class. $\square$

3.4.4 Remark We note that for any $\mathbb{P}$ and a,

$$(a)\mathbb{P}^{-1} = \{y : a\mathbb{P}^{-1}y\} = \{y : y\mathbb{P}a\}$$

Thus,

$$(a)\mathbb{P}^{-1}\uparrow \text{ iff } a \notin \text{ran}(\mathbb{P})$$

and

$$(a)\mathbb{P}^{-1}\downarrow \text{ iff } a \in \text{ran}(\mathbb{P})$$

□ 

3.4.5 Exercise Show that $(\mathbb{P} \mid \mathbb{A})^{-1} = \mathbb{P}^{-1} \mid \mathbb{A}$. □

3.4.6 Definition (Partial order) A relation $\mathbb{P}$ is called a *partial order* or just an *order*, iff it is

(1) *irreflexive* (i.e., $x\mathbb{P}y \to x \neq y$ for all x, y), and

(2) *transitive*.

It is emphasised that in the interest of generality —for much of this section (until we say otherwise)— $\mathbb{P}$ need not be a set.

Some people call this a *strict order* as it imitates the "$<$" on, say, the natural numbers.

□

3.4.7 Remark (1) We will normally use the symbol "$<$" in *the abstract setting* to denote *any unspecified order* $\mathbb{P}$, and it will be pronounced "less than".

It is *hoped* that the context will not allow confusion with any concrete use of the symbol $<$ on numbers (say, on the reals, natural numbers, etc.).

(2) If the order $<$ is a subclass of $\mathbb{A} \times \mathbb{A}$ —i.e., it is $<: \mathbb{A} \to \mathbb{A}$— then we say that $<$ *is an order on* $\mathbb{A}$.

(3) It is easy to check and verify that, for any order $<$ and any class $\mathbb{B}$, we have that $< \cap(\mathbb{B} \times \mathbb{B})$ is an order on $\mathbb{B}$. □ 

3.4.8 Exercise Do (3) above with a simple, short proof. □

3.4.9 Example The concrete "less than", $<$, on $\mathbb{N}$ is an order, but $\leq$ is not (it is *not* irreflexive). The "greater than" relation, $>$, on $\mathbb{N}$ is also an order, but $\geq$ is not. Of course, $>=<^{-1}$.

In general, it is trivial to verify that $\mathbb{P}$ is an order iff $\mathbb{P}^{-1}$ is an order. *Exercise*! □

3.4.10 Example $\emptyset$ is an order. Moreover for any $\mathbb{A}$, $\emptyset \subseteq \mathbb{A} \times \mathbb{A}$, thus $\emptyset$ is also an order *on* $\mathbb{A}$ for the arbitrary $\mathbb{A}$. □

3.4.11 Example The relation $\in$ is irreflexive by the well known $A \notin A$, for all A. It is not transitive though. For example, if a is a set (or atom), then $a \in \{a\} \in \{\{a\}\}$ but $a \notin \{\{a\}\}$. *So it is not an order.*

Let $M = \Big\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\Big\}$. The relation $\varepsilon = \in \cap (M \times M)$ *is* transitive and irreflexive, hence it is an order (on M). *Verify*! □

3.4.12 Example $\subset$ is an order, $\subseteq$—failing irreflexivity— is not. □

3.4.13 Example Consider the order $\subset$ again. In this case we have *none* of $\{\emptyset\} \subset \{\{\emptyset\}\}$, $\{\{\emptyset\}\} \subset \{\emptyset\}$ or $\{\{\emptyset\}\} = \{\emptyset\}$. That is, $\{\emptyset\}$ and $\{\{\emptyset\}\}$ are *non comparable* items. This justifies the qualification *partial* for orders in general (Definition 3.4.18).

On the other hand, the "natural" $<$ on $\mathbb{N}$ is such that one of $x = y$, $x < y$, $y < x$ always holds for any x, y. That is, all (unordered) pairs x, y of $\mathbb{N}$ *are* comparable under $<$. This is a concrete example of a *total* order (see the "official definition" below: 3.4.19).

While *all* orders are "partial", some are total ($<$ above) and others are *nontotal* ($\subset$ above). □

3.4.14 Definition Let $<$ be a partial order on $\mathbb{A}$. We set

$$\leq \stackrel{Def}{=} \mathbf{\Delta}_{\mathbb{A}} \cup <$$

We pronounce $\leq$ "less than or equal". $\mathbf{\Delta}_{\mathbb{A}} \cup >$ is denoted by $\geq$ and is pronounced "greater than or equal".

Let us call $\leq$ a *reflexive order*. □

(1) In plain English, given $<$ on $\mathbb{A}$, we define $x \leq y$ to mean

$$x < y \vee \overbrace{x = y}^{\text{equality is } \Delta_{\mathbb{A}}}$$

for all x, y in $\mathbb{A}$.

(2) The definition of $\leq$ depends on $\mathbb{A}$ due to the presence of $\mathbf{\Delta}_{\mathbb{A}}$. *There is no such dependency on a "reference" class in the case of* $<$.

(3) We remind ourselves once more here that the symbols $<$ and $\leq$ —and their pronunciations— do *not* imply that we are talking about the specific ones on *numbers*. It is just a harmless (I *hope*) notational devise, but *unless said explicitly otherwise,* "$<$" *and* "$\leq$" *are any orders.*

3.4.15 Lemma *For any* $<: \mathbb{A} \to \mathbb{A}$, *the associated relation* $\leq$ *on* $\mathbb{A}$ *is* reflexive, antisymmetric *and* transitive.

Proof (1) Reflexivity is trivial.

(2) For antisymmetry, let $x \leq y$ and $y \leq x$. If $x = y$ then we are done, so assume the remaining case $x \neq y$ (i.e., $(x, y) \notin \mathbf{\Delta}_{\mathbb{A}}$). Then the hypothesis becomes $x < y$ and $y < x$, therefore $x < x$ by transitivity, contradicting the irreflexivity of $<$.

(3) As for transitivity let $x \leq y$ and $y \leq z$.

(a) If $x = z$, then $x \leq z$ (see the  -remark after 3.4.14) and we are done.

(b) The remaining case is $x \neq z$. Now, if it is $x = y$ or $y = z$ (but not both (why?)), then we are done again. So it remains to consider $x < y$ and $y < z$. By transitivity of $<$ we get $x < z$, hence $x \leq z$, since $< \subseteq \leq$. □

3.4.16 Lemma *Let $\mathbb{P}$ on $\mathbb{A}$ be reflexive, antisymmetric and transitive.*
Then $\mathbb{P} - \mathbf{\Delta}_{\mathbb{A}}$ is an order on $\mathbb{A}$.

Proof Since

$$\mathbb{P} - \mathbf{\Delta}_{\mathbb{A}} \subseteq \mathbb{P} \tag{1}$$

it is clear that $\mathbb{P} - \mathbf{\Delta}_{\mathbb{A}}$ is *on* $\mathbb{A}$. It is also clear that it is irreflexive. We only need verify that it is transitive.

So let

$$(x, y) \text{ and } (y, z) \text{ be in } \mathbb{P} - \mathbf{\Delta}_{\mathbb{A}} \tag{2}$$

By (1) (or (2))

$$(x, y) \text{ and } (y, z) \text{ are in } \mathbb{P} \tag{3}$$

hence

$$(x, z) \in \mathbb{P}$$

by transitivity of $\mathbb{P}$.

Can $(x, z) \in \mathbf{\Delta}_{\mathbb{A}}$, i.e., can $x = z$? No, for antisymmetry of $\mathbb{P}$ and (3) would imply $x = y$, i.e., $(x, y) \in \mathbf{\Delta}_{\mathbb{A}}$ contrary to (2).

So, $(x, z) \in \mathbb{P} - \mathbf{\Delta}_{\mathbb{A}}$. □

3.4.17 Remark Often in the literature, but decreasingly so, it is the "reflexive order" $\leq\colon \mathbb{A} \to \mathbb{A}$ that is defined as a "partial order" by the requirements that it is *reflexive*, *antisymmetric* and *transitive*. Then $<$ is obtained as in Lemma 3.4.16, namely, as "$\leq - \mathbf{\Delta}_{\mathbb{A}}$". Lemmas 3.4.15 and 3.4.16 show that the two approaches are interchangeable, but the "modern" approach of Definition 3.4.6 avoids the nuisance of having to tie the notion of order to some particular "field" $\mathbb{A}$ (3.1.6).

For us "$\leq$" is the *derived* notion defined in 3.4.14. □ 

3.4.18 Definition (PO Class) If $<$ is an order on a class $\mathbb{A}$, we call the *informal* pair $(\mathbb{A}, <)$[17] a *partially ordered class*, or *PO class*.

If $<$ is an order on a *set* A, we call the pair $(A, <)$ a *partially ordered set* or *PO set*. Often, if the order $<$ is understood as being on $\mathbb{A}$ or A, one says that "$\mathbb{A}$ is a PO class" or "A is a PO set" respectively. □

3.4.19 Definition (Linear order) A relation $<$ on $\mathbb{A}$ is a *total* or *linear* order *on* $\mathbb{A}$ iff it is

(1) An order, and

(2) For any x, y in $\mathbb{A}$ one of $x = y, \quad x < y, \quad y < x$ holds —this is the so-called "*trichotomy*" property.

If $\mathbb{A}$ is a class, then the informal pair $(\mathbb{A}, <)$ is a *linearly ordered class* —in short, a *LO class*.

If $\mathbb{A}$ is a set, then the pair $(\mathbb{A}, <)$ is a *linearly ordered set* —in short, a *LO set*.

One often calls just $\mathbb{A}$ a LO class or LO set (as the case warrants) when $<$ is understood from the context. □

3.4.20 Example The standard $<: \mathbb{N} \to \mathbb{N}$ is a total order, hence $(\mathbb{N}, <)$ is a LO set.

3.4.21 Definition (Minimal and minimum elements) Let $<$ be an order and $\mathbb{A}$ some class.

We are *not* postulating that $<$ is on $\mathbb{A}$.

An element $a \in \mathbb{A}$ is *$a <$-minimal element in* $\mathbb{A}$, or *$a <$-minimal element of* $\mathbb{A}$, iff $\neg(\exists x \in \mathbb{A})x < a$ —in words, there is *nothing below a in* $\mathbb{A}$.

$m \in \mathbb{A}$ is *$a <$-minimum element in* $\mathbb{A}$ iff $(\forall x \in \mathbb{A})m \leq x$, in words, all x in $\mathbb{A}$ satisfy $m \leq x$.

We also use the terminology minimal or minimum *with respect to* $<$, instead of $<$-minimal or $<$-minimum.

If $a \in \mathbb{A}$ is $>$-minimal in $\mathbb{A}$, that is $\neg(\exists x \in \mathbb{A})x > a$, we call a a *$<$-maximal* element in $\mathbb{A}$. Similarly, a $>$-minimum element —$(\forall x \in \mathbb{A})m \geq x$— is called a *$<$-maximum*.

If the order $<$ is understood, then the qualification "$<$-" is omitted. □

[17] Formally, $(\mathbb{A}, <)$ is *not* an ordered pair since $\mathbb{A}$ may be a proper class and we do not allow class *members* —e.g., in $\{\mathbb{A}, \{\mathbb{A}, <\}\}$— to be proper classes. We may think then of "$(\mathbb{A}, <)$" as *informal* notation that simply "ties" $\mathbb{A}$ and $<$ together. Alternatively, if we are really determined to have class pairs (*we are not!*), we can *define* pairing with proper classes as components, for example as $(\mathbb{A}, \mathbb{B}) =^{Def} (\mathbb{A} \times \{0\}) \cup (\mathbb{B} \times \{1\})$. For our part we will have no use for such formality, and will consider $(\mathbb{A}, <)$ in only the *informal* sense.

3.4.22 Remark In particular, if a ($\in \mathbb{A}$) is *not* in the *field* $\mathrm{dom}(<) \cup \mathrm{ran}(<)$ (cf. 3.1.7) of $<$, then a is *both* $<$-minimal and $<$-maximal *in* $\mathbb{A}$. For example, $(\exists x \in \mathbb{A})x < a$ is false in this case since if, for some x, we have $x \in \mathbb{A}$ and also $x < a$, then $a \in \mathrm{ran}(<)$; impossible.

Because of the duality between the notions of minimal/maximal and minimum/maximum, we will mostly deal with the $<$-notions whose results can be trivially translated for the $>$-notions.

Note how the notation learnt from 3.4.2 and 3.4.1 and 3.4.4 can *simplify*

$$\neg(\exists x \in \mathbb{A})x < a \tag{1}$$

Note that (1) says that *no x is in both $\mathbb{A}$ and $(a) >$*.[18] That is, a is $<$-minimal in $\mathbb{A}$ iff

$$\mathbb{A} \cap (a) > = \emptyset \tag{2}$$

□

3.4.23 Example 0 is *minimal*, also *minimum*, in $\mathbb{N}$ with respect to the "standard" ordering.

In $\mathscr{P}(\mathbb{N})$, $\emptyset$ is both $\subset$-minimal and $\subset$-minimum. On the other hand, all of $\{0\}$, $\{1\}$, $\{2\}$ are $\subset$-minimal in $\mathscr{P}(\mathbb{N}) - \{\emptyset\}$ but *none* are $\subset$-*minimum* in that set.

Observe from this last example that minimal elements in a class are *not* unique. □

3.4.24 Remark (Hasse diagrams) There is a neat pictorial way to depict orders on finite sets known as "*Hasse diagrams*". To do so one creates a so-called "*graph*" of the finite PO set $(A, <)$ where $A = \{a_1, a_2, \ldots, a_n\}$.

How? The graph consists of n *nodes* —which are drawn as points— each labeled by one a_i. The graph also contains 0 or more *arrows* that connect nodes. These arrows are called *edges*.

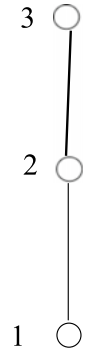
When we depict an arbitrary R on a finite set like A we draw *one* arrow (edge) *from a_i to a_j* iff the two *relate*: $a_i R a_j$.

In Hasse diagrams for PO sets $(A, <)$ we are more selective: We say that b *covers* a iff $a < b$, but there is no c such that $a < c < b$. In a Hasse diagram we will

1. draw an edge from a_i to a_j iff a_j covers a_i.
2. by convention we will draw b higher than a on the page if b covers a.
3. given the convention above, using "arrow-heads" is superfluous: our edges are plain line segments.

So, let us have $A = \{1, 2, 3\}$ and $< = \{(1, 2), (1, 3), (2, 3)\}$.

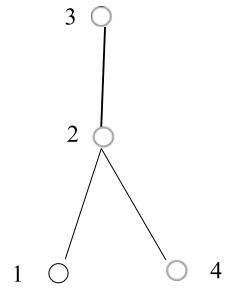
[18] $(a) >= \{x : a > x\} = \{x : x < a\}$ (cf. also 3.4.4).



The above has a minimum (1) and a maximum (3) and is clearly a linear order.

A slightly more complex one is this $(A, <)$, where $A = \{1, 2, 3, 4\}$ and

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



This one has a maximum (3), two minimal elements (1 and 4) but no minimum, and is not a linear order: 1 and 4 are not comparable. □

3.4.25 Lemma *Given an order* $<$ *and a class* $\mathbb{A}$.
(1) *If* m *is a* minimum *in* $\mathbb{A}$*, then it is also* minimal.
(2) *If* m *is a* minimum *in* $\mathbb{A}$*, then it is unique.*

Proof (1) Let m be mini*mum* in $\mathbb{A}$. Then

$$m \leq x, \text{ that is, } m = x \vee m < x \tag{i}$$

for all $x \in \mathbb{A}$. Now, prove that there is no $x \in \mathbb{A}$ such that $x < m$.

Let us argue by way of contradiction:

Let

$$\mathbb{A} \ni a < m \tag{ii}$$

By (i) I also have

$$m = a \vee m < a \qquad (iii)$$

Now, by irreflexivity, (ii) rules out $a = m$. So, (iii) nets $m < a$. (ii) and (iii) and transitivity yield $a < a$; contradiction ($<$ is irreflexive). Done.

(2) Let m and n both be minima in $\mathbb{A}$. Then $m \leq n$ (with m posing as minimum) and $n \leq m$ (now n is so posing), hence $m = n$ by antisymmetry of "$\leq$" (Lemma 3.4.15). □

3.4.26 Example Let m be $<$-minimal in $\mathbb{A}$.

Let us *attempt* to "show" that it is also $<$-minimum (this is, of course, doomed to fail due to 3.4.23 and 3.4.25(2) —but the "faulty proof" below is interesting):

By 3.4.21 we have that *there is no* x *in* $\mathbb{A}$ *such that* $x < m$.

Another way to say this is:

$$\textit{For all } x \in \mathbb{A}, \text{ I have the negation of "} x < m \text{", that is, I have } \neg x < m. \qquad (1)$$

But from "our previous math" (high school? university?) $\neg x < m$ is equivalent to $m \leq x$. Thus (1) says $(\forall x \in \mathbb{A}) m \leq x$, in other words, m is the minimum in $\mathbb{A}$.

Do you believe this? (Don't!) If the order is not total, then *the disjunction of* $x < m, x = m, m < x$ *may fail to be true*, and thus $\neg m < x$ and $x < m \vee x = m$ are *not necessarily* equivalent. See also the counterexample to such expectation in 3.4.13 and also 3.4.23. □

3.4.27 Lemma *If* $<$ *is a* linear *order on* $\mathbb{A}$*, then every minimal element is also minimum.*

Proof The "false proof" of the previous example is valid under the present circumstances. □

The following type of relation has fundamental importance for set theory, and mathematics in general.

3.4.28 Definition 1. An order $<$ satisfies the *minimal condition*, in short *it has MC*, iff *every nonempty* $\mathbb{A}$ has $<$-minimal elements.

2. If a *total* order $<: \mathbb{B} \to \mathbb{B}$ has MC, then it is called a *well-ordering*[19] *on* (or *of*) the class $\mathbb{B}$.
3. If $(\mathbb{B}, <)$ is a LO class (or set) with MC, then it is a *well-ordered class* (or set), or *WO class* (or WO set).

□

[19] The term "well-ordering" is ungrammatical, but it is *the* terminology established in the literature!

3.4.29 Remark

1. What Definition 3.4.28 says in case 1. is —see (2) in 3.4.22— "*if, for some fixed order* $<$ *the following statement*

$$\emptyset \neq \mathbb{A} \to (\exists a \in \mathbb{A})\mathbb{A} \cap (a) > = \emptyset \tag{1}$$

is true in set theory, for any $\mathbb{A}$, *then we say that* $<$ *has MC*".
The following observation is very important *for future reference*:
If $\mathbb{A}$ is given via a defining property $F(x)$, as

$$\mathbb{A} \overset{Def}{=} \{x : F(x)\}$$

then (1) translates —in terms of $F(x)$— into

$$(\exists a)F(a) \to (\exists a)\Big(F(a) \wedge \neg(\exists y)\big(y < a \wedge F(y)\big)\Big) \tag{2}$$

Conversely, for each formula $F(x)$ we get a class $\mathbb{A} = \{x : F(x)\}$ and thus if $<$ has MC, then we may express this fact as in (2) above.
2. Much is to be gained in applications by allowing slightly more generality to the concept of MC by *not* requiring the relation that is so equipped to be an order. To this end we will define the counterpart concepts for $<$-minimal (of Definition 3.4.21) and will also generalise Definition 3.4.28 below by Definition 3.4.32 below. □

3.4.30 Definition This time we are *not* postulating that $\mathbb{P}$ is on $\mathbb{A}$ nor that it is an order.

An element $a \in \mathbb{A}$ is *a* $\mathbb{P}$*-minimal element in* $\mathbb{A}$, or *a* $\mathbb{P}$*-minimal element of* $\mathbb{A}$, iff $\neg(\exists x \in \mathbb{A})x\mathbb{P}a$ —in words, there is *nothing below a in* $\mathbb{A}$ *if you "walk backwards along* $\mathbb{P}$". □

3.4.31 Remark

1. The defining condition for $\mathbb{P}$-minimal —$\neg(\exists x \in \mathbb{A})x\mathbb{P}a$— can be simplified.
Noting that $x\mathbb{P}a$ iff $a\mathbb{P}^{-1}x$, we have

$$\neg(\exists x \in \mathbb{A})x\mathbb{P}a \text{ iff } \mathbb{A} \cap \{x : a\mathbb{P}^{-1}x\} = \emptyset \text{ iff } \mathbb{A} \cap (a)\mathbb{P}^{-1} = \emptyset \text{ (cf. 3.4.22)}$$

2. The following observation is useful:

$$(a)\Big(\mathbb{P} \mid \mathbb{A}\Big) = \begin{cases} \emptyset & \text{if } a \notin \mathbb{A} \\ \mathbb{A} \cap (a)\mathbb{P} & \text{othw} \end{cases}$$

Indeed, in the first case we cannot have an x such that $a\Big(\mathbb{P} \mid \mathbb{A}\Big)x$ since this requires $(a, x) \in \mathbb{A}^2$ that is untenable under the condition for a. If, on the other hand, $a \in \mathbb{A}$, then

we will include in $(a)\Big(\mathbb{P} \mid \mathbb{A}\Big)$ all those x that are in both $(a)\mathbb{P} = \{x : a\mathbb{P}x\}$ and $\mathbb{A}$ (cf. 3.1.4, Notation 2). □

3.4.32 Definition A relation $\mathbb{T}$ —that is *not* necessarily an order— satisfies the *minimal condition*, in short *it has MC*, iff *every nonempty* $\mathbb{A}$ has $\mathbb{T}$-minimal elements in the sense that a $t \in \mathbb{A}$ *exists* such that there is *no* $t' \in \mathbb{A}$ satisfying $t'\mathbb{T}t$. □

3.4.33 Remark Definition 3.4.32 has a formulation identical to (1) of 3.4.29, although it is here for the *general relation with MC* —$\mathbb{T}$— as opposed to an *order* $<$ *with MC*:

$$\mathbb{A} \neq \emptyset \to (\exists a \in \mathbb{A})\mathbb{A} \cap (a)\mathbb{T}^{-1} = \emptyset \tag{1'}$$

Of course, $(a)\mathbb{T}^{-1} = \Big\{x : a\mathbb{T}^{-1}x\Big\} = \Big\{x : x\mathbb{T}a\Big\}$.

If we set $\mathbb{A} = \{x : F(x)\}$, for some formula F, then (1′) becomes the analogue (2′) of 3.4.29 (2) below.

$$(\exists a)F(a) \to (\exists a)\Big(F(a) \wedge \neg(\exists y)\big(y\mathbb{T}a \wedge F(y)\big)\Big) \tag{2'}$$

□

Often one works in a class $\mathbb{A}$ other than the class of everything, $\mathbb{U}$ ($\mathbb{A}$ might still be a proper class). It is then useful to "relativise" a relation $\mathbb{P}$ to $\mathbb{A}$ and perhaps even have this restriction —because a relational restriction is what we have in mind— have MC even if, perhaps, the unrelativised $\mathbb{P}$ does not. Thus we have

3.4.34 Definition (Relations with MC over, or relative to a class) We say that a relation $\mathbb{P}$ has MC *over* (or *on*, or *in*, or *relative* to) a class $\mathbb{A}$ if $\mathbb{P} \mid \mathbb{A}$ does. □

The proof of the following proposition relies on Exercise 3.4.5.

3.4.35 Proposition (MC over a Class Test) *A relation* $\mathbb{P}$ *has MC over a class* $\mathbb{A}$ *iff the schema*

$$\emptyset \neq \mathbb{B} \subseteq \mathbb{A} \to (\exists b \in \mathbb{B})\mathbb{B} \cap (b)\mathbb{P}^{-1} = \emptyset \tag{1}$$

is true in set theory.

Proof That $\mathbb{P}$ has MC over $\mathbb{A}$ means that $\mathbb{P} \mid \mathbb{A}$ has MC, that is, the schema

$$\emptyset \neq \mathbb{B} \to (\exists b \in \mathbb{B})\mathbb{B} \cap (b)\Big(\mathbb{P}^{-1} \mid \mathbb{A}\Big) = \emptyset \tag{2}$$

is true in set theory. We will prove that we have (1) iff we have (2). There are two directions to verify this "iff".

(I) *Assume* (1) and prove (2): Towards proving (2) start with the assumption for (2): $\mathbb{B} \neq \emptyset$.

Note that we do *not* adopt the assumption of (1) that includes $\mathbb{B} \subseteq \mathbb{A}$.

We consider two subcases for $\mathbb{B}$ vs. $\mathbb{A}$ on which $\mathbb{P}$ has MC.

- $\mathbb{B} \cap \mathbb{A} \neq \emptyset$. Using (1) we deduce from the truth of

$$\emptyset \neq \mathbb{B} \cap \mathbb{A} \subseteq \mathbb{A} \tag{2'}$$

the truth of $(\exists b \in \mathbb{B} \cap \mathbb{A})\mathbb{B} \cap \mathbb{A} \cap (b)\mathbb{P}^{-1} = \emptyset$. Since $b \in \mathbb{B} \cap \mathbb{A}$ implies $b \in \mathbb{B}$ we further conclude $(\exists b \in \mathbb{B})\mathbb{B} \cap \mathbb{A} \cap (b)\mathbb{P}^{-1} = \emptyset$ and further obtain —given $b \in \mathbb{A}$ and 3.4.31 2.—

$$(\exists b \in \mathbb{B})\mathbb{B} \cap (b)\Big(\mathbb{P}^{-1} \mid \mathbb{A}\Big) = \emptyset \tag{2''}$$

Having derived (2″) from $\mathbb{B} \neq \emptyset$, we established (2) in this subcase.
- $\mathbb{B} \cap \mathbb{A} = \emptyset$. The assumption $\mathbb{B} \neq \emptyset$ is still the active primary assumption in this subcase. However the additional subcase assumption means that if $b \in \mathbb{B}$ then $b \notin \mathbb{A}$ and by 3.4.31 we have $(b)\Big(\mathbb{P}^{-1} \mid \mathbb{A}\Big) = \emptyset$ which again implies (2″).

(II) *Assume* (2) and prove (1): So let $\emptyset \neq \mathbb{B} \subseteq \mathbb{A}$. By (2) —and only the part $\emptyset \neq \mathbb{B}$ of the assumption— we obtain (2″) above. Noting that $(b)\Big(\mathbb{P}^{-1} \mid \mathbb{A}\Big) = \mathbb{A} \cap (b)\mathbb{P}^{-1}$ by 3.4.31 2., and mindful of $\mathbb{B} \cap \mathbb{A} = \mathbb{B}$, the right hand side of "$\rightarrow$" in (2) —that is, (2″)— becomes $(\exists b \in \mathbb{B})\mathbb{B} \cap (b)\mathbb{P}^{-1} = \emptyset$. This proves (1) due to the "let" at the onset of this (2) $\rightarrow$ (1) case. □

3.4.36 Remark In practice, the *minimal condition* (MC) of an order or, indeed, of an arbitrary relation $\mathbb{P}$ is usually taken relative to a class $\mathbb{A}$, often a *set* class.

Thus it is important to reformulate (1) of Proposition 3.4.35 that succinctly states that a relation $\mathbb{P}$ (*not necessarily an order*) has MC over a class $\mathbb{A}$. □

3.4.37 Corollary *Let $\mathbb{P}$ be a relation with MC over $\mathbb{A}$. (1) of 3.4.35 is equivalent to the truth of the schema below —where $\mathbb{A}$ and $F[x]$ are arbitrary, but $\mathbb{A}$ in any one application of MC is fixed as the class inside which we do mathematics.*

$$(\exists b \in \mathbb{A})F[b] \rightarrow (\exists b \in \mathbb{A})\Big(F[b] \wedge \neg(\exists x \in \mathbb{A})\big(F[x] \wedge x\mathbb{P}b\big)\Big) \tag{†}$$

Proof (I) Assume (1) in 3.4.35 and prove (†). To this end assume the hypothesis

$$(\exists b \in \mathbb{A})F[b] \tag{‡}$$

of (†) and let us define a class $\mathbb{B}$ by

$$\mathbb{B} \overset{Def}{=} \mathbb{A} \cap \{x : F[x]\} \tag{¶}$$

By (‡) and (¶) we have $\emptyset \neq \mathbb{B} \subseteq \mathbb{A}$, thus, by (1) quoted above we get $(\exists b \in \mathbb{B})$[20] $\mathbb{B} \cap (b)\mathbb{P}^{-1} = \emptyset$. This translates (in terms of $F[x]$) into $(\exists b \in \mathbb{A})\Big(F[b] \wedge \neg(\exists x)\big(x \in \mathbb{B} \wedge b\mathbb{P}^{-1}x\big)\Big)$, which after further translation (replacing "$x \in \mathbb{B}$" and "$b\mathbb{P}^{-1}x$") becomes

$$(\exists b \in \mathbb{A})\Big(F[b] \wedge \neg(\exists x \in \mathbb{A})\big(F[x] \wedge x\mathbb{P}b\big)\Big) \tag{¶¶}$$

The conclusion part of (†) is proved.

(II) Next assume (†) and prove (1) of 3.4.35. To this end, assume hypothesis of (1) for some $\mathbb{B}$, namely,

$$\emptyset \neq \mathbb{B} \subseteq \mathbb{A} \tag{††}$$

Let us express $\mathbb{B}$ as a *class defined by a property*, that is, set

$$\mathbb{B} \overset{Def}{=} \{x \in \mathbb{A} : F[x]\}, \text{ for some formula } F[x]$$

(††) implies —in terms of $F[x]$— $(\exists b \in \mathbb{A})F[b]$, which is the same as the hypothesis of the implication in (†). Since (†) is assumed, we have its conclusion part (see (¶¶) above) —i.e., it is true under assumption (††). Let us express it *without using the notation* that employs "$F[x]$". We observe

$$\overbrace{(\exists b \in \mathbb{A})\Big(F[b] \wedge \neg \overbrace{(\exists x \in \mathbb{A})\big(F[x] \wedge \underbrace{x\mathbb{P}b}_{x \in (b)\mathbb{P}^{-1}}\big)}^{(\exists x \in \mathbb{B})}\Big)}^{(\exists b \in \mathbb{B})}$$

hence the following is true $(\exists b \in \mathbb{B})\neg(\exists x \in \mathbb{B})x \in (b)\mathbb{P}^{-1}$.

In short, $(\exists b \in \mathbb{B})\mathbb{B} \cap (b)\mathbb{P}^{-1} = \emptyset$. We have just shown the truth of the conclusion part of the implication (1) in 3.4.35 as we had set out to do. □

3.5 Functions

At last! We consider here a special case of relations that we know them as "functions". Many of you know already that a function is a relation with some special properties.

Let's make this official:

[20] That is, $(\exists b)(b \in \mathbb{B} \wedge \ldots)$ or, equivalently, $(\exists b)(b \in \mathbb{A} \wedge F[b] \wedge \ldots)$ or, equivalently, $(\exists b \in \mathbb{A})(F[b] \wedge \ldots)$ the "…" part being, *before further translation*, "$\mathbb{B} \cap (b)\mathbb{P}^{-1} = \emptyset$".

3.5.1 Definition A *function* R is a *single-valued* relation. That is, whenever we have both $x\,R\,y$ and $x\,R\,z$, we will also have $y = z$.

It is traditional to use, generically, lower case letters from among f, g, h, k to denote functions but this is by no means a requirement. □

Another way of putting it, using the notation from 3.4.2, is: A relation R is a function iff $(a)R$ is either *empty* or contains *exactly one* element.

3.5.2 Example The empty set is a relation of course, the empty set of pairs. It is also a function since

$$(x, y) \in \emptyset \wedge (x, z) \in \emptyset \rightarrow y = z$$

vacuously, by virtue of the left hand side of $\rightarrow$ being false. □

We now turn to notation and concepts specific to functions.

3.5.3 Definition (Function-specific notations, terminology) Let f be a function. First off, the *concepts* of domain, range, and —in case of a function $f : A \rightarrow B$— total and onto *are inherited from those for relations without change*. Even the notations "$a\,R\,b$" and "$(a, b) \in R$" transfer over to functions. Moreover, the notation "$(a)\,f\,\downarrow$" (correspondingly "$(a)\,f\,\uparrow$") meaning $a \in \mathrm{dom}(f)$ (correspondingly $a \notin \mathrm{dom}(f)$) and terminology "f is defined at a" (correspondingly "f is *un*defined at a") are extended to functions.

In particular, the "relational" notation $(a)\,f$ for $\{y : a\,f\,y\}$ *can always be used*.

We noted in 3.5.1 that for functions f and input a we have

$$(a)\,f = \begin{cases} \{y : a\,f\,y\} = \{b\} & \text{for some } b \\ & \text{or} \\ \emptyset & \text{if } \{y : a\,f\,y\} \text{ is empty} \end{cases}$$

But we also have an annoying *difference* in notation that is used extremely widely:

Mathematicians *normally* prefer to write $f(a) = b$ instead of $(a)\,f = \{b\}$ and $f(a)\uparrow$ (undefined at a) if $(a)\,f = \emptyset$.

The qualifier "normally" indicates frequency, but also allows some authors to differ: Notably, Kurosh (1963) writes "$a\,f$" for relations *and* functions, even omitting the brackets around the input a.

We will follow the "normally preferred" notation for functions — $f(a)$— in this work and will give reasons for this "preference" —notation "$f(a)$" over "$(a)\,f$"— when we consider the composition of *functions* below.

Worth recording: If b is such that afb or $(a, b) \in f$ and f is a function, then seeing that b is unique we have $(a)f = \{b\}$.

The relationship between "functional notation" vs. "relational notation" is summarised below.

$$\underbrace{f(a) = b}_{\text{functional notation}} \quad \text{iff} \quad \underbrace{(a)f = \{b\}}_{\text{relational notation}}$$

and

$$\underbrace{f(a)\uparrow}_{\text{functional notation}} \quad \text{iff} \quad \underbrace{(a)f = \emptyset}_{\text{relational notation}}$$

□ 

3.5.4 Definition (Images and Inverse Images) The set of *all* outputs of a function, *when the inputs come from a particular set* X, is called the *image of* X *under* f and is denoted by $f[X]$. Thus,

$$f[X] \stackrel{Def}{=} \{f(x) : x \in X\} \tag{1}$$

Note that careless notation (in many discrete mathematics texts) like $f(X)$ will *not* do. This means the input *is* X. If I want the inputs to be *from inside* X, then we must *not* use the round brackets notation.

The *inverse image* of a set Y under a function is useful as well, that is, the set of *all* inputs that generate f-outputs exclusively in Y. It is denoted by $f^{-1}[Y]$ and is defined as

$$f^{-1}[Y] \stackrel{Def}{=} \{x : f(x) \in Y\} \tag{2}$$

Pause. So far we have been giving definitions regarding functions of *one* variable. Or have we?

Not really: We have already said that the multiple-input case is subsumed by our notation. If $f : A \to B$ and A is a set of n-tuples, then f is a function of "n-variables", essentially. The binary relation that is the alias of f contains pairs like $\left((\vec{x}_n), x_{n+1}\right)$. However, we usually abuse notation and write $(\vec{x}_n)f$ instead of $\left((\vec{x}_n)\right)f$ and $f(\vec{x}_n)$ instead of $f\left((\vec{x}_n)\right)$. ◄ □

3.5.5 Remark Regarding, say, the definition of $f[X]$:

What if $f(a)\uparrow$? *How do you "collect" an "undefined value"* into a set?

Well, you don't. Both (1) and (2) have a rendering that is independent of the notation "$f(a)$".

Let us never forget that a function is no mystery; it is a *relation* and we have access to *relational notation*. Thus,

$$f[X] = \{y : (\exists x \in X)xfy\} \tag{1'}$$

$$f^{-1}[Y] = \{x : (\exists y \in Y)xfy\} \tag{2'}$$

□ 

3.5.6 Example Thus, $f[\{a\}] = \{f(x) : x \in \{a\}\} = \{f(x) : x = a\} = \{f(a)\}$.

Let now $g = \{(1, 2), (\{1, 2\}, 2), (2, 7)\}$, clearly a function. Thus, $g(\{1, 2\}) = 2$, but $g[\{1, 2\}] = \{2, 7\}$. Also, $g(5) \uparrow$ and thus $g[\{5\}] = \emptyset$.

On the other hand, $g^{-1}[\{2, 7\}] = \{1, \{1, 2\}, 2\}$ and $g^{-1}[\{2\}] = \{1, \{1, 2\}\}$, while $g^{-1}[\{8\}] = \emptyset$ since no input causes output 8. □

3.5.7 Example We saw that (3.5.3) $\mathbb{F}(a) = \emptyset$ means $(a)\mathbb{F} = \{\emptyset\}$, that is, $(a, \emptyset) \in \mathbb{F}$ or $a\mathbb{F}\emptyset$ —*not* what one might hastily conclude it means.

We have $\mathbb{F}(a) \downarrow$ here, with output the *object* "$\emptyset$", that is, it is not the case that $\mathbb{F}(a) \uparrow$. □

The following is quite useful in set theory and even has a nickname, "(the Principle of) Replacement".[21]

3.5.8 Theorem *If $\mathbb{F}$ is a function (possibly a proper class of pairs), and A is a* set*, then $\mathbb{F}[A]$ is a set.*

Proof Let

$$\emptyset \neq \mathbb{Y} = \mathbb{F}[A] \tag{†}$$

Thus, *for every* y, $y \in \mathbb{Y}$ iff for some $x \in A$, $\mathbb{F}(x) = y$.

In short, *each* $y \in \mathbb{Y}$ is *labelled* —in the sense of 3.3.6— by *all* the $x \in A$ with the property $\mathbb{F}(x) = y$.

Note that the described label-set is *valid* according to 3.3.6 since

- *No member* of $\mathbb{Y}$ is without an A-label (by (†)). These labels are in A and in $\mathbb{F}^{-1}[\{y\}]$, thus
$$A \cap \mathbb{F}^{-1}[\{y\}] \text{ is } nonempty \tag{1}$$
- The *set* $A \cap \mathbb{F}^{-1}[\{y\}]$ has no repeated members (being a set) thus the labels assigned to y are distinct, and *more importantly*
- If $y \neq y'$ are both in $\mathbb{Y}$, then they receive non overlapping labels because $\mathbb{F}^{-1}[\{y\}] \cap \mathbb{F}^{-1}[\{y'\}] = \emptyset$.

[21] Said "Principle" is given as the *Axiom* of Replacement in axiomatic set theory.

Why? Because if $z \in \mathbb{F}^{-1}[\{y\}] \cap \mathbb{F}^{-1}[\{y'\}]$, then $\mathbb{F}(z) = y$ *and* $\mathbb{F}(z) = y'$; impossible for a *function*.

By Principle 3, $\mathbb{Y}$ —being labelled by the members of A— is a set too. □

3.5.9 Corollary *If the domain of a function $\mathbb{F}$ is a set, then so is $\mathbb{F}$.*

Proof Exercise! □

3.5.1 Lambda Notation

Some texts in discrete mathematics and also in calculus will say "let $f(x)$ be a function …"

Well, "$f(x)$" is *not* a function. Correctly it is known variously as a function *call*, or function *application* or function *invocation*. "f" *is* the function here; a set or "table" — possibly infinite— of input/output pairs. Thus f is a set and $f(x)$ is an output value (when the input is x).

Computer programmers are very much aware of the distinction between a function *call* $f(x)$ and a function *definition* for f, the latter being *defined intentionally* (by behaviour) rather than *extensionally* (by *explicit listing* as a set or table of input/out pairs).

This intentional *definition* of the input/output behaviour of a function f is done by a *program*. Luckily, *unlike tables*, *all programs* being *finite* in length, can fit into a computer!

In mathematics we often want to say "let f be a function of input variables x and z …" but we are not excused to say it incorrectly as "let $f(x, z)$ be a function"; it is *not*!

We can say instead "let (or consider) $\lambda xz.f(x, z)$". This *names* both the function f and its input variables x, z. This is known as Church's λ-*notation* and is a by-product of his foundation of *computability* via "λ calculus".

3.5.10 Definition (λ-notation) The expression

$$\overset{\text{begin input list}}{\underset{\downarrow}{\lambda}} \quad \overbrace{x_1 x_2 \ldots x_n}^{\text{input list}} \quad \overset{\text{end input list}}{\underset{\downarrow}{\cdot}} \quad \overbrace{rule}^{\text{the outputs rule}} \tag{1}$$

denotes a *function* with input variables $x_1, x_2, \ldots, x_n$ and output computed according to the "*rule*" following the end-of-input dot. We can use "vector notation" for the input list and write $\vec{x}_n$ or just $\vec{x}$, if n is understood or is unimportant, for $x_1, x_2, \ldots, x_n$. Then (1) morphs into

$$\lambda \vec{x}_n.rule \tag{1'}$$

Examples:

1. $\lambda x.x + 1$ but also $\lambda y.y + 1$ and $\lambda u.u + 1$. The successor function over the natural numbers. The variables "x, y" and "u" are not able to accept substitutions —unlike the "x" in "$x + 1$" or "$x^2 - 30x + 5$".
   That is, they are "bound" or "dummy" variables just as x is in this expression $\sum_{x=1}^{100} x^2$.
2. $\lambda xw.w^2$. This function inputs x and w, then ignores x and returns w^2 as output.
3. We can give a short (letter) name to a function as always. Thus, we can say "Let $f = \lambda xw.w^2$". Then $f(x, w) = w^2$ and $f(5, 2) = 2^2 = 4$. □

3.5.2 Kleene Extended Equality for Function Calls

When $f(a) \downarrow$, then $f(a) = f(a)$ as is naturally expected. What happens when $f(a) \uparrow$? This begs a more general question that we settle as follows:

3.5.11 Definition (Kleene Equality) Kleene (Kleene (1943)) *extended equality* to include the case when the two sides of "=" are calls $f(\vec{a})$ and $g(\vec{b})$ that are both undefined. For such cases validate the equality. In symbols

$$f(\vec{a}) = g(\vec{b}) \equiv \overbrace{f(\vec{a}) \uparrow \wedge\, g(\vec{b}) \uparrow}^{\text{Case 1}} \vee \overbrace{(\exists z)\Big(f(\vec{a}) = z \wedge g(\vec{b}) = z\Big)}^{\text{Case 2}}$$

There is no universal agreement in the literature as to whether or not to use a *new* symbol for the *extended equality*. We will not do so use, but those (publications, not individuals) who do, use "$\simeq$" as in $f(\vec{a}) \simeq g(\vec{b})$. □

3.5.12 Example Let $g = \{(1, 2), (\{1, 2\}, 2), (2, 7)\}$. Then, $g(1) = g(\{1, 2\})$ and $g(1) \neq g(2)$. Moreover, $g(3) = g(4)$ (both undefined). □

3.5.13 Definition A function f is 1-1 if for all x, y and z where $f(x) = f(y) = z$ we obtain $x = y$. We can also say f is 1-1 iff xfz and yfz imply $x = y$. □

The presence of z (the definedness at x and y) in 3.5.13 ensures that we will not expect anything unreasonable, like $3 = 4$, in the context of a 1-1 function f where $f(3) \uparrow = f(4) \uparrow$.

3.5.14 Example $\{(1, 1)\}$ and $\{(1, 1), (2, 7)\}$ are 1-1. $\{(1, 0), (2, 0)\}$ is not. $\emptyset$ is 1-1 *vacuously*. □

3.5.15 Exercise Prove that if f is a 1-1 function, then the relation converse f^{-1} is a function (that is, single-valued). □

3.5.16 Definition (1-1 Correspondence) A function $f : A \to B$ is called a *1-1 correspondence* iff it is all three: 1-1, total and onto.

Often we say that A and B are *in 1-1 correspondence* writing $A \sim B$, often omitting mention of the function that *is* the 1-1 correspondence. □

The terminology is derived from the fact that every element of A is paired with precisely one element of B and vice versa.

3.5.17 Exercise Show that $\sim$ is a symmetric and transitive relation on sets. □

3.5.3 Function Composition

3.5.18 Remark Composition of functions is inherited from the composition of relations. Thus, $f \circ g$ for two functions naturally means

$$x \; f \circ g \; y \text{ iff, for some } z, \; x \; f \; z \; g \; y \tag{1}$$

In particular,

$f \circ g$ is also a function.

Indeed, if we have

$$x \; f \circ g \; y \text{ and } x \; f \circ g \; y'$$

then

$$\text{for some } z, \; x \; f \; z \; g \; y \tag{2}$$

and

$$\text{for some } w, \; x \; f \; w \; g \; y' \tag{3}$$

As f is a function, (2) and (3) give $z = w$. In turn, this (g is a function too) gives $y = y'$. □

The notation (as in 3.4.2) "$(a)\, f$" for relations is awkward when applied to functions in the presence of composition. In something like

$$x \to \boxed{f} \to z \to \boxed{g} \to y$$

that represents (1) above, note that f *acts first*. Its result or output $z = f(x)$ is then inputed to g —that is, we perform

$$g(z) = g\Big(f(x)\Big)$$

to obtain output y. Thus the *first acting* function f is *called first* with argument x and *after that* g is *called* with argument $f(x)$. "Everyday math" notation places the two calls as in

the displayed formua above: The first call to the right of the 2nd call —order reversal vis a vis relational notation!

So, set theory heeds these observations and defines:

3.5.19 Definition (Composition of Functions; Notation) We just learnt (3.5.18) that the composition of two functions produces a function. The present definition is *about notation only*.

Let $f : A \to B$ and $g : B \to C$ be two functions. The relation $f \circ g : A \to C$, their *relational composition* is given in 3.1.16.

For composition of *functions*, we have the *alternative* —so-called *functional notation for composition*: If f, g are functions then we may use "gf" to stand for "$f \circ g$". *Note the order reversal and the absence of* "$\circ$", the composition symbol.

In particular we write $(gf)(a)$ for $(a)(f \circ g)$ —cf. 3.5.3.

Above we said "alternative", *not* exclusive. For functions we have *two* possible notations for composition: relational *and* functional.

Thus

$$a(gf)y \overset{Def}{\iff} a\ f \circ g\ y \overset{22}{\iff} (\exists z)(af z \wedge z\ g\ y)$$

also

$$a(gf)y \overset{Def}{\iff} a\ f \circ g\ y \overset{Def\ 3.4.2}{\iff} (a)(f \circ g) = \{y\}$$

In particular, we have that $(a)(f \circ g)$ of 3.4.2 is the same as $(gf)(a) = g\Big(f(a)\Big)$ as seen through the "computation"

$$\begin{aligned}(a)(f \circ g) &\overset{3.5.18}{=} \{y\} \iff \text{for some } z,\ a\ f\ z \wedge z\ g\ y \\ &\iff^{3.5.3} \text{for some } z,\ f(a) = z \wedge g(z) = y \\ &\iff^{\text{subst. } z \text{ by } f(a)} g\Big(f(a)\Big) = y \end{aligned} \tag{1}$$

Conclusion:

$$(gf)(a) \overset{3.5.19\ via\ 3.5.3}{=} (a)(f \circ g) \overset{(1)}{=} g\Big(f(x)\Big)$$

Thus the "reversal" $gf = f \circ g$ now makes sense! So does $(gf)(a) = g\big(f(a)\big)$. □

3.5.20 Theorem *Functional composition is associative, that is, $(gf)h = g(fh)$.*

Proof Exercise!

---

[22] "$\iff$" is an equivalence that is different from "$\equiv$". When I write $A \iff B \iff C \iff D$ it means $(A \equiv B) \wedge (B \equiv C) \wedge (C \equiv D)$. Hmm. So what if I wrote the immediately previous "lazily" as $A \equiv B \equiv C \equiv D$? Well, then it means something else: $A \equiv \big(B \equiv (C \equiv D)\big)$, which is evaluated as indicated from right to left. We say that "$\equiv$" is *associative* while $\iff$ is *conjunctional*. More in our chapter on Logic!

Hint. Note that by, 3.5.19, $(gf)h = h \circ (f \circ g)$. Take it from here. □

3.5.21 Example The *identity relation* on a set A is a function since $(a)\mathbf{1}_A$ is the singleton $\{a\}$. □

The following interesting result connects the notions of ontoness and 1-1ness with the "algebra" of composition.

3.5.22 Theorem *Let $f : A \to B$ and $g : B \to A$ be functions. If*

$$gf = \mathbf{1}_A \tag{1}$$

then g is onto *while f is* total and 1-1.

We say that g is a left inverse of f and f is a right inverse of g. "*A*" because these are not in general unique! Stay tuned on this!

Proof **About g:** Our goal, ontoness, means that, for each $x \in A$, I can *"solve the equation $g(y) = x$ for y"*. Indeed I can: By definition of $\mathbf{1}_A$,

$$g\Big(f(x)\Big) \overset{3.5.19}{=} (gf)(x) \overset{(1)}{=} \mathbf{1}_A(x) = x$$

So to solve, take $y = f(x)$.

About f: As seen above, $x = g(f(x))$, *for each* $x \in A$. Since this is the same as "$x\ f \circ g\ x$ is true", there must be a z such that $x\ f\ z$ and $z\ g\ x$. The first of these implies $f(x)\downarrow$. This, along with "for all $x \in A$", settles totalness.

For the 1-1ness, let $f(a) = f(b)$. Applying g to both sides we get $g(f(a)) = g(f(b))$. But this says $a = b$, by $gf = \mathbf{1}_A$, and we are done.

Pause. Should we not "translate" 1-1ness above (per 3.5.13) starting with "let $f(a) = f(b) = c$, for some c, then etc."? ◄ □

3.5.23 Example *The above is as much as can be proved.* For example, say $A = \{1, 2\}$ and $B = \{3, 4, 5, 6\}$. Let $f : A \to B$ be $\{(1, 4), (2, 3)\}$ and $g : B \to A$ be $\{(4, 1), (3, 2), (6, 1)\}$, or in friendlier notation

$f(1) = 4$
$f(2) = 3$
and
$g(3) = 2$
$g(4) = 1$
$g(5)\uparrow$

$g(6) = 1$

Clearly, $gf = \mathbf{1}_A$ holds, but note:
(1) f is not onto.
(2) g is neither 1-1 nor total. □

3.5.24 Example With $A = \{1, 2\}$, $B = \{3, 4, 5, 6\}$ and $f : A \to B$ and $g : B \to A$ as in the previous example, consider also the functions $\tilde{f}$ and $\tilde{g}$ given by

$\tilde{f}(1) = 6$
$\tilde{f}(2) = 3$
and
$\tilde{g}(3) = 2$
$\tilde{g}(4) = 1$
$\tilde{g}(5) = 1$
$\tilde{g}(6) = 1$

Clearly, $\tilde{g}f = \mathbf{1}_A$ and $g\tilde{f} = \mathbf{1}_A$ hold, but note:
(1) $f \neq \tilde{f}$.
(2) $g \neq \tilde{g}$.
Thus, neither left nor right inverses need be unique. The article "a" in the definition of said inverses was well-chosen. □

The following two partial converses of 3.5.22 are useful.

3.5.25 Theorem *Let $f : A \to B$ be total and 1-1. Then there is an* onto $g : B \to A$ *such that $gf = \mathbf{1}_A$.*

Proof Consider the *converse* (also called "inverse") relation (3.4.1) of f —that is, the *relation* f^{-1}— and call it g:

$$x\ g\ y \overset{\text{Def}}{\text{ iff }} y\ f\ x \tag{1}$$

By Exercise 3.5.15, $g : B \to A$ is a (possibly nontotal) function so we can write (1) as $g(x) = y$ iff $f(y) = x$, from which, substituting $f(y)$ for x in $g(x)$ we get $g(f(y)) = y$, *for all* $y \in A$, that is $gf = \mathbf{1}_A$, hence g is onto by 3.5.22. We got both statements that we needed to prove. □

Pause. Why "for all $y \in A$"? ◄

3.5.26 Remark By (1) above, $\text{dom}(g) = \{x : (\exists y)g(x) = y\} = \{x : (\exists y)f(y) = x\} = \text{ran}(f)$. □

3.5.27 Theorem *Let $f : A \to B$ be onto. Then there is a* total and 1-1 *$g : B \to A$ such that $fg = \mathbf{1}_B$.*

Proof By assumption, $\emptyset \neq f^{-1}[\{b\}] \subseteq A$, for all $b \in B$. To define $g(b)$ choose *one* $c \in f^{-1}[\{b\}]$ and set $g(b) = c$. Since $f(c) = b$, we get $f(g(b)) = b$ for all $b \in B$, and hence g is 1-1 and total by 3.5.22. □

3.5.28 Remark (The Axiom of Choice) Strictly speaking, the argument in 3.5.27 is flawed. "Choose *one* $c \in f^{-1}[\{b\}]$, for each b"? How?

Well, "for each b, there *is* at least *one* $c \in f^{-1}[\{b\}]$.[23] For each b, write one such down!"

Hmm! If that process were *finite* I'd be willing to go along, and say, "oh well! A proof is a finite sequence of statements. Like the one above, in quotes. So I accept it".

But then I thought: "wait a minute!" If B is infinite, intuitively speaking, then this "proof" never ends! *But there is no such thing as a never-ending proof!*

Is there a way out of this difficulty? **Answer**: Only if we could *somehow describe these infinitely many choices in a finite proof*!

For example, if all the sets $f^{-1}[\{b\}]$ are subsets of $\mathbb{N}$, and since they are nonempty, we could say "in each $f^{-1}[\{b\}]$ pick the *smallest* natural number therein!" This simple phrase *well-defines* exactly what to do and how to effect each choice, and describes so in a *finite way*, avoiding an "infinite proof".

Russell once illustrated this problem by contrasting choosing *one sock* from *each pair* of an *infinite set of pairs* of socks, on one hand, while, on the other hand, choosing *one shoe* from from *each pair* of a similarly infinite *set of pairs of shoes*.

For shoes the method is simply described: "pick the left shoe in each pair". This totally defines *in a finite manner* how *each* of the infinitely many choices *can be effected; precisely*.

In the absence of a method (back then[24]) of identifying a left from a right sock there is no obvious finite method to avoid the obvious infinite "proof/construction": "Pick a sock pair; pick a sock from it. Pick another sock pair; pick a sock from it. Etc., forever!" Set theorists (reluctantly at first! for a discussion see Wilder (1963)) adopted an axiom —called the *Axiom of Choice*— that postulates

> If S is an infinite family of *nonempty sets*, then there exists a "*choice function*" C that "chooses" one element from each set $A \in S$, in the precise technical sense: $C(A) \in A$, for all $A \in S$.

This axiom is applicable to our $\{f^{-1}[\{b\}] : b \in B\}$ since all $f^{-1}[\{b\}]$ are nonempty by ontoness of f. Thus the "construction" of $g(b)$ in the proof above —"To define $g(b)$ choose *one* $c \in f^{-1}[\{b\}]$ and set $g(b) = c$"— is legitimised by the Axiom of Choice by

[23] Since f is onto.

[24] The illustration with the socks would not be valid with certain branded socks nowadays that have the brand insignia so positioned as to identify left and right socks in a pair.

saying, "let C be a choice function for the family $\{f^{-1}[\{b\}] : b \in B\}$. For each b, define $g(b) \stackrel{Def}{=} C\Big(f^{-1}[\{b\}]\Big)$". □

3.6 Finite and Infinite Sets

Broadly speaking (that is, with very little detail contained in what I will say next) we have sets that are *finite* —intuitively meaning that we can count *all* their elements in a finite amount of time (but see the -remark 3.6.3 below)— and those that are not, naturally called *infinite*!

What is a mathematical way to say all this?

Any counting process of the elements of a finite set A will have us say out loud —every time we pick or point at an element of A— "0th", "1st", "2nd", etc., and, once we reach and pick the last element of the set, we finally pronounce "nth", for some appropriate n that we reached in our counting (again, see 3.6.3).

Thus, mathematically, we are pairing each member of the set with a member from $\{0, \ldots, n\}$.

So we propose,

3.6.1 Definition (Finite and infinite sets) A set A is *finite* iff it is either empty, or is in 1-1 correspondence with $\{x \in \mathbb{N} : x \leq n\}$. This "normalised" small set of natural numbers we usually denote by $\{0, 1, 2, \ldots, n\}$.

If a set is *not* finite, then it is, *by definition*, *infinite*. □

3.6.2 Example For any n, $\{0, \ldots, n\}$ is finite since, trivially, $\{0, \ldots, n\} \sim \{0, \ldots, n\}$ using the identity (Δ) function on the set $\{0, \ldots, n\}$. □

3.6.3 Remark One must be careful when one attempts to explain finiteness via counting by a human.

For example, Achilles[25] could count *infinitely many objects* by constantly accelerating his counting process as follows:

He procrastinated for a *full second*, and then counted the first element. Then, he counted the second object *exactly after* 1/2 a second from the first. Then he got to the third element $1/2^2$ seconds after the previous, …, he counted the n th item at exactly $1/2^{n-1}$ seconds after the previous, and so on *forever*.

Hmm! It was *not* "forever", *was it*? After a total of 2 seconds he was done!

[25] OK, he was a demigod; but only "demi".

You see (as you can easily verify from your calculus knowledge (limits)),[26]

$$1 + \frac{1}{2} + \frac{1}{2^2} + \ldots + \frac{1}{2^{n-1}} + \ldots = \frac{1}{1 - 1/2} = 2$$

So "time" is not a good determinant of finiteness! □

3.6.4 Theorem *If $X \subsetneq \{0, \ldots, n\}$, then there is no* onto *function $f : X \to \{0, \ldots, n\}$.*

I am saying, no such f —whether total or not— exists; *totalness is immaterial.*

Proof First off, the claim holds if $X = \emptyset$, since then any such f equals $\emptyset$ and its range is empty.

Let us otherwise proceed by way of contradiction, and assume that the theorem is *wrong*: That is, *assume that* it *is* possible to have such onto functions, for some n and well chosen X.

Since I assume there are such $n > 0$ values, suppose then that the *smallest* n that allows this to happen is, say, n_0, and let X_0 be a *corresponding* set "X" that works, that is,

$$\text{Assume that we have an onto } f : X_0 \to \{0, \ldots, n_0\} \tag{1}$$

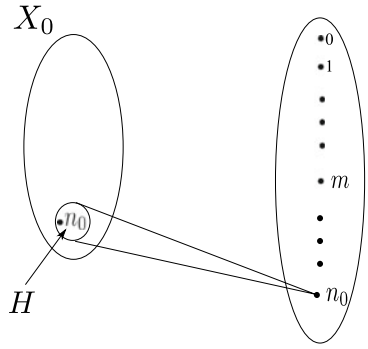
Thus $X_0 \neq \emptyset$, by the preceding remark, and therefore $n_0 > 0$, since otherwise $X_0 = \emptyset$.

Pause. Why "otherwise $X_0 = \emptyset$"? ◀

Let us call H be the set of all x such that $f(x) = n_0$, in short, $H = f^{-1}[\{n_0\}]$. We have $\emptyset \neq H \subseteq X_0$; the $\neq$ by ontoness.

Case 1. $n_0 \in H$. Then removing all pairs (a, n_0) from f —all these have $a \in H$— we get a new function $f' : X_0 - H \to \{0, 1, \ldots, n_0 - 1\}$, which *is still onto* as we only removed inputs that cause output n_0. Moreover, $X_0 - H \subsetneq \{0, 1, \ldots, n_0 - 1\}$. (Why?)

This contradicts minimality of n_0 since $n_0 - 1$ works too!



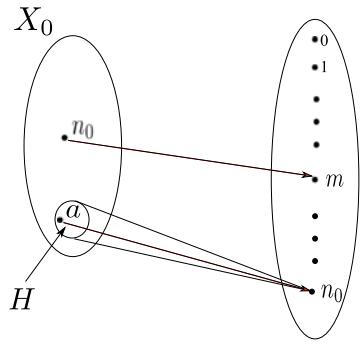
[26] $1 + \frac{1}{2} + \frac{1}{2^2} + \ldots + \frac{1}{2^{n-1}} = \frac{1-1/2^n}{1-1/2}$. Now let n go to infinity at the limit.

Case 2. $n_0 \notin H$.

If $n_0 \notin X_0$, then we argue exactly as in Case 1 and we just remove the base "H" of the cone (in the picture) from X_0.

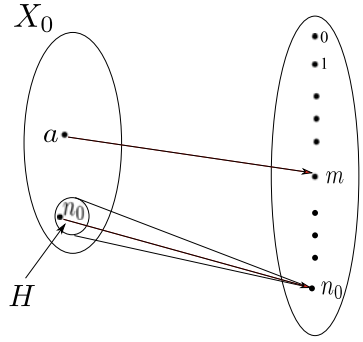
Otherwise, we have two subcases:

- $f(n_0) \uparrow$. Then we (almost) act as in Case 1: The new "X_0" is $(X_0 - H) - \{n_0\}$, since if we leave n_0 in, then the new "X_0" will not be a subset of $\{0, 1, \ldots, n_0 - 1\}$. We get a contradiction per Case 1.
- The picture below —that is, $f(n_0) = m$, for some $m \neq n_0$.



We simply transform the picture to the one below, "correcting" f to have $f(a) = m$ and $f(n_0) = n_0$, that is *defining a new "f"* that we will call f' by

$$f' = \Big(f - \{(n_0, m), (a, n_0)\}\Big) \cup \{(n_0, n_0), (a, m)\}$$



We get a contradiction per Case 1. □

3.6.5 Corollary (Pigeon-Hole Principle) *If $m < n$, then $\{0, \ldots, m\} \nsim \{0, \ldots, n\}$.*

Proof If the conclusion fails then we have an onto $f : \{0, \ldots, m\} \to \{0, \ldots, n\}$, contradicting 3.6.4. □

Important!

3.6.6 Theorem *If A is finite due to $A \sim \{0, 1, 2, \ldots n\}$ then there* can be no justification of finiteness via another canonical set $\{0, 1, 2, \ldots m\}$ *with* $n \neq m$.

Proof If $\{0, 1, 2, \ldots n\} \sim A \sim \{0, 1, 2, \ldots m\}$, then $\{0, 1, 2, \ldots n\} \sim \{0, 1, 2, \ldots m\}$ by 3.5.17, hence $n = m$, otherwise we contradict 3.6.5. □

3.6.7 Definition Let $A \sim \{0, \ldots, n\}$. Since n is uniquely determined by A we say that A has $n + 1$ elements and write $|A| = n + 1$. □

3.6.8 Corollary *For any choice of n, there is no onto function from* $\{0, \ldots, n\}$ *to* $\mathbb{N}$.

Proof Fix an n. By way of contradiction, let $g : \{0, \ldots, n\} \to \mathbb{N}$ be onto. Let

$$Y \stackrel{Def}{=} \{x \leq n : g(x) > n + 1\}$$

Now let

$$X \stackrel{Def}{=} \{0, \ldots, n\} - Y$$

and

$$g' \stackrel{Def}{=} g - Y \times \mathbb{N}$$

The "$g - Y \times \mathbb{N}$" above is an easy way to say "remove all pairs from g that have their first component in Y".

Thus, $g' : X \to \{0, \ldots, n, n + 1\}$ is onto (Why?), contradicting 3.6.4 because $X \subseteq \{0, \ldots, n\} \subsetneq \{0, \ldots, n, n + 1\}$. □

3.6.9 Corollary $\mathbb{N}$ *is infinite.*

Proof By 3.6.1 the opposite case requires that there is an n and a function $f : \{0, 1, 2, \ldots, n\} \to \mathbb{N}$ that is a 1-1 correspondence. Impossible, since any such an f will fail to be *onto*. □

Our mathematical definitions have led to what we hoped they would: That $\mathbb{N}$ *is* infinite as we intuitively understand, notwithstanding Achilles's accelerated counting!

$\mathbb{N}$ is a "canonical" infinite set that we can use to index the members of many infinite sets. Sets that can be indexed using natural number indices

$$a_0, a_1, \ldots$$

are called *countable*.

In the interest of technical flexibility, *we do not insist* that *all* members of $\mathbb{N}$ be used as indices. We might enumerate with gaps:

$$b_5, b_9, b_{13}, b_{42}, \ldots$$

Thus, informally, a set A is *countable* if it is empty or (in the opposite case) if there is a way to index, hence enumerate, all its members in an array, utilising indices from $\mathbb{N}$. Cf. 3.3.6.

It *is* allowed to repeatedly list any element of A, so that finite sets are countable. For example, the set $\{42\}$:

$$42, 42, 42, \overbrace{\ldots}^{\text{42 forever}}$$

We may think that the enumeration above is done by assigning to "42" *all* of the members of $\mathbb{N}$ as indices, in other words, the enumeration is effected, for example, by the constant function $f : \mathbb{N} \to \{42\}$ given by $f(n) = 42$ for all $n \in \mathbb{N}$. This is consistent with our earlier definition of indexing (3.3.6).

Now, mathematically,

3.6.10 Definition (Countable Sets) We call a set A *countable* if $A = \emptyset$, or there is an *onto* function $f : \mathbb{N} \to A$. We do *not* require f to be total. This means that some or many indices from $\mathbb{N}$ need not be used in the enumeration.

If $f(n) \downarrow$, then we say that $f(n)$ is the nth element of A in the enumeration f. We often write f_n instead of $f(n)$ and then call n a "subscript" or "index". □

Thus a nonempty set is countable iff it is the *range* of some function that has $\mathbb{N}$ as its *left field*.

Incidentally, since we allow f to be non total, the hedging "nonempty" (in 3.6.10 above and in this remark) is unnecessary: $\emptyset$ is the range of the empty function that has $\mathbb{N}$ as its left field.

We said that the f that proves countability of a set A need not be total. But such an f can always be "completed", by adding pairs to it, to get an f' such that $f' : \mathbb{N} \to A$ is onto *and* total. Here is how:

3.6.11 Proposition *Let $f : \mathbb{N} \to A \neq \emptyset$[27] be onto. Then we can extend f to f' so that $f' : \mathbb{N} \to A$ is onto* and *total.*

[27] Since we are constructing a *total* onto function to A we need to assume the case $A \neq \emptyset$ as we cannot put any outputs into $\emptyset$.

Proof Pick an $a \in A$ —possible since $A \neq \emptyset$— and keep it fixed. Now, our sought f' is given for all $n \in \mathbb{N}$ by cases as below:

$$f'(n) = \begin{cases} f(n) & \text{if } f(n) \downarrow \\ a & \text{if } f(n) \uparrow \end{cases}$$

□

Some set theorists also define sets that can be enumerated using *all* the elements of $\mathbb{N}$ as indices *without repetitions*.

3.6.12 Definition (Enumerable or denumerable sets) A set A is *enumerable* or *denumerable*[28] iff $A \sim \mathbb{N}$. □

3.6.13 Example Every enumerable set is countable, but the converse fails. For example, $\{1\}$ is countable but not enumerable due to 3.6.8. $\{2n : n \in \mathbb{N}\}$ is enumerable, with $f(n) = 2n$ effecting the 1-1 correspondence $f : \mathbb{N} \to \{2n : n \in \mathbb{N}\}$. □

3.6.14 Theorem *If A is an infinite subset of $\mathbb{N}$, then $A \sim \mathbb{N}$.*

Proof We will build a 1-1 and total enumeration of A, presented in a finite manner as a (pseudo) program below, which enumerates all the members of A in strict ascending order and arranges them in an array

$$a(0), a(1), a(2), \ldots \tag{1}$$

n	$\leftarrow 0$
while	$A \neq \emptyset$
$a(n)$	$\leftarrow \min A$ **Comment**.We are inside the loop $\emptyset \neq A \subseteq \mathbb{N}$, hence min exists.
A	$\leftarrow A - \{a(n)\}$
n	$\leftarrow n + 1$
end while	

Note that the sequence $\{a(0), a(1), \ldots, a(m)\}$ is *strictly increasing* for any m, since $a(0)$ is smallest in A, $a(1)$ is smallest in $A - \{a(0)\}$ and hence *the next higher than* $a(0)$ in A, etc.

Will this loop ever exit? *Suppose that it exits* when it starts (but does not complete) the k-th pass through the loop. Thus A became empty when we did $A \leftarrow A - \{a(k-1)\}$ in the previous pass, that is $A = \{a(0), a(1), \ldots, a(k-1)\}$ and thus, since

$$a(0) < a(1) < \ldots < a(k-1)$$

we have that the function $f : \{0, \ldots, k-1\} \to A$ given by

[28] We will not use this term in this work.

$$f = \{(0, a(0)), (1, a(1)), \ldots (k-1, a(k-1))\}$$

is total, 1-1 and onto, hence $A \sim \{0, \ldots, k-1\}$ *contradicting that A is infinite*!

Thus, we never exit the loop!

Hence, by the remark in the paragraph above, (1) enumerates A in strict ascending order, that is,

$$\text{if we define } f : \mathbb{N} \to A \text{ by } f(n) = a(n), \text{ for all } n$$

then f is 1-1 (by strict increasing property: distinct inputs cause distinct outputs), and is trivially total, and onto. Why the latter? Every $a \in A$ is reached in ascending order, and assigned an "n" from $\mathbb{N}$. □

3.6.15 Theorem *Every infinite countable set is enumerable.*

Proof Let $f : \mathbb{N} \to A$ be onto and total (cf. 3.6.11), where A is infinite. Let $g : A \to \mathbb{N}$ such that $fg = \mathbf{1}_A$ (3.5.27). Thus, if we let $B = \text{ran}(g)$, we have that g is onto B, and by 3.5.22 is also 1-1 and total. Thus it is a 1-1 correspondence $g : A \to B$, that is,

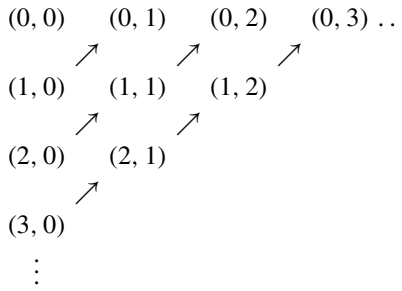
$$A \sim B \tag{1}$$

B *must* be infinite, otherwise (3.6.1), for some n, $B \sim \{0, \ldots, n\}$ and by (1) via Exercise 3.5.17 we have $A \sim \{0, \ldots, n\}$, contradicting that A is infinite. Thus, by 3.6.14, $B \sim \mathbb{N}$, hence (again, Exercise 3.5.17 and (1)) $A \sim \mathbb{N}$. That is, A is enumerable. □

So, if we can enumerate an infinite set at all, then we can enumerate it without repetitions.

We can linearise an infinite square matrix of elements in each location (i, j) by devising a traversal that will go through each (i, j) entry *once*, and will *not miss any entry*!

In the literature one often sees the method diagrammatically, see below, where arrows *clearly* indicate the sequence of traversing, with the understanding that we use the arrows by picking the first unused chain of arrows from left to right.



So the linearisation induces a 1-1 correspondence between $\mathbb{N}$ *and the linearised sequence of matrix entries, that is, it shows that* $\mathbb{N} \times \mathbb{N} \sim \mathbb{N}$. In short,

3.6.16 Theorem *The set* $\mathbb{N} \times \mathbb{N}$ *is countable. In fact, it is enumerable.*

Is there a "mathematical" way to do this? Well, the above *is* mathematical, don't get me wrong, but is given in *outline*. It is like an argument in geometry, where we rely on drawings (figures).

Here are the algebraic details:

Proof (of 3.6.16 with an algebraic argument). Let us call $i + j + 1$ the "*weight*" of a pair (i, j). The weight is the number of elements in the group:

$$(i + j, 0), (i + j - 1, 1), (i + j - 2, 2), \ldots, (i, j), \ldots, (0, i + j)$$

Thus the diagrammatic enumeration proceeds by enumerating *groups* by increasing weight

$$1, 2, 3, 4, 5, \ldots$$

and in each group of weight k we enumerate in *ascending order of the second component*.

Thus the (i, j) th entry occupies position j *in its group* —the first position in the group being the 0 th, e.g., in the group of $(3, 0)$ the first position is the 0 th— and this position *globally* is the number of elements in all groups *before* group $i + j + 1$, *plus* j. Thus the first available position for the first entry of group (i, j) members is just after this many occupied positions:

$$1 + 2 + 3 + \ldots (i + j) = \frac{(i + j)(i + j + 1)}{2}$$

That is,

$$\text{global position of } (i, j) \text{ is this: } \frac{(i + j)(i + j + 1)}{2} + j$$

The function f which for all i, j is given by

$$f(i, j) = \frac{(i + j)(i + j + 1)}{2} + j$$

is the algebraic form of the above enumeration. □

There is an easier way to show that $\mathbb{N} \times \mathbb{N} \sim \mathbb{N}$ without diagrams:

By the unique factorisation of numbers into products of primes (Euclid) the function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ given for all m, n by $g(m, n) = 2^m 3^n$ is 1-1, since Euclid proved that $2^m 3^n = 2^{m'} 3^{n'}$ implies $m = m'$ and $n = n'$. It is not onto as it never outputs, say, 5, but $\text{ran}(g)$ is an *infinite* subset of $\mathbb{N}$ (Exercise!).

Thus, trivially, $\mathbb{N} \times \mathbb{N} \sim \text{ran}(g) \sim \mathbb{N}$, the latter "$\sim$" by 3.6.14.

3.6.17 Exercise If A and B are enumerable, so is $A \times B$.
Hint. So, $\mathbb{N} \sim A$ and $\mathbb{N} \sim B$. Can you show now that $\mathbb{N} \times \mathbb{N} \sim A \times B$?
With little additional effort one can generalise to the case of $\overset{n}{\underset{i=1}{\times}} A_i$. □

3.6.18 Remark

1. Let us collect a few more remarks on countable sets here. Suppose now that we start with a countable set A. Is every subset of A countable? Yes, because the composition of onto functions is onto.
2. **3.6.19 Exercise** What does composition of onto functions have to do with this? Well, if $B \subseteq A$ then there is a *natural* onto function $g : A \to B$. Which one? Think "natural"! Get a *natural* total and 1-1 function $f : B \to A$ and then use f to get g. □
3. As a special case, if A is countable, then so is $A \cap B$ for any B, since $A \cap B \subseteq A$.
4. How about $A \cup B$? If both A and B are countable, then so is $A \cup B$. Indeed, and without inventing a new technique, let

$$a_0, a_1, \ldots$$

be an enumeration of A and

$$b_0, b_1, \ldots$$

for B. Now form an infinite matrix with the A-enumeration as the 1st row, while each remaining row is the same as the B-enumeration. Now linearise this matrix!
Of course, we may alternatively adapt the unfolding technique to an infinite matrix of just two rows. How?

5. **3.6.20 Exercise** Let A be enumerable and an enumeration of A

$$a_0, a_1, a_2, \ldots \tag{1}$$

is given.
So, this is an enumeration without repetitions.
Use techniques we employed in this section to propose a new enumeration in which *every* a_i is listed *infinitely many times* (this is useful in some applications of logic). □

3.7 Diagonalisation and Uncountable Sets

3.7.1 Example Suppose we have a 3×3 matrix

$$\begin{matrix} 1\ 1\ 0 \\ 1\ 0\ 1 \\ 0\ 1\ 1 \end{matrix}$$

and we are asked: Find a sequence of three numbers, *using only* 0 *or* 1, that does not *fit* as a row of the above matrix —i.e., is *different from all rows*.

Sure, you reply: Take 1 1 1. Or, take 0 0 0.

That is correct. But what if the matrix were big, say, $10^{350000} \times 10^{350000}$, or even *infinite*?

Is there a *finitely describable technique* that can produce an "unfit" row for any square matrix, even an infinite one? Yes, it is Cantor's *diagonal method* or technique.

He noticed that any row that fits in the matrix as the, say, i-th row, intersects the main diagonal at the same spot that the i-th column does.

That is, at entry (i, i).

Thus if we take the main diagonal —a sequence that has the same length as any row— and *change every one of its entries*, then it will not fit *anywhere* as a row! *Because no row can have an entry that is different from the entry at the location where it intersects the main diagonal!*

This idea would give the answer 0 1 0 to our original question. While 1000 11 3 also follows the principle "change all the entries of the diagonal" and works, we are constrained here to "use only 0 or 1" as entries. More seriously, in a case of a very large or infinite matrix it is best to have a simple technique that *is finitely describable* and works even if we do not know much about the elements of the matrix. Read on! □

3.7.2 Example We have an infinite matrix of 0-1 entries. Can we produce an infinite sequence of 0-1 entries that does not match *any* row in the matrix? Yes, take the main diagonal and *flip every entry* (0 to 1; 1 to 0).

If we assume that it fits as row i, then we get a contradiction:

Say the *original* row has an a as entry (i, i). But, by our construction, the *new* row has an $1 - a$ in as entry (i, i), so it will not fit as row i after all. So it fits nowhere, i being arbitrary. □

The technique of obtaining a modified copy of the main diagonal of an infinite matrix so that it does not match any row of the matrix is due to Cantor and is called *diagonal method*, or *diagonalisation*.

3.7.3 Example (Cantor) Let S denote the set of all infinite sequences of 0s and 1s.

Pause. What is an *infinite sequence*? Our intuitive understanding of the term is captured mathematically by the concept of a total function f with left field (and hence domain) $\mathbb{N}$. The n-th member of the sequence is $f(n)$.◄

Can we arrange *all* of S in an infinite matrix —one element per row? No, since the preceding example shows that we would miss at least one infinite sequence (i.e., we would fail to list it as a row), because a sequence of infinitely many 0s and/or 1s can be found, as indicated above, that does not match any row!

But arranging all members of S as an infinite matrix —one element per row— is tantamount to saying that we can enumerate all the members of S using members of $\mathbb{N}$ as indices.

So we cannot do that. S is not countable! □

3.7.4 Definition (Uncountable Sets) A set that is *not* countable is called *uncountable*. □

So, an uncountable set is neither finite, nor enumerable. The first observation makes it infinite, the second makes it "more infinite" than the set of natural numbers since it is not in 1-1 correspondence with $\mathbb{N}$ (else it would be enumerable, hence countable) nor with a *subset* of $\mathbb{N}$: indeed, if the latter holds, then our uncountable set would be finite or enumerable (which is absurd) according as it would be in 1-1 correspondence with a finite subset or an infinite subset of $\mathbb{N}$ (cf. 3.6.14 and Exercise 3.5.17).

Example 3.7.3 shows that uncountable sets exist. Here is a more interesting one.

3.7.5 Example (Cantor) The set of real numbers in the interval

$$(0,1) \stackrel{\text{Def}}{=} \{x \in \mathbb{R} : 0 < x < 1\}$$

is uncountable. This is done via an elaboration of the argument in 3.7.3.

Think of a member of $(0, 1)$, *in form*, as an infinite sequence of numbers from the set

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

prefixed with a dot; that is, think of the number's decimal notation.

Some numbers have representations that end in 0s after a certain point. We call these representations *finite*. Every such number has also an "infinite representation" since the non zero digit d immediately to the left of the infinite tail of 0s can be converted to $d - 1$, and the infinite tail into 9s, without changing the value of the number.

Allow only infinite representations.

Assume now by way of contradiction that a listing of all members of $(0, 1)$ exists, listing them via their infinite representations

$$\begin{array}{l} .a_{00}a_{01}a_{02}a_{03}a_{04}\ldots \\ .a_{10}a_{11}a_{12}a_{13}a_{14}\ldots \\ .a_{20}a_{21}a_{22}a_{23}a_{24}\ldots \\ .a_{30}a_{31}a_{32}a_{33}a_{34}\ldots \\ \vdots \end{array}$$

The argument from 3.7.3 can be easily modified to get a "row that does not fit", that is, a representation

$$.d_0d_1d_2\cdots$$

not in the listing.

Well, just let

$$d_i = \begin{cases} 2 & \text{if } a_{ii} = 0 \vee a_{ii} = 1 \\ 1 & \text{otherwise} \end{cases}$$

Clearly $.d_0d_1d_2\cdots$ does not fit in any row i as it differs from the expected digit at the i-th decimal place: should be a_{ii}, but $d_i \neq a_{ii}$. It is, on the other hand, an infinite decimal expansion, being devoid of zeros, and thus *should* be listed. This contradiction settles the issue. □

3.7.6 Example (3.7.3 Revisited) Consider the set of all *total* functions from $\mathbb{N}$ to $\{0, 1\}$. Is this countable?

Well, if there is an enumeration of these one-variable functions

$$f_0, f_1, f_2, f_3, \ldots \tag{1}$$

consider the function $g : \mathbb{N} \to \{0, 1\}$ given by $g(x) = 1 - f_x(x)$. Clearly, this *must* appear in the listing (1) since it has the correct left and right fields, and is total.

Too bad! If $g = f_i$ then $g(i) = f_i(i)$. By definition, it is also $1 - f_i(i)$. A contradiction.

This is just a version of 3.7.3; as already noted there, an infinite sequence of 0s and 1s is just a total function from $\mathbb{N}$ to $\{0, 1\}$. □

The same argument as above shows that the set of all functions from $\mathbb{N}$ to itself is uncountable. Taking $g(x) = f_x(x) + 1$ also works in this case to "systematically change the diagonal" $f_0(0), f_1(1), \ldots$ since we are not constrained to keep the function values in $\{0, 1\}$.

3.7.7 Remark Worth Emphasizing. Here is how we constructed g: We have a list of *in principle available f-indices* for g. We want to make sure that *none of them applies*.

A convenient method to do that is to inspect each available index, i, and using the diagonal method do this: *Ensure that g differs from f_i at input i*, by setting $g(i) = 1 - f_i(i)$.

This ensures that $g \neq f_i$; period. We say that *we cancelled the index i* as a *possible* "f-index" of g.

Since the process is applied *for each i*, *we have cancelled all possible indices for g*: For no i can we have $g = f_i$. □

3.7.8 Example (Cantor) What about the set of all subsets of $\mathbb{N}$ —$\mathscr{P}(\mathbb{N})$ or $2^{\mathbb{N}}$?

Cantor showed that this is uncountable as well: If not, we have an enumeration of its members as

$$S_0, S_1, S_2, \ldots \tag{1}$$

Define the set

$$D \stackrel{\text{Def}}{=} \{x \in \mathbb{N} : x \notin S_x\} \tag{2}$$

So, $D \subseteq \mathbb{N}$, thus it must appear in the list (1) as an S_i. But then $i \in D$ iff $i \in S_i$ by virtue of $D = S_i$. However, also $i \in D$ iff $i \notin S_i$ by (2). This contradiction establishes that a *legitimate subset of* $\mathbb{N}$, *namely* D, *is not an* S_i. That is, $2^{\mathbb{N}}$ *cannot* be so enumerated; it is uncountable. □

3.7.9 Example (Characteristic functions) Let $S \subseteq \mathbb{N}$. We can represent S as an infinite 0/1 array:

array position	$\ldots$	i	$\ldots$	j $\ldots$
array content	$\ldots$	0	$\ldots$	1 $\ldots$
	$\ldots$	$\uparrow$	$\ldots$	$\uparrow$ $\ldots$
means		$\ldots i \notin S \ldots$		$j \in S \ldots$

This array faithfully represents S —tells all we need to know about what S contains— since it contains a "1" in location x iff $x \in S$; contains "0" otherwise.

The array viewed as a total function from $\mathbb{N}$ to $\{0, 1\}$ is called the *characteristic function of* S, denoted by c_S:

$$c_S(x) = \begin{cases} 1 & \textbf{if } x \in S \\ 0 & \textbf{if } x \in \mathbb{N} - S \end{cases}$$

Note that there is a 1-1 correspondence, let's call it F, between subsets of $\mathbb{N}$ and the total 0-1-valued functions from $\mathbb{N}$ simply given by $F(S) = c_S$. (Exercise!)

Thus

$$\{f : f : \mathbb{N} \to \{0, 1\} \text{ and } f \text{ is total}\} \sim 2^{\mathbb{N}}$$

In particular, the concept of characteristic functions shows that Example 3.7.8 fits the diagonalization methodology. Indeed, the argument in 3.7.8 sets $c_D(x) = 1 - c_{S_x}(x)$, for all x, because

$$c_D(x) = 1 \text{ iff } x \in D \text{ iff } x \notin S_x \text{ iff } c_{S_x}(x) = 0 \text{ iff } 1 - c_{S_x}(x) = 1$$

But then, the argument in 3.7.8 essentially applies the diagonal method to the list of 0/1 functions c_{S_x}, for $x = 0, 1, 2, \ldots$, to show that some 0/1 function, namely, c_D cannot be in the list. □

3.7.10 Remark (Cantor) Cantor offered also a generalisation of 3.7.8: For any set X, we have $X \nsim 2^X = \mathscr{P}(X)$.

Assume otherwise, and let $f : X \to 2^X$ be onto, and let us write "W_x" for "$f(x)$" (the latter, hence also the former) being the subset of X enumerated by f at "position" x ($\in X$).

Define

$$D \stackrel{Def}{=} \{x \in X : x \notin W_x\} \tag{1}$$

We show that D ($\subseteq X$) is *not* a "W_x", that is, it is not enumerated by f contradicting ontoness of the latter. Thus, indeed $X \nsim 2^X$.

The details: Suppose $D = W_a$ for some $a \in X$. Then $a \in D \equiv a \in W_a$. But also (by (1)), $a \in D \equiv a \notin W_a$. A contradiction.

Is this diagonalisation? Of course: Let c_D and c_{W_x} be the characteristic functions of D and W_x (any x) respectively.

We have arranged so that (by (1)) $c_D(x) = 1$ iff $c_{W_x}(x) = 0$. So from the infinite 0/1 matrix with entries $c_{W_i}(j)$ we obtained D by flipping the main diagonal entries $c_{W_i}(i)$. □

3.7.11 Remark (Russell Paradox and Diagonalisation) It should be mentioned that the argument in Russell's paradox is a diagonalisation in the model of the above.

In 3.7.10 we show that $\{x : x \notin W_x\}$ is "*not a* W_x".

The W_x above are enumerated using indices x from X.

Well, consider here an (*attempted*) enumeration of *all* sets using as indices the sets themselves —that is, the enumerating function is the identity —$\lambda x.x : \mathbb{U} \to \mathbb{U}$— so while we might imagine that a set a is enumerated as a "W_a" we actually enumerate it as just "a" without the unnecessary burden of the "W-notation".

As in 3.7.10, we consider the question: Have we enumerated *all* sets? Or, for example,

Is

$$\{x : x \notin \overbrace{x}^{\text{simplified "}W_x\text{" notation}}\}$$

a "W_a" —or, in our simplified notation, an "a"?

Well, if yes then $a = \{x : x \notin x\}$, hence $a \in a \equiv a \notin a$, a contradiction.

It is somewhat ironic that Cantor's famous tool of *diagonalisation* was used to find a contradiction in his *set theory*. □

3.8 Operators and the Cantor-Bernstein Theorem

3.8.1 Definition (Operators) An *operator* Γ on a *set* X is a *total* function $\Gamma : \mathscr{P}(X) \to \mathscr{P}(X)$.

It is called *monotone* iff $S \subseteq T \subseteq X$ implies $\Gamma(S) \subseteq \Gamma(T)$. A set $Z \subseteq X$ is Γ-*closed* means that $\Gamma(Z) \subseteq Z$.

The most popular general symbols that name arbitrary operators in the literature are Γ, Φ, Ψ. □

Note that Γ acts on points in $\mathscr{P}(X)$ so the notation "$\Gamma(S)$" (as opposed to "$\Gamma[S]$") is correct.

3.8.2 Example Let $\Gamma : \mathscr{P}(X) \to \mathscr{P}(X)$ be as in 3.8.1. Then $\Gamma(X) \subseteq X$. Indeed, by definition, $\Gamma(X) \in \mathscr{P}(X)$ therefore $\Gamma(X) \subseteq X$. □

3.8.3 Theorem (Fixpoint Theorem) *Given a* monotone *operator* $\Gamma : 2^X \to 2^X$. *It provably has a* ($\subseteq$-)least fixpoint S, *that is,*

- $\Gamma(S) = S$ *and*
- *If* $\Gamma(X) = X$, *then* $S \subseteq X$.

Proof Consider the family of sets

$$\mathbb{F} = \Big\{ Z : Z \subseteq X \wedge \Gamma(Z) \subseteq Z \Big\}$$

By 3.8.2 and the trivial $X \subseteq X$ we have $X \in \mathbb{F}$ so this class is nonempty and it has, therefore, an intersection S that is a sub*set* of X:

$$S = \bigcap \mathbb{F} = \bigcap_{Z \in \mathbb{F}} Z \subseteq X \tag{1}$$

For $Z \in \mathbb{F}$, it is $\bigcap_{T \in \mathbb{F}} T \subseteq Z$, thus we have $\Gamma\left(\bigcap_{T \in \mathbb{F}} T\right) \subseteq \Gamma(Z)$ by monotonicity and hence also

$$\Gamma\left(\bigcap_{T \in \mathbb{F}} T\right) \subseteq \bigcap_{Z \in \mathbb{F}} \Gamma(Z) \subseteq \bigcap_{Z \in \mathbb{F}} Z = S \tag{2}$$

where the rightmost "$\subseteq$" is due to the condition "$\Gamma(Z) \subseteq Z$" in $\mathbb{F}$.

We have shown (in (2)) that

$$\Gamma(S) \subseteq S \tag{3}$$

The converse inclusion is also true. For suppose not; then there is an $x \in S - \Gamma(S)$. By monotonicity, $\Gamma(S - \{x\}) \subseteq \Gamma(S) \subseteq S$. The left hand side of the $\subseteq$-chain cannot include x since $\Gamma(S)$ does not. We conclude that

$$\Gamma(S - \{x\}) \subseteq S - \{x\}$$

hence $S - \{x\}$ is in $\mathbb{F}$ implying by definition of S that $S \subseteq S - \{x\}$. We reached a contradiction as we set out to do.

That is, $\Gamma(S) = S$ is proved.

And *a final observation*: Say a subset Z of X satisfies $\Gamma(Z) = Z$. Then $Z \in \mathbb{F}$ and therefore $S \subseteq Z$. Thus, S as defined in (1) is indeed the $\subseteq$-*smallest* subset Z of X among all that satisfy $\Gamma(Z) = Z$. □

3.8.4 Definition (Least Fixpoint) Let $\Gamma : 2^X \to 2^X$ be a monotone operator. A $Z \in 2^X$ such that $\Gamma(Z) = Z$ is called a *fixpoint* (also *fixed point*) of Γ.

The $\subseteq$-*smallest* fixpoint S among all fixpoints of Γ is called the *least fixpoint* of Γ. It is denoted by $\overline{\Gamma}$ but also by Γ^∞.

Note the italics in "The" above. The uniqueness follows trivially from the $\subseteq$-*smallest* property.[29] □

3.8.1 An Application of Operators to Cardinality

3.8.5 Definition Let $f : A \to B$ be total and 1-1. We indicate this situation by the symbol $A \preccurlyeq B$ and also $A \overset{f}{\preccurlyeq} B$ if the role of f is important. □

3.8.6 Example Thus

1. $A \sim B$ implies $A \preccurlyeq B$ since we have a 1-1 correspondence $f : A \to B$ which in particular is 1-1 and total.
2. For any A we have $A \preccurlyeq 2^A$ since f that maps $a \in A$ to $\{a\} \in 2^A$ is total and 1-1. By 3.7.10, this example establishes the *distinctness* of the concepts captured by "$\sim$" and "$\preccurlyeq$". This item establishes a counterexample to a possible conjecture that item 1. has a converse.
3. If $A \subseteq B$, then $A \preccurlyeq B$. Indeed, the so-called *inclusion map* $i : A \to B$ given by $i(x) = x$, for all $x \in A$, is 1-1 and total on A. Sometimes we write $i : A \subseteq B$.
4. $A \overset{f}{\preccurlyeq} B$ iff for some $C \subseteq B$ we have $A \sim C$. Indeed, for the "if" let the function $g : A \to C \subseteq B$ be 1-1, total on A and be onto C. Trivially, $g : A \to B$ is total and 1-1 (not necessarily onto). For the "only if" define $C \overset{Def}{=} \text{ran}(f)$ and verify claim. □

3.8.62. motivates the introduction of a new symbol:

3.8.7 Definition If $A \nsim B$ but $A \preccurlyeq B$, then we write $A \prec B$. □

The following theorem that variously goes under the pair of names Cantor and Bernstein or, alternatively, Schroeder and Bernstein and even only Bernstein is remarkable in establishing the connection between $\preccurlyeq$ and $\sim$.[30]

3.8.8 Theorem (Dedekind) *Let* $A \overset{f}{\preccurlyeq} B$ *and* $B \overset{g}{\preccurlyeq} A$. *Then* $A \sim B$.

[29] If T is also a least fixpoint, then $T \subseteq S$, and since S is also least, we have $S \subseteq T$.

[30] As a historical footnote, we observe (cf. Levy (1979), Wilder (1963)) that the theorem was actually *proved* by Dedekind in 1887 (cf. (Dedekind, 1888, p.447)), then was *only conjectured* by Cantor 8 years later, in 1895, and was re-proved by F. Bernstein in 1898 (cf. Borel (1928)). E. Schröder offered a proof independently as well, which supports also the attribution "Schröder-Bernstein" to the theorem.

Proof This proof follows an idea from Dieudonné (1960) but here we use operators (loc. cit. does not). The assumption says that $f : A \to B$ and $g : B \to A$ are each total and 1-1. Define the operator $\Gamma : 2^A \to 2^A$ by

$$\text{For all } Z \subseteq A, \quad \Gamma(Z) \stackrel{Def}{=} (A - g[B]) \cup gf[Z] \tag{1}$$

Γ is monotone since $Z \subseteq Z'$ implies $gf[Z] \subseteq gf[Z']$. Let then X be the $\subseteq$-least fixpoint Γ^∞ of Γ, so

$$\Gamma(X) = X \subseteq A \tag{2}$$

Define

$$(B \supseteq)X' = f[X] \tag{3}$$

$$Y = A - X \tag{4}$$

$$Y' = B - X' \tag{5}$$

Thus

$$A = X \cup Y \text{ and } X \cap Y = \emptyset \tag{4'}$$

and

$$B = X' \cup Y' \text{ and } X' \cap Y' = \emptyset \tag{5'}$$

We next derive the counterpart —$g[Y'] = Y$— of (3) that we will call (3′).

$$\begin{aligned}
g[Y'] &= g[B - X'] \quad \text{by (5)} \\
&= g[B] - g[X'] \quad \text{by Exercise 3.9.15} \\
&= g[B] - gf[X] \quad \text{by (3)} \\
&= g[B] \cap (A - gf[X]) \\
&= A - \Big(A - \Big(g[B] \cap (A - gf[X])\Big)\Big) \quad \text{double negation relative to } A \\
&= A - (A - g[B]) \cup gf[X] \quad \text{de Morgan relative to } A \\
&= Y \quad \text{(2) followed by (4)}
\end{aligned} \tag{3'}$$

It is clear now that $h : A \to B$ given below (cf. (4′))

$$h(x) \stackrel{Def}{=} \begin{cases} f(x) & \text{if } x \in X \\ g^{-1}(x) & \text{if } x \in Y \end{cases}$$

is a 1-1 correspondence $A \sim B$. Of course, g^{-1} is a function —since $g : Y' \to Y$ is 1-1— and is onto Y' (by (3′)). □

3.8.9 Example (The self-contradictory set of all sets) The quoted term in the Example caption goes back to the time when mathematicians were still shocked from the discovery of set theory paradoxes by Russell, Burali-Forti and others (cf. Wilder (1963)) that invariably

pointed to huge or enormously inclusive sets as the culprits. Yet, until such time as Russell introduced the remedy —of requiring sets to *be built by stages* and not to just "happen"— they still considered *all collections* as *sets* and employed the nickname "self contradictory sets" for those examples that were proper classes as we call them in the *current terminology*.

This example outlines the discovery that $\mathbb{U}$ was a "self contradictory set".

That $\mathbb{U}$ cannot be a set —and "yet *was* back then, a set" (all collections were; hence it turned out to be a "self contradictory set")— was actually proved early on in the development of set theory. This was *not* deduced as a result of "the Russell non-set set" being a subset of $\mathbb{U}$ (true the latter contains *all* "sets").

Besides, such an avenue (via 2.3.6) would *not* necessarily lead one to conclude that $\mathbb{U}$ is not a set too, considering that the subclass theorem was not yet known. Nor was it known yet that $x \in x$ is false for all x —a fact that would entail $\mathbb{U}$ to equal the Russell non-set set (See also 2.2.1) and thus be a non-set set itself.

But still there was a clear contradiction to "$\mathbb{U}$ being a set" that was argued, essentially, in the following less elementary way:

1. Since $\mathbb{U}$ contains *everything* —$\mathbb{U} \stackrel{Def}{=} \{x : x = x\}$— in particular *every member* of *any* set S is in $\mathbb{U}$ and thus $S \subseteq \mathbb{U}$. In particular, $\mathscr{P}(\mathbb{U}) \subseteq \mathbb{U}$ and thus (3.8.6.3) $\mathscr{P}(\mathbb{U}) \preccurlyeq \mathbb{U}$.
2. On the other hand, by 3.8.6.2 we have $\mathbb{U} \preccurlyeq \mathscr{P}(\mathbb{U})$.

By 3.8.8 we have $\mathscr{P}(\mathbb{U}) \sim \mathbb{U}$ contradicting 3.7.10. □

3.9 Exercises

1. Give an example of two equivalence relations R and S on the set $A = \{1, 2, 3\}$ such that $R \cup S$ is *not* an equivalence relation.

2. Draw the *Hasse diagram* of the order $\subset$ defined on the set $2^{\{1,2,3\}}$.

3. Given the *left field* $A = \{0, 1, 2\}$ and *right field* $B = \{4, 5\}$. Which of the following functions from A to B is *partial, total, nontotal, 1-1, onto*?

(i) $\{(0, 4)\}$
(ii) $\{(0, 4), (1, 4), (2, 4)\}$
(iii) $\{(0, 4), (2, 5)\}$.

4. Prove that if R and S are any equivalence relations on any set A, then $(R \cup S)^+$, the *transitive closure* of their union, is also an equivalence relation.

5. Let P be a reflexive relation on A that satisfies $aPb \wedge aPc \to bPc$. Prove that P is an equivalence relation on A.
Caution. This $aPb \wedge aPc \to bPc$ is not exactly transitivity!

6. ("False theorem") Let me "prove" that *every symmetric and transitive relation P on some set A is an equivalence relation*. OK, aPb implies bPa by symmetry and thus we have aPa by transitivity. Reflexivity proved. Done.

a. Actually give a counterexample to the above "theorem": Propose a small set A and construct a symmetric and transitive relation on it which is not reflexive.
b. So the theorem is false, but if you were to grade the "proof" you would have to find where it misstepped. Where exactly *is* the error in the proof?

7. Find *two* right inverses of $f : A \to B$, where $A = \{1, 2, 3\}$, $B = \{a, b\}$ and $f = \{(1, b), (2, b), (3, a)\}$.

8. Let $\mathbb{F} : \mathbb{X} \to \mathbb{Y}$ be a function, and $\mathbb{A} \subseteq \mathbb{Y}$, $\mathbb{B} \subseteq \mathbb{Y}$. Prove
(a) $\mathbb{F}^{-1}[\mathbb{A} \cup \mathbb{B}] = \mathbb{F}^{-1}[\mathbb{A}] \cup \mathbb{F}^{-1}[\mathbb{B}]$
(b) $\mathbb{F}^{-1}[\mathbb{A} \cap \mathbb{B}] = \mathbb{F}^{-1}[\mathbb{A}] \cap \mathbb{F}^{-1}[\mathbb{B}]$
(c) if $\mathbb{A} \subseteq \mathbb{B}$, then $\mathbb{F}^{-1}[\mathbb{B} - \mathbb{A}] = \mathbb{F}^{-1}[\mathbb{B}] - \mathbb{F}^{-1}[\mathbb{A}]$.
Is this last equality true if $\mathbb{A} \not\subseteq \mathbb{B}$? Why?

9. Let $\mathbb{F} : \mathbb{X} \to \mathbb{Y}$ be a function, and $\mathbb{A} \subseteq \mathbb{X}$, $\mathbb{B} \subseteq \mathbb{X}$. Prove
(a) $\mathbb{F}[\mathbb{A} \cup \mathbb{B}] = \mathbb{F}[\mathbb{A}] \cup \mathbb{F}[\mathbb{B}]$
(b) $\mathbb{F}[\mathbb{A} \cap \mathbb{B}] \subseteq \mathbb{F}[\mathbb{A}] \cap \mathbb{F}[\mathbb{B}]$
(c) if $\mathbb{A} \subseteq \mathbb{B}$, then $\mathbb{F}[\mathbb{B} - \mathbb{A}] \supseteq \mathbb{F}[\mathbb{B}] - \mathbb{F}[\mathbb{A}]$.
Can the above inclusions be sharpened to equalities? Why?

10. Which parts, if any, of the above two problems generalize to the case that $\mathbb{F}$ is just a relation?

11. Let $\mathbb{G}$ be a function and $\mathbb{F}$ a family of sets. Prove
(a) $\mathbb{G}^{-1}[\bigcup \mathbb{F}] = \bigcup \mathbb{G}^{-1}[\mathbb{F}]$
(b) $\mathbb{G}^{-1}[\bigcap \mathbb{F}] = \bigcap \mathbb{G}^{-1}[\mathbb{F}]$
(c) $\mathbb{G}[\bigcup \mathbb{F}] = \bigcup \mathbb{G}[\mathbb{F}]$
(d) $\mathbb{G}[\bigcap \mathbb{F}] \subseteq \bigcap \mathbb{G}[\mathbb{F}]$ (can $\subseteq$ be replaced by $=$? Why?)

12. Let $\mathbb{F}$ be a function, and $\mathbb{A}$ a class. Prove
(a) $\mathbb{F}[\mathbb{F}^{-1}[\mathbb{A}]] \subseteq \mathbb{A}$
(b) $\mathbb{F}^{-1}[\mathbb{F}[\mathbb{A}]] \supseteq \mathbb{A}$, provided that $\mathbb{A} \subseteq \mathrm{dom}(\mathbb{F})$.
Show by appropriate concrete examples that the above inclusions cannot be sharpened, in general, to equalities.

13. Let the function $\mathbb{F}$ be 1-1, while $\mathbb{A} \subseteq \mathrm{dom}(\mathbb{F})$ is an arbitrary class. Show that $\mathbb{F}^{-1}[\mathbb{F}[\mathbb{A}]] = \mathbb{A}$.
State and prove an appropriate converse.

14. Let $\mathbb{B} \subseteq \mathrm{ran}(\mathbb{G})$. Prove $\mathbb{G}[\mathbb{G}^{-1}[\mathbb{B}]] = \mathbb{B}$.
State and prove an appropriate converse.

15. Let $\mathbb{F}$ be a 1-1 function and $\mathbb{A} \subseteq \mathbb{B} \subseteq \text{dom}(\mathbb{F})$.
(a) Prove $\mathbb{F}[\mathbb{B} - \mathbb{A}] = \mathbb{F}[\mathbb{B}] - \mathbb{F}[\mathbb{A}]$
(b) Prove a suitable converse
Is the restriction $\mathbb{A} \subseteq \mathbb{B} \subseteq \text{dom}(\mathbb{F})$ necessary? Why?

16. For any relations $\mathbb{S}$, $\mathbb{T}$ prove
(1) $(\mathbb{S}^{-1})^{-1} = \mathbb{S}$
(2) $\text{dom}(\mathbb{S}) = \text{ran}(\mathbb{S}^{-1})$
(3) $\text{ran}(\mathbb{S}) = \text{dom}(\mathbb{S}^{-1})$
(4) $(\mathbb{S} \cup \mathbb{T})^{-1} = \mathbb{S}^{-1} \cup \mathbb{T}^{-1}$.

17. Show that if a function $\mathbb{F}$ is a set, then so are both $\text{dom}(\mathbb{F})$ and $\text{ran}(\mathbb{F})$.

18. Show for a relation $\mathbb{S}$ that if both the range and the domain are sets, then $\mathbb{S}$ is a set.

19. Show that if a relation $\mathbb{S}$ is transitive, then so is $\mathbb{S}^{-1}$.

20. Show that for any relations $\mathbb{P}$ and $\mathbb{Q}$, $(\mathbb{P} \circ \mathbb{Q})^{-1} = \mathbb{Q}^{-1} \circ \mathbb{P}^{-1}$.

21. Show that $(\mathbb{S}^{-1})^{+} = (\mathbb{S}^{+})^{-1}$.
Hint. $a(\mathbb{S}^{-1})^{+}b$ means for some finite sequence $t_1, \ldots, t_k$ we have

$$a\mathbb{S}^{-1}t_1\mathbb{S}^{-1}t_2 \ldots t_k\mathbb{S}^{-1}b$$

and $a(\mathbb{S}^{+})^{-1}b$ means $b(\mathbb{S}^{+})a$, that is, for some finite sequence $r_1, \ldots, r_m$ we have

$$b\mathbb{S}r_1\mathbb{S}r_2 \ldots r_m\mathbb{S}a$$

22. If $\mathbb{F} : \mathbb{A} \to \mathbb{B}$ is a 1-1 function, show that $\mathbb{F}^{-1} : \mathbb{B} \to \mathbb{A}$ is also a function (single-valued).

23. If $\mathbb{F} : \mathbb{A} \to \mathbb{B}$ is a 1-1 correspondence, show that so is $\mathbb{F}^{-1} : \mathbb{B} \to \mathbb{A}$.

24. Let A be an enumerable set. Prove that we can enumerate it so that every one of its members is enumerated infinitely many times.

25. Prove that if A is infinite and B is finite, then $A \cup B \sim A$.

26. Prove that every infinite set (in the sense of Definition 3.6.1) has an enumerable subset.

27. Prove that if A is infinite and B is enumerable, then $A \cup B \sim A$.

28. (*Dedekind Infinite*) Dedekind gave this alternative definition of infinite set, namely

> A is infinite iff for some proper subset of A —let's call it S— we have $A \sim S$.

Prove that his definition is equivalent (two directions!) to the one we introduced in this chapter (3.6.1).

29. Suppose someone defined certain subsets of $\mathbb{N}$ —one for each natural number x, which thus is *naming* one such subset (possibly several x names are given to the same subset) and they called them W_x.
Is the following subset of $\mathbb{N}$,

$$\{x \in \mathbb{N} : x \notin W_x\}$$

a W_x? Why?[31]

30. Prove that the set of all *finite* subsets of $\mathbb{N}$ is countable.
Hint. Refer to 3.7.9 and use it to uniquely associate an integer with each finite subset of $\mathbb{N}$. Note that there is a last "1" in the array of outputs of the characteristic function of a finite subset of $\mathbb{N}$. Now read that array of outputs backwards and think "binary notation".

31. Cantor proved, using diagonalisation, that $(0, 1) = \{x \in \mathbb{R} : 0 < x < x1\}$ is uncountable.
Prove that if $a < b$, then $(0, 1) \sim (a, b) = \{x \in \mathbb{R} : a < x < b\}$.
Hint. You need to define a function $f : (0, 1) \to (a, b)$ that is total, 1-1 and onto. You have to "stretch" 1 (the length of $(0, 1)$) to $b - a$ (length of (a, b)).

32. Next prove that $\mathbb{R}$ is not "larger" than an interval. To this end show that $(-1, 1) \sim \mathbb{R}$.
Hint. Try the function f on $\mathbb{R}$ defined by

$$f(x) = \frac{x}{1 + |x|}$$

Clearly, f is total. Next, we see that it is 1-1. Indeed, let

$$\frac{a}{1 + |a|} = \frac{b}{1 + |b|} \tag{1}$$

where a and b are in $\mathbb{R}$. This leads to

$$a - b = b|a| - a|b| \tag{2}$$

Since by (1) $ab \geq 0$, (2) has only two solutions. Both lead to 1-1-ness ($a = b$).
For ontoness consider *three cases* of $-1 < c < 1$ and find a $b \in \mathbb{R}$ such that $f(b) = c$:
The cases are $c = 0$, $-1 < c < 0$ and $0 < c < 1$.

33. Let A and C be sets. Take for granted that $\{X, Y\}$ is a set (2.3.1) for any sets or atoms X, Y.
Without using Principles 0, 1, 2, but using Principle 3 prove that $A \times \{C\}$ is a set.

34. Let A and B be sets. Take for granted that $\{X, Y\}$ is a set (2.3.1) for any sets or atoms X, Y.
Without using Principles 0, 1, 2, but using Principle 3 prove that $A \times B$ is a set.
Hint. Use the preceding exercise.

[31] W_x sets are actually defined and used in computability and were introduced by Rogers. Cf. Rogers (1967), Tourlakis (2022).

A Tiny Bit of Informal Logic

4

Overview

We have become somewhat proficient in using informal logic in our arguments about aspects of discrete mathematics.

Although we have already used quantifiers, $\exists$ and $\forall$, we did so mostly viewing them as symbolic abbreviations of English texts about mathematics. In this chapter we will expand our techniques in logic, extending them to include the manipulation of quantifiers, such as formal —i.e., syntax-based— techniques towards adding $\forall$, $\exists$ to the left of a formula, and also removing them when they are prefixes.

4.1 Enriching Our Proofs to Manipulate Quantifiers

Manipulation of quantifiers boils down, mostly, to "how can I remove a quantifier from the very beginning of a formula?" and "how can I add a quantifier at the very beginning of a formula?" Once we learn this technique we will be able to reason within mathematics with ease.

But first let us define once and for all what a mathematical proof *looks like*: its *correct*, *expected syntax* or *form*, that is.

We will need some concepts to begin with.

1. *The alphabet and structure of formulas*. Formulas are strings. The *alphabet* —that is set— of *symbols* that we use to write down formulas contains, *at a minimum*,

$$=, \neg, \wedge, \vee, \rightarrow, \equiv, (,), \forall, \exists, \text{object variables}[1]$$

[1] That is, variables that denote *objects* such as numbers, arrays, matrices, sets, trees, etc.

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0_4

We finitely generate the *infinite set of object variables* using single letters, if necessary with primes and/or subscripts: A, x, y'', w'''_{23}, u_{501}.

2. One normally works in a mathematical area of interest, or *mathematical theory* —such as Geometry, Set Theory, Number Theory, Algebra, Calculus— where one needs additional symbols to write down formulas. That is, symbols like

$$0, \emptyset, \in, \subset, \int, \circ, +, \times$$

and many others.

3. Mathematicians as a rule get —after a lot of practise— to recognise the *formulas* and *terms* in the math areas of their interest without being necessarily taught *the recursive definition of the syntax* of these. We will not give the syntax in these notes either (but see Tourlakis (2008) if you want to know). Thus one learns to be content with getting to know formulas and terms by their behaviour and through use, rather than by their exact definition of syntax.

- *Terms* are "function calls", in the jargon of the computer savvy person. These calls take math *objects* as inputs and return math *objects* as outputs. *Examples* of *calls* or *terms* —all drawn from our familiar arithmetic— are: $x + y$, $x \times 3$, $0 \times x + 1$ (one is told that $\times$ is stronger than $+$, so, notwithstanding the bracket-parsimonious notation "$0 \times x + 1$", we know it means "$(0 \times x) + 1$", so this call returns 1, no matter what we plugged into x).
- *Formulas* are also function calls, but their output is *restricted* (by their syntax that I will not define carefully!) to be one or the other of the truth values *true* or *false* (denoted in this book by **t** or **f**) but nothing else! Their inputs, just as in the case for terms, are any math objects. *Examples* are: $2 < 3$ (which is true, **t**), $(\forall x)x = x$ (**t**), $(\forall x)x = 0$ (**f** over, say, the reals $\mathbb{R}$), $(\exists x)x = 0$ (**t** over the reals and natural numbers), $x = 0$ the latter being neither true nor false; the answer depends on the input we put in the "input variable" x.
  More: $x = x$ (**t**), an answer that is independent of input; $x = 0 \to x = 0$ (**t**), an answer that is independent of input; $x = 0 \to (\forall x)x = 0$, which is neither true nor false; the answer depends on the input in x. The input variable is the *leftmost* x; the other two occurrences of "x" are *bound* —as we say— and *unavailable* to accept inputs. See also below.
- If an *occurrence* of a variable in a formula *is* available to accept inputs, then non logicians would normally call it an *occurrence as an input variable*. Logicians (in their classrooms and in the literature they author) would rather call such occurrences *free occurrences*.
  At the expense of writing style, "occurrence" occurred no less than four times in the short passage above. The aim is *emphasis*: It is *not* a *variable* x itself that is free

(input) or bound (not available for input) in a formula, but it is the *occurrences* of said variable that we are speaking of, as the immediately preceding example makes clear.

4. In $(\forall x)x = 0$ the variable x is non input, it is "bound" we say. Just like this: $\Sigma_{i=1}^{4} i$, which means $1 + 2 + 3 + 4$ and "i" is *not* available for input: Something like $\Sigma_{3=1}^{4} 3$ is, of course, nonsense! Similar comment for $\exists$.
5. We call $\forall, \exists, \neg, \wedge, \vee, \rightarrow, \equiv$ the "*logical connectives*", the last five of them being called *Boolean connectives*. Logicians avoid cluttering notation with a lot of brackets, agreeing that the first three connectives have the same "strength" or "priority"; the *highest*. To the remaining connectives they assign priorities that are decreasing as we walk towards the right. *As a habit, logicians omit outermost brackets outright unless a formula is used as part of another formula —as a subformula.* For example, in $A \wedge (B \vee C)$ the subformula "$(B \vee C)$" is equipped with outermost brackets which protect the "weaker" $\vee$ from the "stronger" $\wedge$ and allow the former to bind B forcing $\wedge$ to wait and act on the *entire* $B \vee C$ instead.
   As other examples, if A and B are —*denote*, is meant by "are"— formulas, then $\neg A \vee B$ means $(\neg A) \vee B$ because $\neg$ wins the competition with $\vee$ as to who binds with A.
   Similarly, if we want $(\forall x)$ to apply to the entire $A \rightarrow B$ we must write $(\forall x)(A \rightarrow B)$.
   What about $A \rightarrow B \rightarrow C$ and $A \equiv B \equiv C$? Brackets are *implied* from right to left: $A \rightarrow (B \rightarrow C)$ and $A \equiv (B \equiv C)$. And this? $(\exists y)(\forall x)\neg A$. Brackets are *implied*, again, from right to left: $(\exists y)\big((\forall x)(\neg A)\big)$.
   The *expression* (or string of symbols) from the left bracket of the indicated "$(\forall x)$" (respectively, "$(\exists x)$") to the end of the formula A on which the quantifier acts (see below)

$$(\forall x)A \text{ —respectively, } (\exists x)A$$

   is called the *scope* of the quantifier $(\forall x)$ (respectively, $(\exists x)$).
   For example, in $(\forall x)A \rightarrow B$ the scope of the $(\forall x)$ is the entire expression "$(\forall x)A$", that is, no part of the string "$\rightarrow B$" belongs to the scope of the displayed $(\forall x)$.
6. **Boolean deconstruction**. A formula like $(\forall x)A \rightarrow B$ can be *deconstructed* in the Boolean sense into the formulas $(\forall x)A$ and B. If I knew more about B —say, I knew that it is "$x = 3 \rightarrow x = 7$", then I can deconstruct further.
   So, now I have got

$$(\forall x)A, \quad x = 3, \quad x = 7$$

   The last two have no Boolean structure so deconstructing stops with them. How about $(\forall x)A$? This cannot be deconstructed either, even if A has Boolean structure! *Such structure is locked up and hidden in the scope of* $(\forall x)$.
   We call the formulas where deconstruction stops "*prime*". A *prime formula* is one with no Boolean structure, e.g., $x < 8$, or one of the form $(\forall x)A$ or $(\exists x)A$.

Every formula is either prime or can be deconstructed into prime components. A prime formula is one with no explicit Boolean connectives. Such connectives are either totally absent in it —e.g., $x < y$— or are buried in the scope of a quantifier —e.g., $(\exists x)(x = 0 \vee x > 5)$.
Thus prime formulas are "atomic" —no further deconstructible— as far as Boolean structure is concerned.

4.1.1 Remark (Tautologies) A formula A is a *tautology* iff it **is true due to its Boolean structure**, according to truth tables (Remark 2.3.4) no matter what the values of its prime formulas into which it is deconstructed *are postulated to be*. *Postulated to be*: This signifies that we do *not* (attempt to) compute the *intrinsic* truth value of a prime formula *when we check whether A is a tautology or not*.[2]

For example, $x = x$ is a prime formula and thus its *postulated* value could be *any* one of **t** or **f**. Thus it is *not* a tautology, even though, *intrinsically* is *true*, no matter what the value of x may be. □

4.1.2 Example

1. $(\forall x)A$ is not a tautology as it has two possible truth values (being a prime formula) in principle.
2. $x = 0 \to x = 0$ is a tautology. Which are its prime (sub) formulas?
3. $(\forall x)x = 0 \to x = 0$ is not a tautology. As noted, to determine tautologyhood we *do not evaluate prime formulas*; we just consider *each* of the two scenarios, **t** or **f**, for each prime formula and use truth tables to compute the overall truth value.

If we *did* evaluate $(\forall x)x = 0$ we would see that (say over the natural numbers, or reals, or complex numbers) it is false.[3] So the implication is true. However it is *not true as a Boolean formula*.

□

So, how do we show that $(\forall x)A$ is true (if it is)? Well, in easy cases we try to see if A is true for all values of x —no matter where these values come from! That failing, we will use a proof (see 4.1.11).

Similarly for $(\exists x)A$. To show it is true (if it is) we try to see if A is true for *some* value of x. Often we just guess one such value that works, say c, and then verify the truth of A when $x = c$. That failing, we will use a proof.

[2] After all, not all prime formulas have intrinsic values; $x = y$ does not. It *depends* on assumed values of x and y.

[3] If we are doing our mathematics restricted to the set {0}, then, in this "theory" the formula *is* true!

4.1.3 Definition (Tautological implication)
We say that the formulas $A_1, A_2, \ldots, A_n$ *tautologically imply* a formula B, in symbols

$$A_1, A_2, \ldots, A_n \models_{taut} B$$

meaning

"the truth of $A_1 \wedge A_2 \wedge \ldots \wedge A_n$ *implies* the truth of B"

that is, that

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \to B \text{ is a tautology} \tag{1}$$

□

4.1.4 Remark Note that we do *not* care to *check*, or even *state*, what happens if $A_1 \wedge A_2 \wedge \ldots \wedge A_n$ is false because the formula in (1) is then trivially true.

So, a tautological implication $A_1, A_2, \ldots, A_n \models_{taut} B$ says that B is true provided we proved (or accepted) that the lhs of $\models_{taut}$ is true.

$\models_{taut}$ *propagates truth from left to right.* □

4.1.5 Example Here are some easy and some involved tautological implications. They can all be verified using truth tables, either building the tables in full, or taking shortcuts.

1. $A \models_{taut} A$
2. $A \models_{taut} A \vee B$
3. $A \models_{taut} B \to A$
4. $A, \neg A \models_{taut} B$ —any B. Because I do *work* only if $A \wedge \neg A$ is true! See above.
5. $\mathbf{f} \models_{taut} B$ —any B. Because I do *work* only if lhs is true! See above.
6. Is this a valid tautological implication? $B, A \to B \models_{taut} A$, where A and B are distinct. No, for if A is false and B is true, then the lhs is true, but the rhs is false!
7. Is this a valid tautological implication? $A, A \to B \models_{taut} B$? Yes! Say $A = \mathbf{t}$ and $(A \to B) = \mathbf{t}$. Then, from the truth table of $\to$, it *must* be $B = \mathbf{t}$.

Statements such as "$B = \mathbf{t}$" are shorthand for "B evaluates as $\mathbf{t}$".

8. How about this? $A, A \equiv B \models_{taut} B$? Yes! Verify!
9. How about this? $A \vee B \equiv B \models_{taut} A \to B$? Yes! I verify:
   First off, *assume* lhs of $\models_{taut}$ —that is, $A \vee B \equiv B$— is true.

Two cases:

- $B = \mathbf{f}$. Then I need the lhs of $\equiv$ to be false ($\mathbf{f}$) to satisfy the italicised "assume". So $A = \mathbf{f}$ as well and clearly the rhs of $\models_{taut}$ is true with these values.
- $B = \mathbf{t}$. Then I need not worry about A on the lhs. The rhs of $\models_{taut}$ is true by truth table of $\rightarrow$.

10. $A \wedge (\mathbf{f} \equiv A) \models_{taut} B$, for any B. Well, just note that the lhs of $\models_{taut}$ is $\mathbf{f}$ so we need to do no work with B to conclude that the implication is valid.
11.

$$A \rightarrow B, C \rightarrow B \models_{taut} A \vee C \rightarrow B$$

This is nicknamed "proof by cases" for the obvious reasons. Verify this tautological implication! □

Before we describe what a logical proof is, we need some discussion and a definition.

4.1.6 Example (A Cautionary Tale) Consider the formula

$$(\exists y)\neg x = y$$

written more simply

$$(\exists y)x \neq y \tag{1}$$

(1) says "for any value of x there is a value of y that is different".

If the set where we do math contains two or more (distinct) elements —which is, *in practice*, always the case— then (1) is true.

The same will be observed —they are *true statements*— if we substitute z or w for x to obtain

$$(\exists y)z \neq y \text{ "for any value of } z \text{ there is a value of } y \text{ that is different".} \tag{1'}$$

and

$$(\exists y)w \neq y \text{ "for any value of } w \text{ there is a value of } y \text{ that is different".} \tag{1''}$$

However suppose I substitute y for x in (1). I obtain

$$(\exists y)y \neq y \tag{2}$$

which says "there is a value of y that is different from itself" —*obviously a false statement*.

What caused this *distortion of (original) meaning* is that **an object that we substituted into a *free variable occurrence* x *contained a variable* y *that was* "*captured*"** by a quantifier, as we say.

So we always disallow substitutions that cause capture! We say they are *illegal* or *impossible*.

□ 

4.1.7 Definition Let A be a formula and x a variable. The symbol $A[x]$ indicates our interest in the *possibly input* variable x of A. If y and z are actually the only input (free) variables of A I can indicate this without words by writing $A(y, z)$.

I explain "possibly". For example,

1. If A is $y = z$ then x *does not even occur* in A. But I said *possibly*! I can still write $A[x]$. I can also write $A(y, z)$.
2. In the case where A is $(\forall x)x = 1$, A *cannot* receive inputs in the so-called *bound variable* x. Even though I *may* write $A[x]$, this is just wishful thinking and x does not occur free in A or as we variously say x is not an *input variable* of A or A does not depend on x.
3. A is $(\forall x)x = y$. I can write $A[x]$ but x is actually not free in A. I actually have $A(y)$.
4. Let now t be any term —a constant or variable or a function call.
   Having declared interest in a (possibly) free variable of A by writing down $A[x]$, I can next write $A[t]$ *in the same context* meaning the substitution of t into x —that is, a search and replace operation: find *all* free occurrences of x in A and *replace them all* by t; but *do abort* the entire substitution (illegal) operation if *any* replacement caused some variable in t to become bound (capture).
   In the first illustration above, and assuming t is $g(w)$, we get $A[t]$ is $y = z$.
   If now $B[x]$ is $(\forall w)w = x$ then $B[t]$ is illegal since it means $(\forall w)w = g(w)$.
5. If I wrote $A(y, z)$, then $A(t, z)$ means $A(g(w), z)$ which is legal if A is as in item 1. above. □

The job of a mathematical proof is to start from *established* (previous theorems) truths, or *assumed* truths (axioms) and unfailingly *preserve truths* **in all the proof's steps** as we develop it.


This description is word-parsimonious and sounds circular: No chicken and egg dilemma here, however. "Previous theorems" can be used *only if* we have any of those at any given moment. Else we use just axioms.

In fact, the concept of proof is defined in terms of axioms (and rules of inference) alone. 

Thus, by the truth-preservation property, we will have produced, in particular, a truth at the very last step of a proof. This is what we call a *proved theorem*.

What are our axioms, our starting assumptions, when we do proofs?

4.1.8 Definition First off, in *any proof that we will write in math* there are axioms that *are independent of the type of math that we do*, whether it is set theory, number theory, algebra, calculus, etc. Such axioms are called *logical* (logic-specific, that is).

Our logical axioms are

1. All tautologies; these need no defence as "start-up truths".
2. Formulas of the *form* $(\forall x)A[x] \to A[t]$, for any formula A, variable x and "object" t. This object can be as simple as a variable y (might also be the same as x), constant c, or as complex as a "*function call*", $f(t_1, t_2, \ldots, f_n)$ where f accepts n-inputs, and the inputs shown here are already available objects.

A couple of comments: This is a *bona fide* start-up *truth* as its says "if $A[x]$ is true for all x-values,[4] then it is true also if we plug a specific value/object into x".
Refer also to Definition 4.1.7 which leads to one more comment: If $A[t]$ is aborted due to capture, then the *axiom form* 2 or —*axiom schema* 2 as we call them properly in writings on logic— *does not contribute an axiom* for the specific A, x and t that participated in the capture phenomenon.

3. Formulas of the form $A[x] \to (\forall x)A[x]$, for any formula A where x is *not* an input variable (*not free*).
For example say A is $3 = 3$. This axiom says then, "if $3 = 3$ is true, then so is $(\forall x)3 = 3$". Sure! $3 = 3$ does not *depend* on x. So saying "for all values of x we have $3 = 3$" is the same as saying just "we have $3 = 3$".
4. Formulas of the form $(\forall x)(A[x] \to B[x]) \to (\forall x)A[x] \to (\forall x)B[x]$, for any formulas A, B, and variable x.
This is a useful start-up *truth*. We verify it informally (semantically) for the special case that A, B have no other free variables besides (possibly) x: Since by truth tables we have that $X \wedge Y \to Z$ and $X \to Y \to Z$ are equivalent, we view this axiom as

$$(\forall x)(A[x] \to B[x]) \wedge (\forall x)A[x] \to (\forall x)B[x]$$

Now the hypothesis of the last "$\to$" says that

$$A[x] \text{ is true for all } x \tag{$\dagger$}$$

and

$$A[x] \to B[x] \text{ is true for all } x \tag{$\ddagger$}$$

By ($\ddagger$), every x that makes A true makes B true. But that is all values of x by ($\dagger$). So $B[x]$ is true for all values of x as we wanted to verify.
5. For any choice of variable (here I use "x") $x = x$ is the *identity* axiom, no matter what (is the value of) "x". Note that "For any choice of variable" means that $y = y$ and $w = w$ are also instances of the axiom.
6. $x = y \to y = x$ and $x = y \wedge y = z \to x = z$ are the *equality* axioms.
They can be expressed equally well using variables other than x and y (e.g., u, v and w).

[4] Practicioners usually say "for all x", meaning for all **values** of x.

7. The $\exists$ vs. $\forall$ axiom. For any formula A and any choice of quantified variable, $(\exists x)A[x] \equiv \neg(\forall x)\neg A[x]$ is an axiom.


Note that the right hand side of the "$\equiv$" says "it is not the case that all values of x make $A[x]$ false".


□

The "rules of proving", or *rules of inference*. These are two rules provided up in front and as such they are essential to *start up the proof mechanism*. To indicate this "start-up role" we often call these rules *primary* or *primitive*.

Incidentally you will find I am grossly miscounting the rules in item 2 below:

4.1.9 Definition (Rules)

1. From $A[x]$ I may infer $(\forall x)A[x]$. Logicians write the up-in-front ("primary") rules as fractions without words:

$$\frac{A[x]}{(\forall x)A[x]}$$

this rule we call *generalisation*, or we are using the nickname "*Gen*".

2. I may *construct* (and *use*), using *any* tautological implication *that I have verified*, say, one of this *shape* (*form*)

$$A_1, A_2, \ldots, A_n \models_{taut} B$$

the rule

$$\frac{A_1, A_2, \ldots, A_n}{B}$$

For example, seeing readily that $A, A \to B \models_{taut} B$, we have the rule

$$\frac{A, A \to B}{B} \qquad (MP)$$

This is a very popular rule, known as *modus ponens*, in short MP.

It turns out that MP and Gen is *all* you need to prove theorems, thus, *officially* they are *THE two primary rules*. However, additional tautological implications from 2. help in the *practice* of proofs.

□

1. **Rule Use**: How do we *use* rules? See also Definition 4.1.11 below. If *in a proof* that we are writing we have already written all the formulas of the *numerator* of some rule, then *it is correct* to write next (or at any later step) the *denominator* of the rule.

We say that we have *applied the rule* in order to obtain and write the denominator.

2. The second "rule" above is a rule constructor. Any tautological implication we come up with is fair game: It leads to a *valid rule* since the name of the game (in a proof) is *preservation/propagation of truth*.
   This is *not* an invitation to learn and memorise infinitely many rules (!) but is rather a license to build your own rules as you go, *as long as you endeavoured to* **verify** *first* the validity of the tautological implication they are based on.
3. Gen is a rule that indeed propagates truth: If $A[x]$ is true, that *means* that it is so for all values of x and all values of any other variables on which A depends, which variables I did not show in the $[\ldots]$ notation. But then so is $(\forall x)A[x]$ true, as it says precisely the same thing: "$A[x]$ is true, for all values of x and all values of any other variables on which A depends but I did not show in the $[\ldots]$ notation".
   The only difference between the two notations is that I added some notational *emphasis* in the second —$(\forall x)$.
   For example, if I know that B has just two variables, u and v, I can write it as $B(u, v)$. Then

$$B(u, v) = \mathbf{t} \text{ iff } (\forall u)B(u, v) = \mathbf{t} \text{ iff } (\forall v)B(u, v) = \mathbf{t} \text{ iff } (\forall u)(\forall v)B(u, v) = \mathbf{t}$$

4. Hmm. So is $(\forall x)$ redundant? Yes, but *only as a formula prefix*. In something like this

$$x = 0 \to (\forall x)x = 0 \tag{1}$$

   it is *not* redundant!
   Dropping $\forall$ we *change* the meaning of (1).
   As is, (1) is *not* a true statement (for all values of x, that is). For example, if we set x to be 0, then (1) becomes the false statement $0 = 0 \to (\forall x)x = 0$ since $0 = 0$ is true but (over the integers, say) "$(\forall x)x = 0$" is $\mathbf{f}$. However dropping $(\forall x)$, (1) morphs into "$x = 0 \to x = 0$" which is a tautology; always true.
5. The axioms in 4.1.8 are indispensable to do just logic; that is why we call them *logical axioms*.
   We also use them in *all* math reasoning no matter what type of math it is. However, mathematical theories have their own ***additional*** axioms! These are called *special axioms* but most often "*mathematical axioms*".

We are not going to list them. Why? Because every math branch, or "theory" as we say, has different axioms! Unless we do, say, (axiomatic) set theory there is absolutely no need to list all the set theory axioms.

4.1.10 Example Here is only a *sample* of axioms from *math* (*theories*):

1. Number theory for $\mathbb{N}$:

   - $x < y \vee x = y \vee x > y$ (*trichotomy*)
   - $\neg x < 0$ this axiom indicates that 0 is *minimal* in $\mathbb{N}$. Adding the previous one makes $<$ a total order, so 0 is also *minimum*.
   - Many other axioms that we omit.

2. Euclidean geometry:

   - From two distinct points passes one and only one line.
   - ("Axiom of parallels") From a point A off a line named k —both A and k being on the same plane— passes a unique line on said plane that is parallel to k.
   - Many others that we omit.

3. Axiomatic set theory:

   - For any set A,

   $$(\exists y) y \in A \to (\exists x)\Big(x \in A \wedge \neg(\exists z)(z \in x \wedge z \in A)\Big)$$

   This is the axiom of "foundation" from which one can prove things like $A \in A$ is always *false*.
   It says that *IF* there is *any* element in A *at all* —this is the hypothesis part "$(\exists y) y \in A$"— *THEN* there is some element —this is the part "$(\exists x)\Big(x \in A$"— *below which*, if you follow "$\in$" backwards from it, you will *not* find a z ("$\neg(\exists z)$") that is *both* below x *along* $\in$ *backwards*, *and* also a member of A —this part is "$(z \in x \wedge z \in A)$".

4. And a few others that we omit. □

So what is the *shape* of proofs?

4.1.11 Definition (Proofs and theorems) A proof is a finite sequence of formulas

$$F_1, F_2, \ldots, F_i, \ldots, F_n \tag{1}$$

such that, for *each* $i = 1, 2, \ldots, n$, F_i is obtained as *one* of:

1. It is an axiom from among the ones we listed in 4.1.8.
2. It is an axiom of the *theory* (area of Mathematics) that we are working in.
3. It is a *non-axiom hypothesis* that we find convenient to assume (see examples below for when such hypotheses become applicable).

   In annotations of proofs we denote that a formula written down is a hypothesis by labelling it "hyp" (no quotes).

4. It is the result of "Gen" (nickname for "generalisation") applied to a previous formula F_j. That is, $F_i = (\forall x)F_j$, for some x and $j < i$.
5. It is the result of "$\models_{taut}$" applied to previous formulas F_{j_k}, $k = 1, 2, \ldots, m$. That is, $F_{j_1}, F_{j_2}, F_{j_3}, \ldots, F_{j_m} \models_{taut} F_i$, and all j_r for $r = 1, 2, \ldots, m$ are $< i$.

Such proofs are known as "Hilbert-style proofs". We write them vertically, *one* formula per line, every formula *consecutively numbered*, with annotation to the right of formulas (the "why did I write this?"). Like this

$$\begin{array}{lll} 1) & F_1 & \langle\text{because}\rangle \\ 2) & F_2 & \langle\text{because}\rangle \\ \vdots & \vdots & \vdots \\ n) & F_n & \langle\text{because}\rangle \end{array}$$

Every F_n in (1) is called a theorem. Thus we define

> A theorem A *from* Σ *in* theory $\mathcal{T}$ is a formula that **appears** in a proof *in said theory* with *hypotheses* Σ. We may call A a Σ-theorem in $\mathcal{T}$. We often omit the "in $\mathcal{T}$".

Often one writes $\vdash A$ to symbolically say that A is a theorem. If we must indicate that we worked in some specific theory, say ZFC (set theory), then we may indicate this as

$$\vdash_{ZFC} A$$

If moreover we have had some "*non-axiom* hypotheses" (read on to see when this happens!) that form a set Σ, then we may indicate so by writing

$$\Sigma \vdash_{ZFC} A$$

□

Why Σ for a set of (*non-axiom*) assumptions? Because we reserve upper case latin letters for formulas. For *sets* of formulas we use a *distinguishable* capital letter, so, we chose distinguishable Greek capital letters, such as Γ, Σ, Δ, Φ, Θ, Ψ, Ω. Obviously, Greek capital letters like A, B, E, Z will not do!

Before we do a few *example proofs*, some easy and some more complex, let us establish a few properties of proofs.

4.1.12 Proposition *If $A_1, A_2, \ldots, A_k, A_{k+1}, \ldots, A_n$ is a proof then so is $A_1, A_2, \ldots, A_k$.*

Proof The *syntactic fitness* for a non-axiom, non-hypothesis A_i for inclusion in a proof depends on formulas to its left, not to its right. Thus dropping the "tail" $A_k, A_{k+1}, \ldots, A_n$ will leave a shorter proof $A_1, \ldots A_k$. □

4.1.13 Corollary *A theorem is precisely a formula at the end of some proof.*

Proof Let A be a formula.

If it is at the *end* of a proof, then it is also "in" the proof, hence it is a theorem by 4.1.11.

Let then A appear as A_k in a proof of length $n > k$. Is there a proof with A, that is, A_k, at its end?

Yes. Just chop the tail $A_{k+1}, \ldots, A_n$. Now A finds itself at the end of a proof! □

4.1.14 Proposition (Proof Concatenation) *If $A_1, \ldots, A_n$ and $B_1, B_2, \ldots, B_m$ are proofs, then so is $A_1, \ldots, A_n, B_1, B_2, \ldots, B_m$.*

Proof Exercise 4.2.4 □

4.1.15 Corollary *The concatenation of any finite number of proofs is a proof.*

4.1.16 Proposition (Hypothesis Strengthening) *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash A$, then also $\Delta \vdash A$.*

Proof Exercise 4.2.5 □

4.1.17 Proposition (Quoting Theorems in a Proof) Proved *theorems can be quoted (included* without proof*) in a proof.*

Proof Suppose we have proved A as $\Gamma \vdash A$. With this fact in hand, a proof from Γ as hypotheses is legitimate if along with quoting members of Γ, members of the theory in hand, and logical axioms we also allow ourselves to quote A.

Let such a proof be

$$\boxed{B_1, B_2, \ldots, B_k, A, B_{k+1}, \ldots, B_n} \tag{1}$$

where for simplicity we quoted the Γ-theorem A only once in the above proof.

We view the A as a "macro" that we omitted its expansion (its proof) from (1). The proof (1) is legitimised by noting that it is only an *abbreviation* that omits A's proof and that we can add the proof at any time (see details below).

The benefit from just quoting A is that the part from B_{k+1} to the end may refer to A in applications of rules of inference but need not *know* or *care* what A's proof is. Nothing from the proof of A is made available (in (1)) to quote/use by B_j, where $j > k$.

(1) would transform into the following sequence *if* we *included* the Γ-*proof* of A: $\boxed{\ldots A}$.

$$\boxed{B_1, B_2, \ldots, B_k}, \boxed{\ldots A}, \boxed{B_{k+1}, \ldots, B_n} \tag{2}$$

As before, the part from B_{k+1} to the end *may continue to refer to* A and again the formulas other than A in the proof box for A *are not used* by proof (2).

Now consider: $\boxed{B_1, B_2, \ldots, B_k}$ and $\boxed{\ldots A}$ are Γ-proofs, thus so is their concatenation (4.2.4)

$$\boxed{B_1, B_2, \ldots, B_k}, \boxed{\ldots A}$$

Finally, the tail part $\boxed{B_{k+1}, \ldots, B_n}$ has all its B_j pass the proof test (1.–5. in 4.1.11) as these depend *ONLY* on the sequence $B_1, B_2, \ldots, B_k, A$.

In short, (2) IS a proof according to 4.1.11

The whole point is: By the devise of including not only A but also its (inaccessible) proof we *rendered* (2) *formally correct*, unlike (1). We legitimised (1) as a *practical shorthand*.

□

The following is related:

4.1.18 Proposition (Derived Rules of Inference) *A derived rule of inference is usually written as* $A_1, A_2, \ldots, A_n \vdash B$ *rather than*

$$\frac{A_1, A_2, \ldots, A_n}{B}$$

They are applicable exactly as the primary rules are, that is, if we proved all of

1. $\Gamma \vdash A_1$
2. $\Gamma \vdash A_2$

3. $\Gamma \vdash A_3$
   $\vdots$
n. $\Gamma \vdash A_n$

then we have $\Gamma \vdash B$.

Proof We are given that we have proofs

$$\boxed{A_1, A_2, \ldots, A_n, \ldots, B} \tag{1}$$

$$\boxed{\ldots, A_1}$$

$$\boxed{\ldots, A_2}$$

$$\vdots$$

$$\boxed{\ldots, A_n}$$

Concatenate the previous $n + 1$ derivations in this order

$$\boxed{\ldots, A_1}\boxed{\ldots, A_2}\boxed{\ldots}\boxed{\ldots, A_n}\boxed{A_1, A_2, \ldots, A_n, \ldots, B} \tag{2}$$

The first n proofs, as concatenated together, form a proof from Γ (4.1.14, corollary). Now concatenating box (1) at the right end of the foregoing sequence we get the sequence (2).

Since all the A_i that are hypotheses in the proof (1) are *copies* of those proved from Γ in the previous n boxes —and since in a proof we may repeat a formula that we proved earlier as many times, and anywhere after its first occurrence, we please— it follows that the sequence (2) is a Γ-proof. □

4.1.19 Example (New (derived) rules) A *derived rule* is one we were not given as *primitive* —in 4.1.9— to bootstrap logic, but we can still prove that it propagates truth.

1. We have a new (derived) rule: $(\forall x)A[x] \vdash A[t]$.
   This is called *Specialisation*, or *Spec*.

Aha! We used a non-axiom assumption (hypothesis) here! I write a Hilbert proof to show that $A[t]$ is a theorem if $(\forall x)A[x]$ is a (non-axiom) hypothesis (assumption) —shortened to "hyp".

1)	$(\forall x)A[x]$	⟨hyp⟩
2)	$(\forall x)A[x] \to A[t]$	⟨axiom 2⟩
3)	$A[t]$	⟨1 + 2 + MP⟩

2. Taking t to be x we have $(\forall x)A[x] \vdash A[x]$, simply written as $(\forall x)A \vdash A$.
3. The *Dual Spec* derived rule: $A[t] \vdash (\exists x)A[x]$. We prove it:

1)	$A[t]$	⟨hyp⟩
2)	$(\forall x)\neg A[x] \to \neg A[t]$	⟨axiom 2⟩
3)	$A[t] \to \neg(\forall x)\neg A[x]$	⟨2 + Post⟩
4)	$\neg(\forall x)\neg A[x]$	⟨3 + MP⟩

Line 4 contains $(\exists x)A[x]$ by axiom 7, or, if you prefer, "$(\exists x)A[x]$ is obtained from axiom 7 and an application of Post".

Instead of "tautological implication" we may give as reason just "Post" (no quotes) see line 3 above since it is Post's *completeness theorem* for Boolean Logic that is at play when we invoke "tautological implication" *as a rule of inference* (see 4.1.9, 2.)

Taking t to be x we have $A[x] \vdash (\exists x)A[x]$, simply written as $A \vdash (\exists x)A$. □

There are two principles of proof that we state without proof here (but you *should* try to prove them as Exercises (Sect. 4.2) where several helpful hints are included.).

4.1.20 Remark (Deduction theorem and proof by contradiction)

1. The *deduction theorem* (also known as "proof by assuming the antecedent") states, if

$$\Gamma, A \vdash B \tag{1}$$

then also $\Gamma \vdash A \to B$, provided that in the proof of (1), *all free variables of A were treated as constants*: That is we neither used them to do a Gen, nor substituted objects into them.

The notations "Γ, A" and "$\Gamma + A$" are standard for the more cumbersome $\Gamma \cup \{A\}$.

In practice, this principle is applied to prove $\Gamma \vdash A \to B$, by doing instead the "easier" (1). Why easier? We are helped by an extra hypothesis, A, and the formula to prove, B, is less complex than $A \to B$.

2. Proof by contradiction. To prove $\Gamma \vdash A$ —where A has *no free variables* or is *closed* or is a *sentence*— is equivalent to proving the "constant formula" $\mathbf{f}$ from hypothesis $\Gamma, \neg A$.
3. Why the burden of the non-axiom hypotheses Γ? Because in applying the deduction theorem we usually start with a task like "do $\vdash A \to B \to C \to D$".
   So we go like this:
   - By DThm, it suffices to prove $A \vdash B \to C \to D$ instead (here "Γ" was $\emptyset$).
   - Again, by DThm, it suffices to prove $A, B \vdash C \to D$ instead (here "Γ" was A).
   - Again, by DThm, it suffices to prove $A, B, C \vdash D$ instead (here "Γ" was A, B). □

4.1.21 Remark (Ping-Pong) For any formulas A and B, the formula —where I am using way more brackets than I have to, ironically, to *improve* readability—

$$(A \equiv B) \equiv \Big((A \to B) \wedge (B \to A) \Big)$$

is a tautology (draw up a truth table with one row for each of the possible values of A and B and verify that the equivalence is always $\mathbf{t}$).

Thus to prove the lhs of the second $\equiv$ suffices to prove the rhs:

	$\vdots$	$\vdots$
1)	$(A \to B) \wedge (B \to A)$	⟨suppose I proved this⟩
2)	$(A \equiv B) \equiv \Big((A \to B) \wedge (B \to A) \Big)$	⟨tautology, hence also *axiom*⟩
3)	$A \equiv B$	⟨1 + 2 + tautological implication⟩

In turn, to prove the rhs it suffices to prove each of $A \to B$ and $B \to A$ separately. This last idea encapsulates the *ping-pong* approach to proving equivalences.

Here are a few applications. □

4.1.22 Example 1. Establish $\vdash (\forall x)(A \wedge B) \equiv (\forall x)A \wedge (\forall x)B$.

By ping-pong.

- Prove $\vdash (\forall x)(A \wedge B) \to (\forall x)A \wedge (\forall x)B$. By DThm suffices to do $(\forall x)(A \wedge B) \vdash (\forall x)A \wedge (\forall x)B$ instead.

1)	$(\forall x)(A \wedge B)$	⟨hyp⟩
2)	$A \wedge B$	⟨1 + Spec⟩
3)	A	⟨2 + tautological implication⟩
4)	B	⟨2 + tautological implication⟩
5)	$(\forall x)A$	⟨3 + Gen; OK : x is not free in line 1⟩
6)	$(\forall x)B$	⟨4 + Gen; OK : x is not free in line 1⟩
7)	$(\forall x)A \wedge (\forall x)B$	⟨5 + 6 + tautological implication⟩

- Prove $\vdash (\forall x)A \wedge (\forall x)B \to (\forall x)(A \wedge B)$. By DThm suffices to do $(\forall x)A \wedge (\forall x)B \vdash (\forall x)(A \wedge B)$ instead.

1)	$(\forall x)A \wedge (\forall x)B$	⟨hyp⟩
2)	$(\forall x)A$	⟨1 + tautological implication⟩
3)	$(\forall x)B$	⟨1 + tautological implication⟩

Complete the above proof!

2. Prove $\vdash (\forall x)(\forall y)A \equiv (\forall y)(\forall x)A$. By ping-pong.

- Prove $\vdash (\forall x)(\forall y)A \to (\forall y)(\forall x)A$.
By DThm suffices to do $(\forall x)(\forall y)A \vdash (\forall y)(\forall x)A$ instead.

1)	$(\forall x)(\forall y)A$	⟨hyp⟩
2)	$(\forall y)A$	⟨1 + Spec⟩
3)	A	⟨2 + Spec⟩
4)	$(\forall x)A$	⟨3 + Gen; OK, no free x in line 1⟩
5)	$(\forall y)(\forall x)A$	⟨4 + Gen; OK, no free y in line 1⟩

- Prove $\vdash (\forall y)(\forall x)A \to (\forall x)(\forall y)A$.
Exercise! □

We have seen how to *add* an $(\exists x)$ in front of a formula (4.1.19 3.).

How about *removing* an $(\exists x)$-prefix? This is much more complex than removing a $(\forall x)$-prefix:

The technique can be *proved* to be correct (e.g., Tourlakis (2003a)) but I will omit the proof here —albeit I will ask you to prove it in the Exercises Sect. 4.2 with hints.

Note that I also omitted the proof of the deduction theorem technique. This I will help you prove in the Exercises section.

In preparation for the removal of an $\exists$-prefix proof we will need an important and very useful result, that of renaming the bound variable:

4.1.23 Definition (Substitution Again) Recall Definition 4.1.7. We indicate there that once we declared our interest in the (possibly) free variable x of A by writing $A[x]$, we can in the same context write $A[t]$ to indicate that *all* the free occurrences of x (if any) in A are everywhere replaced by the object (term) t. Recall the caution needed in such substitutions (4.1.6).

Here we add an *explicit notation* for the process "find and replace by the term t all the free occurrences of x in A": The symbol is $A[x \leftarrow t]$.

The substitution operation *compound symbol* $[x \leftarrow t]$ is viewed as an *operator* or *connective* and as such it has the *highest priority* of all connectives.

Thus, if we write $A \wedge B[x \leftarrow t]$ then we mean $A \wedge (B[x \leftarrow t])$. Also, $(\forall y)A[x \leftarrow t]$ means $(\forall y)\Big(A[x \leftarrow t]\Big)$ thus there is no capture of y (if it appears in t) since the substitution took effect **before** the quantifier was applied. For example, obtaining $(\forall x)x = 0$ from $x = 0$ we do not speak of capture of x! It is just the process of formula formation! □

4.1.24 Theorem Technique of removing an ∃-prefix*: Suppose I have that* $(\exists x)A[x]$ *is true —either as an* assumption *or a* theorem that I proved earlier*— and I want to use this and prove* B.

Then I assume *that* $A[z]$ *is true —for* some new variable z *that does not occur in* B *nor in* $(\exists x)A$ *—we call such a variable "fresh".*

Logicians annotate this step in a proof as "aux. hyp. associated with $(\exists x)A[x]$*".*

Now proceed to prove B *using all that is known to you —that is, the axioms of the theory* $\mathcal{T}$ *that you work in, perhaps some non-axiom hypotheses* Γ*, and* $(\exists x)A[x]$, and the *new* non-axiom hypothesis $A[z]$.

Do so by using all free (input-) variables of $A[z]$ *as constants in your proof!*[a]

[a] This is a side-effect of using the deduction theorem in the proof of correctness of this ∃-elimination technique. See 4.2.11.

The technique of removing an ∃ -prefix *guarantees that you did better than*

$$\Gamma, \boxed{A[z]} \vdash_{\mathcal{T}} B$$

In fact, you actually achieved

$$\Gamma \vdash_{\mathcal{T}} B$$

as if you **never** assumed nor used $A[z]$*!*

That is why they call it "auxiliary hypothesis". Once it helps you prove B it drops out*; it does not stay around to get credit!*

4.1.25 Example In practice we often have an assumption $(\exists x)Q$ from which we want to eliminate $(\exists x)$ to benefit from the (possibly) uncovered Boolean structure of Q. This fits with the theorem above taking $\Gamma = \{(\exists x)Q\}$.

For example, prove $\vdash (\exists y)(\forall x)A[x, y] \rightarrow (\forall x)(\exists y)A[x, y]$.

By the DThm it suffices to prove $(\exists y)(\forall x)A[x, y] \vdash (\forall x)(\exists y)A[x, y]$ instead.

1) $(\exists y)(\forall x)A[x, y]$ ⟨hyp⟩
2) $(\forall x)A[x, z]$ ⟨aux. hyp. caused by 1; z is some fresh variable, not in the conclusion⟩
3) $A[x, z]$ ⟨2 + Spec⟩
4) $(\exists y)A[x, y]$ ⟨3 + Dual Spec⟩
5) $(\forall x)(\exists y)A[x, y]$ ⟨4 + Gen; OK, no free x in lines 1 and 2⟩

I said in line 2, "z is some fresh variable, not in the conclusion". Doesn't "fresh" cover the "not in the conclusion?" NO! "*Fresh*" ensures that none of the lines *before* the introduction of z contain it. Freshness is not global to the proof! So, non occurrence in B must be added explicitly.

□

4.1.26 Example Can I also prove the converse of the above? That is $\vdash (\forall x)(\exists y)A[x, y] \rightarrow (\exists y)(\forall x)A[x, y]$.

I will try.

By the DThm it suffices to prove $(\forall x)(\exists y)A[x, y] \vdash (\exists y)(\forall x)A[x, y]$ instead.

1) $(\forall x)(\exists y)A[x, y]$ ⟨hyp⟩
2) $(\exists y)A[x, y]$ ⟨1 + spec⟩
3) $A[x, z]$ ⟨aux. hyp. for 2; z is fresh and not in the conclusion⟩
4) $(\forall x)A[x, z]$ ⟨3 + Gen; Hmm!
Illegal: I should treat the free x of *aux. hyp.* as a constant!⟩

Still, can **anyone** prove this even if I cannot?

A question like this, if you are to answer "no", must be resolved by offering a *counterexample*. That is, a special case of A for which I can *clearly see* that the claim is *not true*.

Here is one such special case:

$$(\forall x)(\exists y)x = y \rightarrow (\exists y)(\forall x)x = y \tag{1}$$

Say we work in $\mathbb{N}$. The lhs of $\rightarrow$ is true, but the rhs is false as it claims that there is a number such that *all* numbers are equal to it. So the implication fails in the special case invalidating also the general case.[5]

□ 

Another useful principle that can be proved, but we will not do so, is that one can *replace equivalents-by-equivalents.* That is, if C is some formula, and if I have

[5] If the general case is valid so would be any of its special cases!

1. $A \equiv B$, via proof, or via assumption, and also
2. A is a subformula of C

then I can *replace* one (or more) occurrence(s) of A in C (as subformula(s)) by B and call the resulting formula C', and be guaranteed the conclusion $C \equiv C'$. That is, from $A \equiv B$, I can prove $C \equiv C'$.

This principle is called the *equivalence theorem*.

Let's do a couple of ad hoc additional examples before we move to the section on Induction.

4.1.27 Example $A \to B \vdash (\forall x)A \to (\forall x)B$.

By the DThm it suffices to prove $A \to B, (\forall x)A \vdash (\forall x)B$ instead.

1)	$A \to B$	⟨hyp⟩
2)	$(\forall x)A$	⟨hyp⟩
3)	A	⟨2 + Spec⟩
4)	B	⟨1 + 3 + MP⟩
5)	$(\forall x)B$	⟨4 + Gen; OK as the DThm hypothesis (line 2) has no free x⟩

□

4.1.28 Example Refer to 4.1.8(7). Let us apply it to $\neg A$ for arbitrary A. We get

$$\vdash (\exists x)\neg A \equiv \neg(\forall x)\neg\neg A \tag{1}$$

We apply the equivalence theorem above. To this end, note that $A \equiv \neg\neg A$ is a tautology, hence an axiom in group 1, and thus a theorem: $\vdash A \equiv \neg\neg A$. Applying the equivalence theorem to (1) we thus obtain:

$$\vdash (\exists x)\neg A \equiv \neg(\forall x)A \tag{2}$$

Applying another tautological implication to (2) we get

$$\vdash (\forall x)A \equiv \neg(\exists x)\neg A$$

which is of the same form as 4.1.8(7) with the roles of $\exists$ and $\forall$ reversed. □

4.1.29 Example $A \equiv B \vdash (\forall x)A \equiv (\forall x)B$.

True due to the equivalence theorem! "C" is "$(\forall x)A$". We replaced (one occurrence of) A by B in C, and we have assumed as starting point that $A \equiv B$. □

4.1.30 Exercise Prove $A \equiv B \vdash (\forall x)A \equiv (\forall x)B$ without relying on the equivalence theorem. Rather use 4.1.27 in your proof, remembering the ping-pong tautology (4.1.21). □

4.1.31 Example (A-Intro, or $\forall$-Intro) We establish here the theorem "$A \to B \vdash A \to (\forall x)B$, provided A contains no free x", or, as one often says, "there is no free x in the conclusion" (right hand side of $\vdash$).

This is proved without the Deduction theorem as we will use this result in the exercises section (4.2) *towards proving* the Deduction theorem.

1)	$A \to B$	⟨hyp⟩
2)	$(\forall x)(A \to B)$	⟨1 + Gen⟩
3)	$(\forall x)(A \to B) \to (\forall x)A \to (\forall x)B$	⟨axiom 2⟩
4)	$(\forall x)A \to (\forall x)B$	⟨2 + 3 + MP⟩
5)	$A \to (\forall x)A$	⟨axiom 3⟩
6)	$A \to (\forall x)B$	⟨4 + 5 + Post⟩ □

4.1.32 Example (Variant Theorem for $\forall$) Another useful result that practitioners use without quoting and without notice is the "bound variable renaming", which some people, uncharitably, call the "dummy renaming" theorem. In Shoenfield (1967), Tourlakis (2003a) it goes as the *variant theorem*. Suppose that the variable z is fresh for $(\forall x)A[x]$. Then we have the theorem

$$\vdash (\forall x)A[x] \equiv (\forall z)A[z]$$

The proof uses ping-pong and is straightforward, except that in the "$\leftarrow$ *Direction*" below it requires some combinatorial thinking that is not part of logic.

$\rightarrow$ *Direction*. Note that step 1 below has a legal substitution $[x \leftarrow z]$ since *freshness* of z entails that *no part* of A is "$(\forall z)(\ldots\ x\ \ldots)$" to give trouble when we do $[x \leftarrow z]$.

1)	$(\forall x)A[x] \to A[z]$	⟨axiom 2⟩
2)	$(\forall x)A[x] \to (\forall z)A[z]$	⟨1 + A-Intro (4.1.31); recall that z is fresh for $(\forall x)A$⟩

$\leftarrow$ *Direction*. Here we start the two-line proof with "$(\forall z)A[z] \to A[z][z \leftarrow x]$". We need to argue that "$A[z][z \leftarrow x]$" is the same as "$A[x]$" or just "A" in simpler notation.

First off, the rightmost substitution in $A[z][z \leftarrow x]$ is *legal*. *Why?* Because there is NO $(\forall x)(\ldots\ z\ \ldots)$-part in $A[z \leftarrow x]$ to capture x. Note that

There are NO preexisting z in A since z is fresh for $(\forall x)A$.

Thus a z could only appear in $(\forall x)(\ldots\ \square\ \ldots)$ in the spot $\square$ iff "$\square$" were a **free** x —*impossible* when such an x is in the scope of $(\forall x)$.

Indeed, we see that $A[z][z \leftarrow x]$ is $A[x \leftarrow z][z \leftarrow x]$. Now, we already noted that $A[x \leftarrow z]$ is legal. At the end of this operation we introduce the symbol "z" in *precisely those spots* where A held *originally free* occurrences of x.

But then, $\Big(A[x \leftarrow z]\Big)[z \leftarrow x]$ will change back to x precisely those z that were originally *free* x. Seeing that there were no preexisting z, *all* z *change back* to x. A is restored.

Now the $\leftarrow$ *Direction* proof.

1)	$(\forall z)A[z] \rightarrow A[z][z \leftarrow x]$	$\langle$axiom 2$\rangle$
1′)	$(\forall z)A[z] \rightarrow A[x]$	$\Big\langle$discussion above and "$\Big(A[x \leftarrow z]\Big)[z \leftarrow x]$" is "$A[x]$" conclusion$\Big\rangle$
2)	$(\forall z)A[z] \rightarrow (\forall x)A[x]$	$\langle 1' +$ A-Intro (4.1.31); recall that z is fresh hence not same as $x\rangle$

□ 

4.2 Exercises

1. *Define*: The formula A is true over the real numbers $\mathbb{R}$.
2. Let, *non standardly*, $A \vee B$ be *defined* to be true over some domain of interest just in case A is true, or B is true, or both.
   Let now A stand for "x is even" and B stand for "x is odd", x varying over $\mathbb{N}$. Is $A \vee B$ true according to the above *non standard* definition?
3. a. Show through a general (syntactic) proof that $x < y \vdash y < x$ ($<$ is an uninterpreted relation; the choice of symbol here is meant to provoke)
   b. Show that $\nvdash x < y \rightarrow y < x$ (*Hint*: Show that $x < y \rightarrow y < x$ cannot be possibly true over, say, the real numbers.)
   c. Does this invalidate the deduction theorem? Explain.
4. Prove Proposition 4.1.14.
5. Prove Proposition 4.1.16.
6. ($\exists$-*Introduction*) Prove that if x is not free in the *conclusion*, then $A \rightarrow B \vdash (\exists x)A \rightarrow B$. This derived rule is also called "L$\exists$", since $\exists$ is introduced to the left. *Hint*. Use axiom 7 in conjunction with 4.1.31.
7. (*Variant theorem for* $\exists$) Suppose that the variable z is fresh for $(\exists x)A[x]$. Then we have the theorem

$$\vdash (\exists x)A[x] \equiv (\exists z)A[z]$$

8. (*The Deduction Theorem*) The reader is asked here to prove the Deduction Theorem:

If A is a *sentence* or *closed* —meaning it has no free variables, then from

$$\Gamma, A \vdash B \tag{1}$$

follows

$$\Gamma \vdash A \to B$$

Hints. For the proof.
Do *induction* on the length of the proof indicated by (1). (You may want to come back to this after you study the next chapter; but only if induction intimidates you.)
For the *Basis* we have a proof of length one, so it contains only B, in which case we have subcases

(i) B is an axiom. Then $\Gamma \vdash B$ (justify!) and *hence*, $\Gamma \vdash A \to B$ (justify "hence"!)
(ii) B is in Γ. As in item (i).
(iii) B is A ("Γ, A" is an alias of $\Gamma \cup \{A\}$ or "$\Gamma + A$". Thus it is the "whole hyp" in (1)). $A \to B$ is thus the tautology $A \to A$ hence an axiom in group 1. Why should you conclude $\Gamma \vdash A \to A$, that is, $\Gamma \vdash A \to B$?

For (*Induction Hypothesis*) *I.H.* fix an n and *assume* all $\Gamma + A$-proofs of lengths $\leq n$ satisfy the theorem.
We now embark discussing a $\Gamma + A$-proof of length $n + 1$.

$$A_1, A_2, \ldots, A_j, \ldots, A_k \to B, \ldots, A_n, B \tag{1}$$

Cases for B:

a. B is placed because it is in Γ, or is A or is a logical axiom. No work needed as all these cases were discussed in the *Basis*.
b. B was obtained by MP from two *previous* formulas in (1), say A_k and $A_k \to B$. By I.H. we have

$$\Gamma \vdash A \to A_k \tag{2}$$

$$\Gamma \vdash A \to (A_k \to B) \tag{3}$$

See if you can now prove

$$\Gamma \vdash A \to B$$

c. B was obtained by Gen from one *previous* formula in (1), say A_j, that is, B is $(\forall x) A_j$. By the I.H. we have

$$\Gamma \vdash A \to A_j \tag{4}$$

See if you can prove that $G \vdash A \to B$, which is the *same as* $G \vdash A \to (\forall x)A_j$.

End of Hints.

9. (*The Deduction Theorem version 2*) The reader is asked here to prove this version of the Deduction Theorem:

> Suppose that
>
> $$\Gamma, A \vdash B \tag{1}$$
>
> and that there is a *proof* of this fact that treated *all the free variables of A as constants*, meaning, if x is such a variable, then we never used it in said proof with $(\forall x)$ nor with $[x \leftarrow t]$. Under these conditions prove that we have
>
> $$\Gamma \vdash A \to B$$

10. (*Proof by Contradiction*) Let us (somewhat informally) consider, as we did before, the truth values **f** and **t** as "*constant*" *atomic* (i.e., devoid of connectives hence of Boolean structure) formulas.
We then state the principle of proof by contradiction as "to prove $\Gamma \vdash A$, where A has no free variable, is the same as proving a falsehood, such as **f**, from premises $\Gamma + \neg A$".
Thus, *prove* that, for a *sentence* A, we have $\Gamma \vdash A$ *iff* $\Gamma, \neg A \vdash \mathbf{f}$.
Hint. In the *if-direction* use the deduction theorem to obtain $\Gamma \vdash \neg A \to \mathbf{f}$. Follow up with an application of Post.
In the *only if-direction* use 4.1.16 to show $\Gamma, \neg A \vdash A$ and then the definition of proof to also show $\Gamma, \neg A \vdash \neg A$. Follow up these two with an application of Post.

11. (*Proof by Auxiliary Hypothesis, or* $\exists$ *-Elimination*) See also 4.1.24 Suppose that $\Gamma \vdash (\exists x)A$ and let z be *fresh* for $(\exists x)A$ and B. Then

> If $\Gamma, A[x \leftarrow z] \vdash B$, we will also have $\Gamma \vdash B$

Hint. The assumption is that we have a proof with the help of the auxiliary hypothesis,

$$\Gamma, \overbrace{A[x \leftarrow z]}^{aux.\ hyp} \vdash B$$

Treating all the variables (incl. z) of the auxiliary hypothesis as constants we apply the deduction theorem to get $\Gamma \vdash A[x \leftarrow z] \to B$.

By 4.2.6 obtain $\Gamma \vdash (\exists z)A[z] \to B$. Then $\vdash (\exists x)A[x] \equiv (\exists z)A[z]$ and the previous yield $\Gamma \vdash (\exists x)A[x] \to B$ by Post. Now use one of the assumptions and Post to get $\Gamma \vdash B$.

12. Prove by $\exists$-elimination that $\vdash (\exists x)\Big((A \vee B) \to C\Big) \to (\exists x)(A \to C) \wedge (\exists x)(B \to C)$.

13. Find a proof other than via $\exists$-elimination for the above.

14. Prove by $\exists$-elimination that $\vdash (\exists x)\Big((A \vee B) \to C\Big) \to (\exists x)\Big((A \to C) \wedge (B \to C)\Big)$.

15. Find a proof other than via $\exists$-elimination for the above.

16. Prove by $\exists$-elimination that $\vdash (\exists x)\Big((A \to B) \wedge (A \to C)\Big) \to (\exists x)\Big(A \to B \wedge C\Big)$.

17. Find a proof other than via $\exists$-elimination for the above.

18. Prove by $\exists$-elimination that $\vdash (\forall x)A \to (\exists x)(A \to B) \to (\exists x)B$.

19. Prove by $\exists$-elimination that $\vdash (\exists x)A \to (\forall x)(A \to B) \to (\exists x)B$.

20. Find a proof other than via $\exists$-elimination for the above.

21. Let ϕ stand for an unspecified relation of *two* variables.
This could be anything like: $<, >, \leq, \geq, \in, \ni, \subset$ and many others!
Prove (1) within *pure logic*, that is, logic without any theory-specific axioms, no math hypotheses, and no special meaning for symbols. *Anyway*, "symbols" —other than *logical* symbols, that is, *connectives*, *brackets* and $=$— *never have any inherent meaning* relating to their shape *if* **there are no axioms** *about said symbols*!

$$\vdash \neg(\exists y)(\forall x)\Big(\phi(x, y) \equiv \neg\phi(x, x)\Big) \tag{1}$$

Hint. Prove (1) via a proof by contradiction. In the initial setup you end up with a formula, which has a leading $(\exists y)$ and it can prove **f**. Now apply $\exists$-elimination to construct the latter proof.

22. What just happened in 21 above? Contextualise within set theory and discuss.

Induction

5

Overview

This chapter is about two of the most important topics in a course on discrete mathematics —*induction* and *inductive definitions*. Nowadays most authors prefer to call "inductive" definitions "recursive". These topics are called upon in numerous sequel courses such as *logic*, *data structures*, *theory of computation*, *design and analysis of algorithms*.

In this chapter we introduce the *induction* (proof) principle on $\mathbb{N}$ as an *equivalent* principle to the *least (integer) principle* on $\mathbb{N}$.

But we also generalise induction *in two important directions* making this tool sophisticated enough to be applicable to advanced readings, for example (axiomatic) set theory, which is relevant to mathematics students: One direction is to recognise that the induction principle (equivalently, the *minimal condition* or principle, MC, which is a generalisation of the least principle on $\mathbb{N}$) which on $\mathbb{N}$ is an attribute of the "natural" *order* $<$, can be extended to *arbitrary orders* on arbitrary classes. This opens applicability of induction to any classes that are equipped with *an order that has MC*.

The other direction of our generalisation is to recognise that a relation —whether we denote it by "$\mathbb{P}$" or "$<$"— does *not have to be an order* for us to do induction along it. All it needs is to satisfy MC. For example, we can do induction along $\in$ —this is *not* an order on all sets (it fails transitivity)— to prove properties of classes!

The chapter concludes with the important topic of *inductive* or *recursive* definitions of *functions*, such as, for example, the recursive definition of the *factorial* function $0! = 1$ and $(n+1)! = (n+1) \times (n!)$, for $n \geq 0$. Such inductive definitions are central in the theory of computation, and in *practical* computation in fact, since it turns out that in practical computation we cannot compute functions beyond the so-called primitive recursive functions; cannot even compute *all* primitive recursive functions (except only "in principle") due to

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0_5

the fact that "most of them" have astronomical *outputs* —like the function that outputs the "ladder" of x 2s on input x below[1]— and equally astronomical *run times* needed for their computation. For example it can trivially be proved that the function that with input x outputs the following number is primitive recursive.

$$\left.\begin{matrix} 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \end{matrix}\right\} x \text{ 2s}$$

What is the connection between primitive recursive functions and recursive definitions?

Primitive recursive functions (cf. e.g., Tourlakis (2022)) are formed by starting from trivial functions, such as the function that for all inputs returns zero, using repeatedly compositions and so-called "primitive" *recursive definitions*. A primitive recursive definition is a special simple form of a general inductive definition where f is defined from functions h and g by the two equations below, valid for all x, y from $\mathbb{N}$,

$$\begin{aligned} f(0, y) &= h(y) \\ f(x+1, y) &= g(x, y, f(x, y)) \end{aligned}$$

Definitions just as the above are based on the fact that $\mathbb{N}$ supports induction along the standard "$<$". We will not omit a generalisation of recursive definitions along *any* relation $\mathbb{P}$ that may have MC *without* being an order. As an example, taking $\mathbb{P} = \in$ we give an inductive definition over the class $\mathbb{U}$ of the so-called "support" function that for any set A as input returns the set of all the atoms used to build A. For example, if $A = \{\{\{2\}\}, \{1\}\}$, it returns $\{2, 1\}$.

5.1 Inductiveness Condition (IC)

In Remark 3.4.29 we concluded with a formulation of the *minimal condition* (MC) for *any order* $<$, for which fields (left/right) *have not* been specified, an *unrelativised* order, that is. We did this as follows:

> *That an "order $<$ has MC" is* captured *by —i.e., is equivalent to— the statement*
> *For any "property", that is, formula $F[x]$[2] we have that the following is true*

$$(\exists a) F[a] \to (\exists a)\Big(F[a] \wedge \neg(\exists y)\big(y < a \wedge F[y]\big)\Big) \qquad (\dagger)$$

More generally, for a *non-order relation* $\mathbb{P}$ we saw that "$\mathbb{P}$ has MC" (see also 3.4.30 for the concept "a is $\mathbb{P}$-minimal in $\mathbb{A}$") is expressed in terms of classes (3.4.32, (see $(1')$)) as

[1] $2^{2^2} = 16$ but $2^{2^{2^2}} = 65536$ while $2^{2^{2^{2^2}}}$ is astronomical.

[2] Recall that this notation, *square brackets*, indicates our interest in *one* among the, possibly many, free variables of F.

$$\mathbb{A} \neq \emptyset \to (\exists a \in \mathbb{A})\mathbb{A} \cap (a)\mathbb{P}^{-1} = \emptyset \tag{1'}$$

while in terms of "properties" $F[x]$ it is expressed by 3.4.32, (see (2′)), reproduced below.

$$(\exists a)F(a) \to (\exists a)\Big(F(a) \wedge \neg(\exists y)\big(y\mathbb{P}a \wedge F(y)\big)\Big) \tag{2'}$$

which formally looks exactly like (†) above but using the symbol "$\mathbb{P}$" instead of "$<$".

> Thus, (†) and (2′) are *formally* (in form!) *identical*, but semantically we will need to recall that "$<$" represents any **order** with MC in (†) while "$\mathbb{P}$ represents any **relation** with MC" in (2′) (and (1′)).

Let us logically manipulate (2′) to bring it into an equivalent form that goes under the nickname "Inductiveness Condition" —in short IC— or, alternatively, is called the "Principle of Induction."

Let us rewrite (2′) replacing $F[x]$ by $\neg G[x]$ everywhere, where $G[x]$ is arbitrary. We get the theorem

$$(\exists a)\neg G[a] \to (\exists a)\Big(\neg G[a] \wedge \neg(\exists y)\big(y\mathbb{P}a \wedge \neg G[y]\big)\Big) \tag{2}$$

Using the equivalence theorem (p. 137) and Axiom 7 (p. 125), we obtain from (2)

$$\neg(\forall a)G[x] \to \neg(\forall a)\neg\Big(\neg G[a] \wedge (\forall y)\neg\big(y\mathbb{P}a \wedge \neg G[y]\big)\Big)$$

and then —the tautology $(X \to Y) \equiv (\neg Y \to \neg X)$ known as "contrapositive" is used— also

$$(\forall a)\neg\Big(\neg G[a] \wedge (\forall y)\neg\big(y\mathbb{P}a \wedge \neg G[y]\big)\Big) \to (\forall a)G[a]$$

Using the tautology

$$\neg(A \wedge B) \equiv \neg A \vee \neg B$$

and the equivalence theorem, we transform the above to this theorem:

$$(\forall a)\Big(G[a] \vee \neg(\forall y)\big(\neg y\mathbb{P}a \vee G[y]\big)\Big) \to (\forall a)G[a]$$

Again, this time using the tautology

$$\neg A \vee B \equiv A \to B$$

(twice) and the equivalence theorem, we transform the above to this theorem:

$$(\forall a)\Big((\forall y)\big(y\mathbb{P}a \to G[y]\big) \to G[a]\Big) \to (\forall a)G[a] \tag{3}$$

Display (3) above *expresses* the *Inductiveness Condition* (IC) for $\mathbb{P}$, or, as we usually say, expresses the *principle* of *strong induction*, or *complete induction*, or *course-of-values induction* for the —*not necessarily an order*— relation $\mathbb{P}$.

5.1.1 Remark The above method of showing the equivalence between MC and IC is not mentioned much in the literature (see however Barwise (1975) who applies it in the case where $\mathbb{P}$ is $\in$ on $\mathbb{U}$). □

We should state an obvious and trivial corollary.

5.1.2 Corollary *An order $<$ has MC iff it has IC.*

Proof We never relied on whether or not $\mathbb{P}$ is an order, so the preceding equivalence of MC and IC proof holds if $\mathbb{P}$ is, in particular, an order $<$. In fact, a proof for an order $<$ is obtained from the above simply by replacing $\mathbb{P}$, everywhere in the reasoning, by $<$. □

If we replace, everywhere in (3), the formula $G[y]$ by the class $\mathbb{B} \overset{Def}{=} \{y : G[y]\}$, then we directly obtain (3′) below from (3):

$$(\forall a)\Big((a)\mathbb{P}^{-1} \subseteq \mathbb{B} \to a \in \mathbb{B}\Big) \to (\forall a)a \in \mathbb{B} \tag{3′}$$

Note that the y in $(a)\mathbb{P}^{-1}$ are *the* $\mathbb{P}$*-predecessors* of a in the sense that they are precisely those y that satisfy $y\mathbb{P}a$ —"y is *before* a along $\mathbb{P}$".

It is extremely useful to state (3′) in words.

> If we want to prove that all a are in some class $\mathbb{B}$ and we have a relation $\mathbb{P}$ with IC (equivalently MC), then it *suffices* to prove, *for any arbitrary unspecified* a, that a is in $\mathbb{B}$ *provided all* its $\mathbb{P}$*-predecessors* are.

The part "*for any arbitrary unspecified* a" is English for the part $(\forall a)$. For such an a we prove $a \in \mathbb{B}$ with the *help* of the *condition* (assumption) $\boxed{(a)\mathbb{P}^{-1} \subseteq \mathbb{B}}$. The last implication in (3′) says that "*for any arbitrary unspecified* a, a *is in* $\mathbb{B}$, **unconditionally**".

The boxed formula above is called the *Induction Hypothesis* or *I.H.* for a —and so is the corresponding part "$(\forall y)\big(y\mathbb{P}a \to G[y]\big)$" in (3).

The essence of the I.H. in either formulation —(3) or (3′)— is that it *assists* in the proof of the leftmost (conditional) "$a \in \mathbb{B}$" (or "$G[a]$") for the "arbitrary unspecified a". Having proved $a \in \mathbb{B}$ *under the I.H.*, the fact that our $\mathbb{P}$ has IC *also implies* (the last implication in (3) or (3′)) the **unconditional** truth of $a \in \mathbb{B}$ for the arbitrary a.

Thus the rightmost (unconditional) "$a \in \mathbb{B}$" is established for all *a only* from the axioms and assumptions of the theory we are working in (e.g., set theory, number theory, etc) and the I.H. *drops out from the hypotheses list*.

Of course, $(\forall a)a \in \mathbb{B}$ implies $\mathbb{U} \subseteq \mathbb{B}$, hence $\mathbb{B} = \mathbb{U}$, the set theoretic class of "all things". This is fine for (informal) set theory, indeed useful, but we often work within much smaller than $\mathbb{U}$ classes $\mathbb{A}$.

For example in number theory we work in $\mathbb{N}$. Then the classes $\mathbb{B}$ *of interest* will be *subsets* of $\mathbb{N}$. Therefore let us formulate IC *relativised to a class* or set $\mathbb{A}$ before we move on to practical considerations and examples.

We reproduce here the formula (†) that says "$\mathbb{P}$ has MC *relative to a class* $\mathbb{A}$" (cf. 3.4.37) —which is the same as "$\mathbb{P} \mid \mathbb{A}$ has MC" (Definition 3.4.34):

$$(\exists b \in \mathbb{A})F[b] \to (\exists b \in \mathbb{A})\Big(F[b] \wedge \neg(\exists x \in \mathbb{A})\big(F[x] \wedge x\mathbb{P}b\big)\Big) \tag{†}$$

Expressing (†) in terms of the formula $\neg G[x]$ instead of $F[x]$ we obtain

$$(\exists b \in \mathbb{A})\neg G[b] \to (\exists b \in \mathbb{A})\Big(\neg G[b] \wedge \neg(\exists x \in \mathbb{A})\big(\neg G[x] \wedge x\mathbb{P}b\big)\Big) \tag{†$'$}$$

Applying the very same transformations we introduced on p. 145 to (†$'$) we obtain the equivalent formula (‡) below

$$(\forall b \in \mathbb{A})\Big((\forall x \in \mathbb{A})\big(x\mathbb{P}b \to G[x]\big) \to G[b]\Big) \to (\forall b \in \mathbb{A})G[b] \tag{‡}$$

If we replace, everywhere in (‡), the formula $G[y]$ by the class $\mathbb{B} \overset{Def}{=} \{y \in \mathbb{A} : G[y]\}$, then we directly obtain (‡$'$) below from (‡):

$$(\forall b \in \mathbb{A})\Big((b)\Big(\mathbb{P}^{-1} \mid \mathbb{A}\Big) \subseteq \mathbb{B} \to b \in \mathbb{B}\Big) \to (\forall b \in \mathbb{A})b \in \mathbb{B} \tag{‡$'$}$$

or, by 3.4.31 2. (cf. also 3.4.5),

$$(\forall b \in \mathbb{A})\Big(\mathbb{A} \cap (b)\mathbb{P}^{-1} \subseteq \mathbb{B} \to b \in \mathbb{B}\Big) \to (\forall b \in \mathbb{A})b \in \mathbb{B} \tag{¶}$$

Let us render (‡) more recognisable: By applying MP (*modus ponens*, cf. rule (*MP*) on p. 125) I can transform (‡) in "rule of inference form", indeed I will write it like a rule that says, like all rules do, "*if you proved my numerator, then my denominator is also proved!*"

$$\frac{(\forall b \in \mathbb{A})\Big((\forall x \in \mathbb{A})\big(x\mathbb{P}b \to G[x]\big) \to G[b]\Big)}{(\forall b \in \mathbb{A})G[b]}$$

Dropping the $(\forall b \in \mathbb{A})$-*prefix*[3] we have the rule in the form:

$$\frac{\overbrace{(\forall x \in \mathbb{A})\big(x\mathbb{P}b \to G[x]\big)}^{I.H.}\overbrace{\to G[b]}^{I.S.}}{G[b]} \qquad (CVI)$$

"CVI" for Course-of-Values Induction. CVI says

To prove $G[b]$ (for all $b \in \mathbb{A}$ is implied!) **do as follows**:

Step (a) Fix an **arbitrary** b-value. Now, **assume** $(\forall y \in \mathbb{A})\big(y\mathbb{P}b \to G[y]\big)$ for all said y. We call

the assumption *Induction Hypothesis* (for y), in short, *I.H.*

Step (b) Next, using the I.H. —and the axioms / assumptions of the theory you are working in— **prove** $G[b]$, for the same fixed unspecified b. This proof step we call the *Induction Step* or *I.S.*

Note that what is described by (a) and (b) is precisely an application of the Deduction theorem towards proving "If, for all $x\mathbb{P}b$,[4] $G[x]$ is true, then $G[b]$ is true", that is, **proving the implication on the numerator of CVI for any given** b.

Step (c) If you have done **Step** (a) and **Step** (b) above, then you **have proved** $G[x]$ (for all $x \in \mathbb{A}$ is implied!)

Important. **Step** (a) above says "**arbitrary** b".

So, I should not *leave any b-value out of the proof!*

The case where b is $\mathbb{P}$-minimal in $\mathbb{A}$ is singular (no pun intended). How do I prove the I.S. when there is no x below b along $\mathbb{P}$ in $\mathbb{A}$? There is no I.H. to rely on. No problem: The numerator implication in CVI now reads

$$(\forall y \in \mathbb{A})\Big(\overbrace{\overbrace{y\mathbb{P}b}^{\mathbf{f}} \to G[y]}^{\mathbf{t}}\Big) \to G[b]$$

The lhs of the second "$\to$" is true. Thus, to certify the truth of that *implication I must prove* $G[b]$ *without I.H. help*.

This step was hidden in **Steps** (a) – (b) above. It is called the **Basis** of the induction.

[3] In Chapter 4 we noted that $A[x]$ is true iff $(\forall x)A[x]$ is true.

[4] x in the class of interest $\mathbb{A}$.

5.2 IC Over $\mathbb{N}$

With the above general considerations in hand, the present section focuses in some practise with induction over $\mathbb{N}$.

Taking $\mathbb{P}$ here to be the order $<$ restricted to $\mathbb{N}$ and taking for granted that $< \mid \mathbb{N}$ has MC (as we argued *informally* in 3.3.14; but see also the counterpoint in  -delimited comments on p. 151 !) and we conclude that $< \mid \mathbb{N}$ also has IC.

Thus we have, for some arbitrary property $P[y]$ of natural numbers, the special form of (‡) below:

$$(\forall n \in \mathbb{N})\Big((\forall k \in \mathbb{N})\big(k < n \to P[k]\big) \to P[n]\Big) \to (\forall n \in \mathbb{N})\, P[n]$$

Dropping the $\forall$-*prefix* we have the above in "rule form":

$$\frac{\overbrace{(\forall k \in \mathbb{N})\big(k < n \to P[k]\big)}^{I.H.} \overbrace{\to P[n]}^{I.S.}}{P[n]} \qquad (CVI \text{ on } \mathbb{N})$$

5.2.1 Remark Of course, we have the proof technique we called CVI (after Kleene) for any POset $(A, <)$ where $<$ has IC (equivalently MC) over A, that is, $< \mid A$ has IC. □

There is another simpler induction principle over $\mathbb{N}$ that we call, well, *simple* induction:

$$\frac{P[0],\ P[x] \to P[x+1]}{P[x]} \qquad (SI)$$

"SI" above stands for Simple Induction. That is, to prove $P[x]$ for all x (denominator) do *three* things:

Step 1. Prove/verify $P[0]$

Step 2. **Assume** $P[x]$ for fixed ("frozen") x (unspecified!).

Step 3. **prove** $P[x+1]$ for that same x. The assumption is the I.H. for simple induction. The I.S. is the step that proves $P[x+1]$.


Note that what is described here is precisely an application of the Deduction theorem towards proving "$P[x] \to P[x+1]$", that is, **proving the implication for any given** x.

Step 4. If you have done **Step** 1. through **Step** 3. above, then you **have proved** $P[x]$ (for all x in $\mathbb{N}$ is implied!)

Is the principle SI *correct*? I.e., if I do all that the numerator of SI asks me to do (or **Steps** 1. – 3.), then do I *really* get that the denominator is true (for all x implied)?

5.2.2 Theorem *The validity of SI is a consequence of MC on* $\mathbb{N}$.

Proof Suppose SI is *not* correct. Then, for some property $P[x]$, despite having completed **Steps** 1. – 3., still, $P[x]$ is *not true* for all x!

Well, if so, then by MC let $n \in \mathbb{N}$ be *smallest* such that $P[n]$ is *false*. Now, $n > 0$ since I *did* verify the truth of $P[0]$ (**Step** 1.). Thus, $n - 1 \geq 0$. But then, when I proved "$P[x] \rightarrow P[x+1]$ for all x (in $\mathbb{N}$)" —in **Steps** 2. and 3.— this includes **proving** the case

$$P[n-1] \rightarrow P[n] \tag{4}$$

Now, by the smallest-ness of n, $P[n-1]$ is *true*, hence $P[n]$ is true by (4) and the truth table of "$\rightarrow$". I have just got a contradiction! I conclude that no such smallest n exists, i.e., $P[x]$ is true (for all $x \in \mathbb{N}$).

We conclude that SI works —if MC does (cf. discussion in the  -passage on p. 151). □

How do the simple and course-of-values induction relate? They are equivalent tools, or, *they have the same (proof) power* as we say. Here is why:

5.2.3 Theorem *From the validity of SI I can obtain the validity of MC.*

Proof Suppose $< \mid \mathbb{N}$ has SI.

We prove it *also* has MC. *Suppose not*.

Then there is a set

$$\emptyset \neq S \subseteq \mathbb{N} \tag{1}$$

that has *no* minimal (same as *least*, since $< \mid \mathbb{N}$ is a total order) element.

Let us call T the set $\mathbb{N} - S$. I will use SI and prove that $T = \mathbb{N}$. The property I am proving for all $n \in \mathbb{N}$ using SI is

$$\{0, 1, \ldots, n\} \subseteq T \tag{2}$$

Basis. $\{0\} \subseteq T$, that is, $0 \in T$. This is so, because otherwise $0 \in S$ contradicting that S *has no least element*.

Now fix an unspecified n and take (2) as the I.H.

I prove next the I.S. that

$$\{0, 1, \ldots, n, n+1\} \subseteq T \tag{3}$$

Towards (3) —given the I.H.— I need to show $n+1 \in T$. Suppose this is not true. But then $n+1 \in S$ and the I.H. implies that none of $0, 1, \ldots, n$ is in S. This means $n+1$ is minimal in S, a contradiction.

Having shown (2), for all $n \in \mathbb{N}$, we have $\mathbb{N} \subseteq T$ hence $T = \mathbb{N}$ and thus $S = \emptyset$. A contradiction to (1). Done. □

5.2.4 Corollary *All three of SI, CVI and MC are equivalent.*

Proof For $< \mid \mathbb{N}$ we have CVI iff we have MC iff we have SI. □

5.2.5 Remark

1. When do I use CVI and when SI? SI is best to use when to prove $P[x]$ (in the I.S.) I only need to know $P[x-1]$ is true. CVI is used when we need a more flexible I.H. that $P[n]$ is true for all $n < x$. See the examples below!
2. "0" is the boundary case if the claim we are proving is valid "for all $n \in \mathbb{N}$", or simply put, "for $n \geq 0$". If the claim is "for all $n \geq a$, $P[n]$ is true" then usually $P[n]$ is meaningless for $x < a$ and thus *the Basis is for* $n = a$. □

Having established that MC and CVI (and SI) are equivalent for the "standard" order $<$, it follows that over $\mathbb{N}$ we have *both* or we have *none*. **Which one is it?**

We have *informally* argued earlier (e.g., in the section on congruences, 3.3.14) that $<$ *does* have MC over $\mathbb{N}$ but the informal argument we gave there does not have the force of a proof.

A mathematical proof requires that *established* properties of natural numbers and the set $\mathbb{N}$ be *known* and *used*. We only offered a tentative argument within informal set theory *where we took for granted* that $\mathbb{N}$ is one of this theory's sets. If so, *what properties does this set have?*

The proper way to prove *within set theory* that $\mathbb{N}$ has CVI or equivalently MC is to *build* a *copy* of $\mathbb{N}$ within set theory and prove such properties as theorems.

Indeed this *can* be done (e.g., in Tourlakis (2003b)) but we did not do it here. One *builds* a counterpart of $\mathbb{N}$ and gives it an alternative name —ω— as follows: "0" is defined to be the object $\emptyset$. If we defined the number n as a *set*, then its successor, $n+1$, is defined as the *set* $n \cup \{n\}$. Thus $n+1$ stands for $\{0, 1, 2, \ldots, n\}$. We then prove that the class of all so *constructed* natural numbers is a set, and call it ω.

Next we prove that $<$ on ω *defined* by

$$n < m \overset{Def}{\text{ iff }} n \subsetneqq m$$

satisfies MC and hence also both CVI and SI.

More elegantly, one may axiomatise $\mathbb{N}$ *outside* set theory, via the *Peano's axioms*. One such axiom schema states that the "$<$" on the set of natural numbers —with its basic properties axiomatically postulated— satisfies simple induction.

Hm. But we have seen arguments that directly "prove" simple induction "works" employing a "falling dominoes" argument. Haven't we? It goes like this:

We are equipped with a proof that

$$P[n] \text{ implies } P[n+1] \tag{1}$$

for the *arbitrary* n. We also verify that $P[0]$ is true.

Thus the argument

$$P[0] \text{ and } P[0] \to P[1] \text{ yield } P[1]; \text{ next } P[1] \text{ and } P[1] \to P[2] \text{ yield } P[2]; \text{ next } \ldots \text{ next } P[n] \text{ and } P[n] \to P[n+1] \text{ yield } P[n+1]; \ldots \tag{2}$$

proves $P[n+1]$ is true for no matter what n.

That is, we can prove $P[x]$ for any natural number x, the argument x being reachable by adding *one* repeatedly. However, this does *not* say that we proved $(\forall x)\, P[x]$.

One, a proof has finite length and we cannot extend the proof (2) by just repeating the parts "$P[n]$ and $P[n] \to P[n+1]$ yield $P[n+1]$" *an infinite number of times*.

Two, nor can we be sure that *all Informal* natural numbers can be reached from 0 by just repeatedly adding one. What *are* the natural numbers? Unless we know more about the natural numbers we cannot be sure that, for example, there are no "*infinite* natural numbers" *after the end of* the sequence $0, 1, 2, 3, \ldots, n, \ldots$ The subclass $\mathbb{I}$ of $\mathbb{N}$ consisting of only "infinite numbers" has no least member, since if X is an infinite natural number, then so is $X - 1$.

In particular, such an observation invalidates the argument in support of the thesis that $<$ on $\mathbb{N}$ has MC that we offered in 3.3.14, namely,

> But we *cannot* have an infinite descending sequence of *nonnegative* integers
>
> $$\ldots < x''' < x'' < x' < x$$

The boxed statement is false if your descent downwards along natural numbers starts from an *infinite* integer. We need to be able to prove that infinite walks downwards are impossible —that is, infinite natural numbers do not exist— and accepting IC as an axiom (as we do in Peano arithmetic) or, equivalently, *postulating* MC, is *one elegant way* to show such an impossibility.

So, we accept *one* of MC, IC (CVI) or SI *axiomatically*.

The discussion above triggers the motivation to connect the non existence of infinite downwards walks with the presence of MC (or IC).

5.2.1 Well-Foundedness

5.2.6 Definition For any relation $\mathbb{P}$, an *infinite descending* $\mathbb{P}$*-chain* is a function f with the properties

(1) $\mathrm{dom}(f) = \mathbb{N}$, and
(2) $(\forall n \in \mathbb{N})\, f(n+1)\, \mathbb{P}\, f(n)$. □

Intuitively, an infinite descending $\mathbb{P}$-chain is an infinite sequence $a_0, a_1, \ldots$ such that $\ldots a_3 \mathbb{P} a_2 \mathbb{P} a_1 \mathbb{P} a_0$ (that is, $a_{n+1} \mathbb{P} a_n$, for all $n \geq 0$).

5.2.7 Definition A relation $\mathbb{P}$ is *well-founded* iff it has *no* infinite descending chains.

$\mathbb{P}$ is *well-founded over* $\mathbb{A}$ iff $\mathbb{P} \mid \mathbb{A}$ is well-founded. □

Intuitively, $\mathbb{P}$ is well-founded if the universe of all sets and atoms $\mathbb{U}$ cannot contain an infinite descending chain, while it is well-founded *over* $\mathbb{A}$ if $\mathbb{A}$ cannot contain an infinite descending $\mathbb{P}$-chain. Clearly, no infinite descending $\mathbb{P}$-chain can "start" anywhere outside $\mathrm{ran}(\mathbb{P})$ in any case.

There is some disagreement on the term "well-founded" in the literature. In some of the literature it applies *definitionally* to what we have called relations "with MC". However, in the presence of AC well-founded relations are precisely those that have MC, so the slight confusion —if any— is harmless.

5.2.8 Theorem *If a relation* $\mathbb{P}$ *has MC over* $\mathbb{A}$*, then* $\mathbb{P}$ *is well-founded over* $\mathbb{A}$*.*

Proof Let instead f be an infinite descending $\mathbb{P}|\mathbb{A}$-chain.

Then $\emptyset \neq \mathrm{ran}(f) \subseteq \mathbb{A}$ hence there is an $a \in \mathrm{ran}(f)$ which is $\mathbb{P}|\mathbb{A}$-minimal.

Now, $a = f(n)$ for some $n \in \mathbb{N}$, but $f(n+1)(\mathbb{P}|\mathbb{A})\, f(n)$ contradicting the $\mathbb{P}|\mathbb{A}$-minimality of a. □

5.2.9 Corollary *Let A be set. Then the following are equivalent:*

(1) $\mathbb{P}$ *has MC over A.*
(2) $\mathbb{P}$ *has IC over A.*
(3) $\mathbb{P}$ *is well-founded over A.*

Proof The equivalence of (1) and (2) as well as (1)$\Longrightarrow$(3) have already been proved. Thus we only need to prove that (3) implies (1). So assume (3) and let (1) fail. Let $\emptyset \neq B \subseteq A$ such that B has no $\mathbb{P}$-minimal elements. Pick an $a \in B$. Since it cannot be $\mathbb{P}$-minimal, pick an $a_1 \in B$ such that $a_1 \mathbb{P} a$. Since a_1 cannot be $\mathbb{P}$-minimal, pick an $a_2 \in B$ such that $a_2 \mathbb{P} a_1$.

This process can continue *ad infinitum* to yield an infinite descending chain $\ldots a_3 \mathbb{P} a_2 \mathbb{P} a_1 \mathbb{P} a$ in A, contradicting (3). Done.

This argument used AC, and more precisely it goes like this:

Let g be a choice function for $2^B - \{\emptyset\}$, that is, for each $S \in 2^B - \{\emptyset\}$, we have $g(S) \in S$ (cf. 3.5.28).

Define now f on $\mathbb{N}$ as

$$f(n) = \begin{cases} a & \textbf{if } n = 0 \\ g\big(B \cap (f(n-1))\mathbb{P}^{-1}\big) & \textbf{if } n > 0 \end{cases}$$

f is total on $\mathbb{N}$ for $B \cap (f(n-1))\mathbb{P}^{-1} \neq \emptyset$ for all $n > 0$, by assumption. By $g(x) \in x$, for all $x \in 2^B - \{\emptyset\}$, we get $f(n) \in B \cap (f(n-1))\mathbb{P}^{-1}$, i.e., $f(n) \mathbb{P} f(n-1)$ and $f(n) \in B$, for all $n > 0$, thus f is an infinite descending chain in $B \subseteq A$.

□

5.2.10 Remark The implication (3)$\Longrightarrow$(1) and hence the entire corollary goes through for any classes $\mathbb{A}$ and $\emptyset \neq \mathbb{B} \subseteq \mathbb{A}$ as long as $\mathbb{P}$ is *left-narrow* as we say, that is, the class $\{x : x \mathbb{P} a\} = (a)\mathbb{P}^{-1}$ is a set for all $a \in \mathbb{A}$.

Indeed, the part "let $a \in \mathbb{B}$" needs no elaboration, and moreover all $\mathbb{B} \cap (f(n-1))\mathbb{P}^{-1}$ are sets by left-narrowness. □

5.2.11 Definition **(λ-notation)** λ-notation is very useful in both discrete mathematics and in the theory of computation (Tourlakis (2012, 2022)). It easily allows us to separate the *intensional notation* of a function —i.e., what it does on any given input— as opposed to its *extensional notation*, that is, the function as a possibly infinite table of input/output pairs.

The format of λ-notation is

$$\overset{\text{begin input}}{\underset{\downarrow}{\boldsymbol{\lambda}}} \quad \text{list of inputs} \quad \overset{\text{end input}}{\underset{\downarrow}{\boldsymbol{.}}} \quad \text{rule for how to obtain the output}$$

Examples:

1. $\lambda x.x + 1$.
2. $\lambda xy.x - y$.
3. $\lambda xy.x + 42$. In this example the input y is ignored. Its value is not used to compute the output. □

5.2.12 Example If $\mathbb{P}$ is well-founded, then it is irreflexive.

Indeed, if $a\mathbb{P}a$ for some a, then the function $\lambda n.a$ on $\mathbb{N}$ is an infinite descending chain $(\ldots a\mathbb{P}a\mathbb{P}a\mathbb{P}a)$.

By Corollary 5.2.9, if $\mathbb{P}$ has IC (equivalently MC) then it is irreflexive.

If $\mathbb{P}$ is irreflexive but *not* well-founded, is then $\mathbb{P}^+$ a partial order? (A legitimate question since $\mathbb{P}^+$ is transitive.)

Well, no, for consider $R = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 1\rangle\}$ which is irreflexive. It is not well-founded: for example, we have an infinite descending chain

$$\ldots R1R2R3R1R2R3R1R2R3R1$$

Now $R^+ = \{\langle 1, 1\rangle, \langle 2, 2\rangle, \langle 3, 3\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 1\rangle, \langle 1, 3\rangle, \langle 2, 1\rangle, \langle 3, 2\rangle\}$ which is *not* a partial order (it is reflexive), *nor is it a "reflexive" order*, since it is *not* antisymmetric (e.g., $1R3 \wedge 3R1$ requires $1 = 3$).

It turns out that if $\mathbb{P}$ has MC, then so does $\mathbb{P}^+$ and hence, in particular, it is a partial order, being irreflexive. □

5.2.13 Theorem *If $\mathbb{P}$ has MC (IC), then so does $\mathbb{P}^+$.*

Proof Let $\emptyset \neq \mathbb{A}$. Let $a \in \mathbb{A}$ be $\mathbb{P}$-minimal, i.e.,

$$(a)\mathbb{P}^{-1} \uparrow \tag{1}$$

Suppose now that $b\mathbb{P}^+a$ for some b. Then, for some $b_1, b_2, \ldots, b_k$ we have

$$b\mathbb{P}b_1\mathbb{P}b_2\mathbb{P}\ldots\mathbb{P}b_k\mathbb{P}a$$

But $b_k\mathbb{P}a$ contradicts (1). Therefore a is also $\mathbb{P}^+$-minimal. □

5.2.14 Corollary *If $\mathbb{P}$ has MC (IC) over $\mathbb{A}$, then $(\mathbb{P}|\mathbb{A})^+$ has MC (IC).*

Proof It is given that $\mathbb{P}|\mathbb{A}$ has MC (IC). By 5.2.13 $(\mathbb{P}|\mathbb{A})^+$ has MC (IC). □

We cannot sharpen the above to "$\mathbb{P}^+$ has MC (IC) *over* $\mathbb{A}$", for that means that $\mathbb{P}^+|\mathbb{A}$ has MC. This is not true though: Let $\mathbb{O}$ be the odd natural numbers, and R be defined *on* $\mathbb{N}$ by xRy iff $x = y + 1$, thus $R^+ = >$.

Now, R has MC over $\mathbb{O}$ (for $R|\mathbb{O} = \emptyset$), yet R^+ does *not*, for $R^+|\mathbb{O}$ has an infinite descending chain in $\mathbb{O}$:

$$\ldots > 7 > 5 > 3 > 1$$

In particular, we note from this example that $(\mathbb{P}|\mathbb{A})^+ \neq \mathbb{P}^+|\mathbb{A}$ in general.

5.2.15 Example Let $\prec$ on $\mathbb{N}$ be defined by $n \prec m$ iff $m = n + 1$. It is obvious that $\prec$ is well-founded, hence it has MC and IC by 5.2.9.

What is $\prec$-induction? For notational convenience let "$(\forall x)$" stand for "$(\forall x \in \mathbb{N})$". Thus, for any formula $F(x)$

$$(\forall n)\big((\forall x \prec n)F(x) \to F(n)\big) \to (\forall n)F(n) \qquad (\prec\text{-}IC)$$

holds.

In other words, if $F(0)$ is proved —this is $(\forall x \prec 0)F(x) \to F(0)$— and if also $F(n-1) \to F(n)$ is proved for all $n > 0$, then $(\forall n)F(n)$ holds.

This is just our familiar (from $\mathbb{N}$) "simple" (as opposed to "course-of-values") induction SI over $\mathbb{N}$.

The "natural" $<$ on $\mathbb{N}$ is $\prec^+$. $<$-induction over $\mathbb{N}$ coincides with the "usual" CVI over $\mathbb{N}$ displayed at the top of Section 5.2. □

5.2.16 Example The *Axiom of Foundation* of axiomatic set theory ZFC[5] is

$$(\exists y)F[y] \to (\exists y)\Big(F[y] \wedge \neg(\exists z \in y)F[z]\Big)$$

It says that $\in$ has MC. Therefore properties of sets can be proved by $\in$-IC ($\in$-CVI) over $\mathbb{U}$.

□ 

5.2.2 Induction Examples

5.2.17 Example This is the "classical *first example* of induction use" in the discrete math bibliography! Prove that

$$0 + 1 + 2 + \ldots + n = \frac{n(n+1)}{2} \qquad (1)$$

So, the property to prove is the entire expression (1). One must learn to not have to rename the "properties to use" as "$P[n]$".

I will use SI. So let us do the *Basis*. Boundary case is $n = 0$. We verify: $lhs = 0$. $rhs = (0 \times 1)/2 = 0$. Good!

Now fix n and take the expression (1) as I.H.

[5] According to Zermelo and Fraenkel, with the Axiom of Choice.

Do the I.S. Prove:

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

Here it goes

$$\begin{aligned} 0 + 1 + 2 + \ldots + n + (n+1) &\overset{\text{using I.H.}}{=} \frac{n(n+1)}{2} + (n+1) \\ &\overset{\text{arithmetic}}{=} (n+1)(n/2+1) \\ &\overset{\text{arithmetic}}{=} \frac{(n+1)(n+2)}{2} \end{aligned}$$

□

I will write more concisely in the examples that follow.

5.2.18 Example Same as above but doing away with the "0+". Again, I use SI.

$$1 + 2 + \ldots + n = \frac{n(n+1)}{2} \tag{1}$$

- *Basis*. $n = 1$: (1) becomes $1 = (1 \times 2)/2$. True.
- Take (1) as I.H. with fixed n.
- I.S.:

$$\begin{aligned} 1 + 2 + \ldots + n + (n+1) &\overset{\text{using I.H.}}{=} \frac{n(n+1)}{2} + (n+1) \\ &\overset{\text{arithmetic}}{=} (n+1)(n/2+1) \\ &\overset{\text{arithmetic}}{=} \frac{(n+1)(n+2)}{2} \end{aligned}$$

□

5.2.19 Example Prove

$$1 + 2 + 2^2 + \ldots 2^n = 2^{n+1} - 1 \tag{1}$$

By SI.

- Basis. $n = 0$: $1 = 2^0 = 2^1 - 1$. True.
- As I.H. take (1) for fixed n.
- I.S.

$$\begin{aligned} 1 + 2 + 2^2 + \ldots + 2^n + 2^{n+1} &\overset{\text{using I.H.}}{=} 2^{n+1} - 1 + 2^{n+1} \\ &\overset{\text{arithmetic}}{=} 2 \cdot 2^{n+1} - 1 \\ &\overset{\text{arithmetic}}{=} 2^{n+2} - 1 \end{aligned}$$

□

5.2.20 Example (Euclid) Every natural number $n \geq 2$ has a prime factor.
I do CVI (as you will see why!)

- *Basis*: For $n = 2$ we are done since 2 is a prime and $2 = 2 \times 1$.[6]
- I.H. Fix an n and assume the claim for all k, such that $2 \leq k < n$.
- I.S.: Prove for n: Two subcases:

1. If n is prime, then OK! n divides n.
2. If not, then $n = a \cdot b$, where $a \geq 2$ **and** $b \geq 2$. By I.H.[7] a has a prime factor, thus so does $n = a \cdot b$. □

5.2.21 Example (Euclid) Every natural number $n \geq 0$ is expressible base-10 as an expression

$$n = a_m 10^m + a_{m-1} 10^{m-1} + \cdots + a_1 10 + a_0 \tag{1}$$

$$\text{where each } a_i \text{ satisfies } 0 \leq a_i < 10 \tag{2}$$

Proof by CVI again. You will see why.

- *Basis*. For $n = 0$ the expression "0" has the form of the rhs of (1) *and* satisfies inequality (2).
- Fix an $n > 0$ and assume (I.H.) that if $k < n$, then k can be expressed as in (1) and (2).
- For the I.S. express the n of the I.H. using Euclid's theorem (3.3.14) as

$$n = 10q + r$$

where $0 \leq r < 10$. By the I.H. —since $q < n$— let

$$q = b_t 10^t + b_{t-1} 10^{t-1} + \cdots + b_1 10 + b_0$$

with $0 \leq b_j < 10$.

Then

$$\begin{aligned} n &= 10q + r \\ n &= 10\Big(b_t 10^t + b_{t-1} 10^{t-1} + \cdots + b_1 10 + b_0\Big) + r \\ n &= b_t 10^{t+1} + b_t 10^t + \cdots + b_1 10^2 + b_0 10 + r \end{aligned}$$

We see that n has the right form since $0 \leq r < 10$. □

---

[6] You will recall that a number $\mathbb{N} \ni n > 1$ is a *prime* iff —*by definition*— its **only** factors are 1 and n.

[7] You see? Do you know many natural numbers n such that $n - 1$ divides n?! Only 2 has this property, but 2 is just our Basis!

5.2.22 Example An inequality this time. Prove $n < 2^n$, for $n \geq 0$. We do SI.

Basis. $n = 0$. We prove $0 < 2^0$. This is true since $2^0 = 1$.

Take as I.H.

$$n < 2^n \tag{1}$$

for fixed unspecified n.

The I.S. requires $n + 1 < 2^{n+1}$. Well, $1 \leq$ [8] 2^n. Adding this to the assumed (1), term by term, we get

$$n + 1 < 2^n + 2^n = 2^{n+1}$$

□

5.2.23 Example Another inequality. Let p_n denote the n-th prime number, for $n \geq 0$. Thus $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, etc.

We prove that

$$p_n \leq 2^{2^n} \tag{1}$$

I use CVI on n. This is a bit of a rabbit out of a hat if you never read Euclid's proof that there are infinitely many primes.

- Basis $p_0 = 2 \leq 2^{2^0} = 2^1 = 2$.
- Fix $n > 0$ and take (1) as I.H.
- The I.S.: I will work with the fixed n above and the expression (product of primes, plus 1; this is inspired from Euclid's proof quoted above).

$$p_0 p_1 p_2 \cdots p_n + 1$$

I have

$$\begin{aligned}
p_0 p_1 p_2 \cdots p_n + 1 &\leq 2^{2^0} 2^{2^1} 2^{2^2} \cdots 2^{2^n} + 1 && \text{by I.H.} \\
&= 2^{2^0 + 2^1 + 2^2 + \cdots + 2^n} + 1 && \text{algebra} \\
&= 2^{2^{n+1}-1} + 1 && \text{by 5.2.19} \\
&< 2^{2^{n+1}-1} + 2^{2^{n+1}-1} && \text{smallest } n \text{ possible is } 0 \\
&= 2^1 \cdot 2^{2^{n+1}-1} \\
&= 2^{2^{n+1}}
\end{aligned}$$

[8] The "=" is attained for $n = 0$.

Now we have two cases on $q = p_0 p_1 p_2 \cdots p_n + 1$

1. q is a prime. Because of the "$+1$" q is different from all p_i in the product, so q is p_{n+1} or p_{n+2} or p_{n+3} or …

   Since the sequence of primes is strictly increasing, p_{n+1} is the least that q can be.

   Thus
   $$p_{n+1} \leq p_0 p_1 p_2 \cdots p_n + 1 \leq 2^{2^{n+1}}$$
   in this case.
2. q is composite. By 5.2.20 some prime r divides q. Now, none of the
   $$p_0, p_1, p_2, \cdots, p_n$$
   divides q because of the "$+1$". Thus r is different from all of them, so it must be one of p_{n+1} or p_{n+2} or p_{n+3} or …

   Thus,
   $$p_{n+1} \leq r < q = p_0 p_1 p_2 \cdots p_n + 1 \leq 2^{2^{n+1}}$$

Done! □

5.2.24 Example Let

$$b_1 = 3,\ b_2 = 6$$
$$b_k = b_{k-1} + b_{k-2}, \text{ for } k \geq 3$$

Prove by induction that b_n *is divisible by 3 for* $n \geq 1$. (Be careful to distinguish between what is *basis* and what are *cases* arising from the **induction step**! As you know, many texts are careless about this.)

Proof So the boundary condition is (from the italicised part above) $n = 1$. This is the *Basis*.

1. *Basis*: For $n = 1$, I have $b_1 = 3$ and this is divided by 3. We are good.
2. *I.H.* Fix n and **assume claim** for all $k < n$.
3. *I.S.* **Prove claim** for the above fixed n. There are two cases, as the I.H. is *not useable* for $n = 2$. Why? Because it would require entries b_0 and b_1. The b_0 entry does not exist since the sequence starts with b_1. So,

   Case 1. $n = 2$. Then I am OK as $b_2 = 6$; it *is* divisible by 3.

Case 2. $n > 2$. Is b_n divisible by 3? Well, $b_n = b_{n-1} + b_{n-2}$ in this case. By I.H. (valid for all $k < n$) I have that $b_{n-1} = 3t$ and $b_{n-2} = 3r$, for some integers t, r. Thus, $b_n = 3(t + r)$. Done!

□

5.2.25 Example (The Binomial Theorem) We prove in this example the so-called *binomial theorem*, for any $\mathbb{N} \ni n > 0$ and any real or complex numbers a and b.

$$(a + b)^n = \sum_{i=0}^{n} \binom{n}{i} a^{n-i} b^i \tag{1}$$

First let us take care of notation.

5.2.26 Definition (Binomial Coefficients) The notation

$$\binom{n}{m}$$

is called a *binomial coefficient* and stands for

$$\frac{n!}{(n-m)!m!}$$

where in turn $n!$ stands for $1 \times 2 \times 3 \times \cdots \times n$, that is, it is inductively defined as

$$0! = 1$$
$$(n + 1)! = (n + 1) \times n!$$

We call "$n!$" "n-factorial". □

Suppose we have n objects. In how many ways can we choose m among them ($m < n$) *ignoring* repetitions? Well, the first of the m elements I can choose in n ways. The second of the m I can choose in $n - 1$ ways after I chose and removed the first. Clearly then I can choose the 3rd in $n - 2$ ways after I removed the 2nd; etc.

All in all, I can choose all m members in $n(n-1)(n-2)\cdots\big(n-(m-1)\big)$ ways.

Wait! I have $m!$ repetitions in my choices of m elements if I do nothing else. So the final answer is the above divided by $m!$

$$\frac{n(n-1)(n-2)\cdots\big(n-(m-1)\big)}{m!} =$$

$$\frac{n(n-1)(n-2)\cdots\big(n-(m-1)\big)\overbrace{(n-m)\big(n-(m+1)\big)\cdots 3\cdot 2\cdot 1}^{(n-m)!}}{m!(n-m)!} =$$

$$\frac{n!}{m!(n-m)!}$$

Before we embark on the proof of the binomial theorem here are some properties of the symbol $\binom{n}{m}$ that we will use:

I. $\binom{n}{0} = 1$. Indeed, $\binom{n}{0} = \frac{n!}{(n-0)!0!}$, but $0! = 1$ by definition.

II. $\binom{n}{n} = 1$. Indeed, $\binom{n}{n} = \frac{n!}{n!(n-n)!}$.

III.

$$\binom{n}{m} + \binom{n}{m-1} = \binom{n+1}{m}$$

Indeed we work from left to right using the definition of $\binom{n}{m}$.

$$\begin{aligned}\binom{n}{m} + \binom{n}{m-1} &= \frac{n!}{(n-m)!m!} + \frac{n!}{\big(n-(m-1)\big)!(m-1)!} \\ &= \frac{n!}{(n-m)!(m-1)!}\left(\frac{1}{m} + \frac{1}{n-(m-1)}\right) \\ &= \frac{n!}{(n-m)!(m-1)!}\left(\frac{n+1}{m(n-m+1)}\right) \\ &= \frac{(n+1)!}{(n+1-m)!m!} \\ &= \binom{n+1}{m}\end{aligned}$$

The proof of the binomial theorem now. By induction on n.

Basis. $n = 1$. We have $(a+b)^1 = \sum_{i=0}^{1} \binom{1}{i} a^{n-i}b^i = \binom{1}{0}a^1b^0 + \binom{1}{1}a^0b^1 = a + b$, since $\binom{1}{0} = 1 = \binom{1}{1}$.

We now *fix* n and take as I.H. (1) at the head of this example.

For the same fixed n we next probe $n + 1$:

$$
\begin{aligned}
(a+b)^{n+1} &= (a+b)\Big((a+b)^n\Big) \\
&\overset{I.H.}{=} (a+b)\left(\binom{n}{0}a^n + \binom{n}{1}a^{n-1}b^1 + \binom{n}{2}a^{n-2}b^2 + \cdots + \binom{n}{n}b^n\right) \\
&\overset{multiply}{=} \binom{n}{0}a^{n+1} + \binom{n}{1}a^n b^1 + \binom{n}{2}a^{n-1}b^2 + \binom{n}{3}a^{n-2}b^3 + \cdots + \binom{n}{n}a^1 b^n \\
&\quad + \binom{n}{0}a^n b + \binom{n}{1}a^{n-1}b^2 + \binom{n}{2}a^{n-2}b^3 + \cdots \\
&\quad + \binom{n}{n-1}a^1 b^n + b^{n+1} \\
&\overset{\text{I. II. III.}}{=} \binom{n+1}{0}a^{n+1} + \binom{n+1}{1}a^n b^1 + \binom{n+1}{2}a^{n-1}b^2 + \binom{n+1}{3}a^{n-2}b^3 + \\
&\quad \cdots + \binom{n+1}{n}a^1 b^n + \binom{n+1}{n+1}b^{n+1}
\end{aligned}
$$

□ 

Here are a few additional exercises for you to try.

5.2.27 Exercise

1. Prove that $2^{2n+1} + 3^{2n+1}$ is divisible by 5 for all $n \geq 0$.
2. Using induction prove that $1^3 + 2^3 + \ldots + n^3 = \left[\dfrac{n(n+1)}{2}\right]^2$, for $n \geq 1$.
3. Using induction prove that $\sum_{i=1}^{n+1} i2^i = n2^{n+2} + 2$, for $n \geq 0$.
4. Using induction prove that $\sqrt{n} < \dfrac{1}{\sqrt{1}} + \dfrac{1}{\sqrt{2}} + \ldots + \dfrac{1}{\sqrt{n}}$, for $n \geq 2$.
5. Let

$$
\begin{aligned}
b_0 &= 1,\ b_1 = 2,\ b_3 = 3 \\
b_k &= b_{k-1} + b_{k-2} + b_{k-3}, \text{ for } k \geq 3
\end{aligned}
$$

Prove by induction that $b_n \leq 3^n$ for $n \geq 0$. (Once again, be careful to distinguish between what is *basis* and what are *cases* arising from the **induction step**!) □

As a postscript to our examples of induction proofs we offer this comment. It is clear that since sets such as $\mathbb{N} - \{0, 1, 3, 4, 5\}$ and $\mathbb{N} \cup \{-3, -2, -1\}$ are well-ordered (by $<$) we can carry induction proofs over them. In the former case the "basis" case is at 6, in the latter case it is at -3. In fact, in the preceding problem 4. the basis is at $n = 2$.

5.3 Inductive Definitions of Functions

Inductive definitions are increasingly being renamed to "recursive definitions" in the modern literature, thus using "recursive" for *definitions*, and "induction" for *proofs*. I will not go out of my way to use this dichotomy of nomenclature. Here are some familiar examples of inductive definitions of functions.

5.3.1 Example For any integer $a > 0$ we define

$$\begin{aligned} a^0 &= 1 \\ a^{n+1} &= a \cdot a^n \end{aligned} \tag{†}$$

This is an example of an inductive (recursive) definition of the non-negative integer powers of a non zero number a.

One can use SI to prove for $n \geq 1$ that the above definition ensures that

$$a^n = \overbrace{a \times a \times a \times \cdots \times a}^{n\ a} \tag{1}$$

□

5.3.2 Example Another example is the Fibonacci sequence,[9] given by

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1, \text{ and for } n \geq 1 \\ F_{n+1} &= F_n + F_{n-1} \end{aligned} \tag{‡}$$

Unlike the function (sequence) $a^0, a^1, a^2, a^3, \ldots$, for which we only need the value at n to compute the value at $n+1$, the Fibonacci function needs two previous values, at $n-1$ and at n, to compute the value at $n+1$. □

The question is: Given an inductive *definition* of a function, can we *prove* that a function f *exists* —that is, a *potentially infinitely long* table of input/output pairs— that *satisfies* the "inductive specification"?

This translates, in the first example above, into "is there a realisation f —as a function, an infinite table in this case— of what the definition (†) specifies as the behaviour of the function?"

Such a function must obey the two equations below:

$$\begin{aligned} f(0) &= 1 \\ f(n+1) &= a \cdot f(n) \end{aligned} \tag{†′}$$

[9] The "sequence" $F_0, F_0, F_0, \ldots$ is, of course, a total function $F : \mathbb{N} \to \mathbb{N}$.

How **NOT** *to answer this: "Of course it exists. This* f *satisfies* $f(0) = 1$*, so we got an output for input* $x = 0$*. If we now assume that we* do have an output $f(n)$ *at input* $x = n$ *(I.H.), then at input* $x = n + 1$ *we have the output* $a \times f(n)$*."*

What just happened here? We proved that **IF** a function f that satisfies (†′) **exists**, *then it is total*. We *never proved that the infinite table, which the function* f *is supposed to be,* **exists** and has the stated property (i.e., obeys the inductive definition).

In fact *we took for granted* that f *exists* and satisfies the recurrence equations (†′) and proceeded to prove that *then* it will be total!

The above (non) "proof" of **function existence** has actually appeared in print in a Discrete Math text!

This section looks into inductive definitions in general, and proves that a function defined inductively as in, for example, (†′) above *exists* and is *unique*.

We said "in general" above. So we will present the existence and uniqueness theorem by *generalising* the Fibonacci example above in several directions, as follows:

1. Will have the second equation depend on *several* (more than just *two* that we used in the Fibonacci definition) *recursive calls*, as we name them in computer programming.
2. The defined function *will not need to be total* nor will the functions —which make the recursive calls (such as the function "+" in the Fibonacci example and "×" in the exponential example)— need be total.
3. The inductive definition in the Fibonacci example defines F_n in terms of *two* recursive calls on the two *immediately preceding* (along $<$) arguments $n - 1$ and $n - 2$ of n. We generalise this in two directions:

   - The *order* $\mathbb{P}$ we use in an inductive definition to "sort the inputs" in the general case (equation number two) is not necessarily $<$ on $\mathbb{N}$, *nor is necessarily total*.
   - The 2nd equation defines some function g at an argument a using *one recursive call for each predecessor* of a along the order $\mathbb{P}$.

Thus, to motivate the general inductive definition of a function $\mathbb{F}$ *over any class* equipped with an order $\mathbb{P}$ that has IC, we first sketch the case of an inductive definition of a function K over $\mathbb{N}$ equipped with the standard order $<$.

5.3.3 Tentative Definition We consider in this section a general recursive definition of a function $K : \mathbb{N} \to A$, for a given set A.

The function $G : \mathbb{N} \times 2^A \to A$ is *given* and performs the needed recursive calls. A typical call to G is $G(n, X)$ where $n \in \mathbb{N}$ and $X \subseteq A$, that is, $X \in 2^A$. Let us also fix a $C \in A$.

This inductive definition of K has the *form* below.

$$\begin{aligned} &K(0) = C, \text{ and, for } n > 0 \\ &K(n) = G\Big(n, \Big\{K(0), K(1), \ldots, K(n-1)\Big\}\Big) \end{aligned} \tag{1}$$

□

5.3.4 Remark The notation of the set-argument

$$\Big\{K(0), K(1), \ldots, K(n-1)\Big\} \tag{2}$$

in the definition (1) above is *significantly less* informative than the notation implies! Its members —listed again in (2)— are just members of the set A and the marking of the inputs responsible for the various $K(i)$ is *not* embedded in these output values! So neither we, nor G, knows which is which if we are just given the *values* in the set (2).

Can we modify the right hand side of $K(n)$ to $G\Big(n, K(0), K(1), \ldots, K(n-1)\Big)$? No, because a function G cannot have a *variable number* of arguments! ($n+1$ arguments in all), that increases or decreases with the value of n.

This final idea however works: Tag along the input values that cause the $K(i)$, that is, use

$$\begin{aligned} &K(0) = C, \text{ and, for } n > 0 \\ &K(n) = G\Big(n, \Big\{\big(0, K(0)\big), \big(1, K(1)\big), \ldots, \big(n-1, K(n-1)\big)\Big\}\Big) \end{aligned} \tag{3}$$

A more elegant way to write down

$$\Big\{\big(0, K(0)\big), \big(1, K(1)\big), \ldots, \big(n-1, K(n-1)\big)\Big\}$$

is (cf. 3.1.4)

$$K \restriction (n) >$$

which we will use in all that follows. □ 

Our theorem of the existence and uniqueness of inductively defined functions will be for recursions along an arbitrary *partial* order $\mathbb{P}$ with IC and then we will obtain as a corollary the case where $\mathbb{P}$ has IC but is not necessarily an order. Of course, a trivial corollary to all that will be the existence and uniqueness of functions K defined as in (3) above.

5.3.5 Definition (Levy (1979)) A relation $\mathbb{P}$ is *left-narrow* iff $(x)\mathbb{P}^{-1}$ is a *set* for all x. It is *left-narrow over* $\mathbb{A}$ iff $\mathbb{P} \mid \mathbb{A}$ is left-narrow. □

For example, $\in$ is left-narrow by the *foundation axiom* (5.2.16 and p. 127), while $\ni$ is not.

5.3.6 Definition (Initial Segments) If $<$ is an order on $\mathbb{A}$ and $a \in \mathbb{A}$, then the *class* $\{x : x < a\} = (a) >$ is called the (initial) *open segment* defined by a, while the class $(a) \geq = \{x : x \leq a\}$ is called the *closed* segment defined by a. □

$\leq$ is $< \cup =$, of course, so that $(a) \geq = (a) > \cup \{a\}$. Segments of left-narrow relations are sets.

5.3.7 Theorem (Recursive or inductive definitions) *Let* $<: \mathbb{A} \to \mathbb{A}$ *be a left-narrow order with IC, and* $\mathbb{G}$ *a* (not necessarily total) *function* $\mathbb{G} : \mathbb{A} \times \mathbb{U} \to \mathbb{X}$*, for some class* $\mathbb{X}$*.*

Then there exists *a* unique *function* $\mathbb{F} : \mathbb{A} \to \mathbb{X}$ *satisfying:*

$$(\forall a \in \mathbb{A})\mathbb{F}(a) = \mathbb{G}(a, \mathbb{F} \restriction (a) >) \tag{1}$$

The requirement of left-narrowness guarantees (via Principle 3, 3.3.6) that the second argument of $\mathbb{G}$ in (1) is a set. This restriction does not adversely affect applicability of the theorem as the reader will be able to observe.

In (1) above "=" is Kleene's *extended equality*, so that in the recurrence (1) above we have either *both* sides are *defined* and *equal* (as sets or atoms), or *both are undefined* (see 3.5.11).

Proof We prove uniqueness first, so let $\mathbb{H} : \mathbb{A} \to \mathbb{X}$ also satisfy (1). Let $a \in \mathbb{A}$ and adopt the I.H. that

$$(\forall b < a)\mathbb{F}(b) = \mathbb{H}(b)$$

that is, for all $b < a$, $(\forall y)(\langle b, y\rangle \in \mathbb{F} \leftrightarrow \langle b, y\rangle \in \mathbb{H})$, and therefore

$$\mathbb{F} \restriction (a) >= \mathbb{H} \restriction (a) >$$

It follows that

$$\begin{aligned} \mathbb{F}(a) &= \mathbb{G}(a, \mathbb{F} \restriction (a) >) \\ &= \mathbb{G}(a, \mathbb{H} \restriction (a) >) \\ &= \mathbb{H}(a) \end{aligned}$$

This settles the claim of uniqueness: $(\forall a \in \mathbb{A})\mathbb{F}(a) = \mathbb{H}(a)$, that is, $\mathbb{F} = \mathbb{H}$. Define now,

$$\begin{aligned} \mathcal{F} = \Big\{ f : (\exists a \in \mathbb{A})\Big(& f : (a) \geq \to \mathbb{X} \\ & \wedge \big(\forall x \in (a) \geq \big) f(x) = \mathbb{G}(x, f \restriction (x) >)\Big)\Big\} \end{aligned} \tag{2}$$

Note that $\mathcal{F} \neq \emptyset$. For example, if $a \in \mathbb{A}$ is $<$-minimal,[10] then $(a) >= \emptyset$ and hence $f \restriction (a) >= \emptyset$ for any f, thus $\mathcal{F}$ contains $\{(a, \mathbb{G}(a, \emptyset))\}$, if $\mathbb{G}(a, \emptyset)\downarrow$, else it contains the empty function $\emptyset : (a) \geq \to \mathbb{X}$.

For the latter we clearly have $\emptyset(a) = \mathbb{G}(a, \emptyset \restriction (a) >)$, where both sides are undefined.

A trivial adaptation of the uniqueness argument to the case that $\mathbb{A}$ is a closed segment $(a) \geq$, shows that if $f : (a) \geq \to \mathbb{X}$ and $g : (a) \geq \to \mathbb{X}$ are in $\mathcal{F}$, then $f = g$. We use "f_a" to denote the unique $f : (a) \geq \to \mathbb{X}$ for each $a \in \mathbb{A}$, *if it exists*.

To remove the hedging, fix a and assume (I.H.) that, for each $b < a$, $f_b : (b) \geq \to \mathbb{X}$ satisfying (1) (where here $\mathbb{A} = (b) \geq$) *exists*.

Let us argue that so does f_a. Indeed, define $h : (a) \geq \to \mathbb{X}$ from the existing (by the I.H.) f_b and $\mathbb{G}$ by

$$h = \begin{cases} \{\langle a, \mathbb{G}(a, \bigcup_{b<a} f_b)\rangle\} \cup \bigcup_{b<a} f_b & \textbf{if } \mathbb{G}(a, \bigcup_{b<a} f_b)\downarrow \\ \bigcup_{b<a} f_b & \textbf{otherwise} \end{cases} \tag{3}$$

Observe next that, by transitivity of $<$,[11] we have $(c) \geq \subseteq (b) \geq$ whenever $c \leq b$,[12] therefore $f_c \subseteq f_b$, due to $f_c = f_b \restriction \leq (c)$ (by uniqueness).

We draw two conclusions:

First, to retire the induction, note that $\bigcup_{b<a} f_b$ in (3) is single-valued (a function) and is equal to $h \restriction (a) >$. Thus h satisfies the recurrence (1) at a outright, and also at $b < a$ because

$$\begin{aligned} h(b) &= f_b(b) \\ &= \mathbb{G}(b, f_b \restriction (b) >) \\ &= \mathbb{G}(b, h \restriction (b) >), \text{ since } f_b \subseteq h \end{aligned}$$

It follows that $h = f_a$, and hence by CVI f_a exists for all $a \in \mathbb{A}$.

Second,

$$f_a \in \mathcal{F} \wedge f_b \in \mathcal{F} \wedge x \in \mathrm{dom}(f_a) \cap \mathrm{dom}(f_b) \to f_a(x) = f_b(x) \tag{4}$$

because $(x) \geq \subseteq (a) \geq \cap (b) \geq$, hence $f_a \restriction (x) \geq = f_b \restriction (x) \geq$ by uniqueness on $(x) \geq$.

By (4),

$$\mathbb{F} = \bigcup \mathcal{F} \quad \text{is a function} \quad \mathbb{F} : \mathbb{A} \to \mathbb{X}$$

$\mathbb{F}$ satisfies the recurrence (1) of the theorem. Indeed, let $a \in \mathbb{A}$. Then

[10] Since $<$ has MC on $\mathbb{A}$, it does have minimal elements.

[11] We have just used the assumption that $<$ is an order.

[12] Let $x \in (a) \geq$. Then $x \leq a$ and $a \leq b$ hence $x \leq b$.

$$\begin{aligned}\mathbb{F}(a) &= f_a(a), \text{ since } f_a \subseteq \mathbb{F}\\ &= \mathbb{G}(a, f_a \restriction (a) >)\\ &= \mathbb{G}(a, \mathbb{F} \restriction (a) >), \text{ since } f_a \subseteq \mathbb{F}\end{aligned}$$

□

5.3.8 Remark

(1) Since $(a) \geq$ is a set for each $a \in \mathbb{A}$ (by left-narrowness), so is $\text{dom}(f)$ for each $f \in \mathcal{F}$ and hence each f itself is a set, by Principle 3, so forming the class $\mathcal{F}$ is legitimate.[13]

(2) The simple recursion on the natural numbers, where g is total

$$\begin{aligned} f(0) &= a\\ \text{for } n \geq 0,\ \ f(n+1) &= g(n, f(n))\end{aligned} \tag{1}$$

is a special case of 5.3.7:
Indeed we can rewrite (1) as

$$(\forall n \in \mathbb{N}) f(n) = G(n, f \restriction (n) >) \tag{2}$$

where

$$G(n,h) = \begin{cases} a & \textbf{if } n = 0\\ g(n-1, h(n-1)) & \textbf{if } h \text{ is a function} \wedge \text{dom}(h) = \{x : x < n\}\\ \uparrow & \textbf{otherwise}\end{cases}$$

Note that G on $\mathbb{N} \times \mathbb{U}$ is nontotal. In particular, if the second argument is not of the correct type (middle case above), G will be undefined. We can still prove that $f(n) \downarrow$ for all $n \in \mathbb{N}$.
Indeed, assume the claim for $m < n$ (I.H.). For $n = 0$, $f(0) = G(0, \emptyset) = a$; defined. Let next $n > 0$. Now $f(n) = G(n, f \restriction (n) >)$ and $\text{dom}(f \restriction (n) >) = (n) >$ by I.H., hence $f(n) = G(n, f \restriction (n) >) = g\big(n-1, \big(f \restriction (n) > \big)(n-1)\big) = g(n-1, f(n-1))$; defined ($g$ total).

(3) In view of the above, it is worth noting that a recursive definition *à la* 5.3.7 can still define a total function, even if $\mathbb{G}$ is nontotal. □

5.3.9 Corollary (Inductive Definition with Respect to Any $\mathbb{P}$ with IC) *Montague (1955), Tarski (1955)*

Let $\mathbb{P} : \mathbb{A} \to \mathbb{A}$ *be a left-narrow relation* —not necessarily an order— *with IC, and* $\mathbb{G}$ *a* (not necessarily total) *function* $\mathbb{G} : \mathbb{A} \times \mathbb{U} \to \mathbb{X}$, *for some class* $\mathbb{X}$. *Then there* exists *a*

[13] A class must contain only sets or atoms.

unique *function* $\mathbb{F} : \mathbb{A} \to \mathbb{X}$ *satisfying:*

$$(\forall a \in \mathbb{A})\mathbb{F}(a) = \mathbb{G}(a, \mathbb{F} \restriction (a)\mathbb{P}^{-1})$$

Proof Define $\widetilde{\mathbb{G}} : \mathbb{A} \times \mathbb{U} \to \mathbb{X}$ by

$$\widetilde{\mathbb{G}}(a, f) = \begin{cases} \uparrow & \textbf{if } f \text{ is not a function} \\ \mathbb{G}(a, f \restriction (a)\mathbb{P}^{-1}) & \textbf{othw} \end{cases}$$

Let $<$ stand for $\mathbb{P}^+$ and hence $>$ is $(\mathbb{P}^{-1})^+$ (cf. Exercise 3.9.21). Now $<$ is an order on $\mathbb{A}$ that has IC, and is left-narrow since

$$(a)(\mathbb{P}^{-1})^+ = \bigcup \{(a)(\mathbb{P}^{-1})^n : n > 0\}$$

and an easy argument shows that each $(a)(\mathbb{P}^{-1})^n$ is a set (Exercise 5.4.27). Thus, by 5.3.7, there is a unique $\mathbb{F} : \mathbb{A} \to \mathbb{X}$ such that

$$\begin{aligned} (\forall a \in \mathbb{A})\mathbb{F}(a) &= \widetilde{\mathbb{G}}(a, \mathbb{F} \restriction (a) >) \\ &= \mathbb{G}\big(a, \big(\mathbb{F} \restriction (a) > \big) \restriction (a)\mathbb{P}^{-1}\big) \end{aligned} \tag{1}$$

Now, since $(a)\mathbb{P}^{-1} \subseteq (a) >$ we have $(\mathbb{F} \restriction (a) >) \restriction (a)\mathbb{P}^{-1} = \mathbb{F} \restriction (a)\mathbb{P}^{-1}$, hence (1) becomes

$$(\forall a \in \mathbb{A})\mathbb{F}(a) = \mathbb{G}(a, \mathbb{F} \restriction (a)\mathbb{P}^{-1})$$

□

5.3.10 Corollary (Recursion with a total $\mathbb{G}$) *Let* $\mathbb{P} : \mathbb{A} \to \mathbb{A}$ *be a left-narrow relation* — not necessarily an order— *with IC, and* $\mathbb{G}$ *a* total *function* $\mathbb{G} : \mathbb{A} \times \mathbb{U} \to \mathbb{X}$, *for some class* $\mathbb{X}$. *Then there* exists *a* unique total *function* $\mathbb{F} : \mathbb{A} \to \mathbb{X}$ *satisfying*

$$(\forall a \in \mathbb{A})\mathbb{F}(a) = \mathbb{G}(a, \mathbb{F} \restriction (a)\mathbb{P}^{-1})$$

Proof We only need to show that $\mathrm{dom}(\mathbb{F}) = \mathbb{A}$. By 5.3.9, there is a *unique* $\mathbb{F}$ satisfying

$$(\forall a \in \mathbb{A})\mathbb{F}(a) = \mathbb{G}(a, \mathbb{F} \restriction (a)\mathbb{P}^{-1})$$

Clearly the right hand side of $=$ is defined for all $a \in \mathbb{A}$. □

5.3.11 Remark (Notation Moschovakis (1969)**)** In the following corollaries we use some notation introduced by Moschovakis:

Define, the functions π and δ by

$$\pi(z) \text{ is the } x \text{ such that } (\exists y)z = (x, y)$$

$$\delta(z) \text{ is the } y \text{ such that } (\exists x)z = (x, y)$$

Incidentally, π is for πρώτο (first) and δ for δεύτερο (second). □

5.3.12 Corollary (Recursive definition with parameters I) *Let* $\mathbb{P} : \mathbb{A} \to \mathbb{A}$ *be a left-narrow relation* —not necessarily an order— *with IC, and* $\mathbb{G}$ *a* (not necessarily total) *function* $\mathbb{G} : \mathbb{S} \times \mathbb{A} \times \mathbb{U} \to \mathbb{X}$, *for some classes* $\mathbb{S}$ *and* $\mathbb{X}$. *Then there* exists *a* unique *function* $\mathbb{F} : \mathbb{S} \times \mathbb{A} \to \mathbb{X}$ *satisfying:*

$$(\forall (s, a) \in \mathbb{S} \times \mathbb{A})\mathbb{F}(s, a) = \mathbb{G}\Big(s, a, \Big\{\Big(s, x, \mathbb{F}(s, x)\Big) : x\mathbb{P}a\Big\}\Big) \tag{1}$$


In equation (1) s persists throughout (unchanged), hence it is called a "*parameter*".


Proof Define the relation $\widetilde{\mathbb{P}}$ on $\mathbb{S} \times \mathbb{A}$ by

$$(u, a)\widetilde{\mathbb{P}}(v, b) \quad \text{iff} \quad u = v \wedge a\mathbb{P}b$$

It is clear that $\widetilde{\mathbb{P}}$ has MC. Now, (1) can be rewritten as

$$\begin{aligned}(\forall (s, a) \in \mathbb{S} \times \mathbb{A})\mathbb{F}(s, a) &= \mathbb{G}\Big(s, a, \Big\{\Big(s, x, \mathbb{F}(s, x)\Big) : (s, x)\widetilde{\mathbb{P}}(s, a)\Big\}\Big) \\ &= \mathbb{G}\Big(s, a, \mathbb{F} \upharpoonright (s, a)\widetilde{\mathbb{P}}^{-1}\Big)\end{aligned}$$

The result follows from 5.3.9 by using the $\mathbb{J}$ given below as the "$\mathbb{G}$-function"

$$\mathbb{J}(g, f) = \begin{cases} \uparrow & \textbf{if } g \notin \mathbb{S} \times \mathbb{A} \\ \mathbb{G}(\pi(g), \delta(g), f) & \textbf{othw} \end{cases}$$

Thus,

$$\begin{aligned}(\forall g \in \mathbb{S} \times \mathbb{A})\mathbb{F}(g) &= \mathbb{G}\Big(\pi(g), \delta(g), \mathbb{F} \upharpoonright (g)\widetilde{\mathbb{P}}^{-1}\Big) \\ &= \mathbb{J}\Big(g, \mathbb{F} \upharpoonright (g)\widetilde{\mathbb{P}}^{-1}\Big)\end{aligned}$$

□

5.3.13 Corollary (Recursive Definition with Parameters II) *Let all assumptions be as in Corollary 5.3.12, except that the recurrence now reads*

$$(\forall (s, a) \in \mathbb{S} \times \mathbb{A})\mathbb{F}(s, a) = \mathbb{G}\Big(s, a, \Big\{\Big(x, \mathbb{F}(s, x)\Big) : x\mathbb{P}a\Big\}\Big) \tag{1}$$

Then there exists *a* unique *function* $\mathbb{F} : \mathbb{S} \times \mathbb{A} \to \mathbb{X}$ *satisfying* (1).

Proof Apply Corollary 5.3.12 with $\widetilde{\mathbb{P}}$ as above and a "$\mathbb{G}$-function" $\mathbb{J}$, given by

$$\mathbb{J}(s, a, f) = \mathbb{G}(s, a, p_{23}(f))$$

where $p_{23} : \mathbb{U} \to \mathbb{U}$ —to get the right hand side of (2) to be the same as that in (1)— is

$$p_{23}(f) = \begin{cases} \uparrow & \textbf{if } f \text{ is } \textit{not} \text{ a class of 3-tuples} \\ \left\{\left(\delta(\pi(z)\right), \delta(z)) : z \in f\right\} & \textbf{othw} \end{cases}$$

Thus, (1) takes the format of 5.3.12,

$$(\forall (s, a) \in \mathbb{S} \times \mathbb{A})\mathbb{F}(s, a) = \mathbb{J}\left(s, a, \mathbb{F} \restriction (s, a)\widetilde{\mathbb{P}}^{-1}\right) \tag{2}$$

Note that,

$$\mathbb{F} \restriction (s, a)\widetilde{\mathbb{P}}^{-1} = \left\{\left((s, x), \mathbb{F}(s, x)\right) : x\mathbb{P}a\right\}$$

thus, setting $z = \left((s, x), \mathbb{F}(s, x)\right)$, we have $\delta\left(\pi(z)\right) = x$ and $\delta(z) = \mathbb{F}(s, x)$ as needed by (1). □

5.3.14 Corollary (Pure Recursion Along a Well-Ordering with a Partial $\mathbb{G}$) *Let* $<: \mathbb{A} \to \mathbb{A}$ *be a left-narrow* well-ordering*, and* $\mathbb{G}$ *a* (not necessarily total) *function* $\mathbb{G} : \mathbb{U} \to \mathbb{X}$*, for some class* $\mathbb{X}$*.*

Then there exists *a* unique *function* $\mathbb{F} : \mathbb{A} \to \mathbb{X}$ *satisfying* (1)–(2) *below:*
(1) $(\forall a \in \mathbb{A})\mathbb{F}(a) = \mathbb{G}(\mathbb{F} \restriction (a) >)$,
(2) *dom*$(\mathbb{F})$ *is either* $\mathbb{A}$*, or* $(a) >$ *for some* $a \in \mathbb{A}$*.*

"Pure recursion" refers to the fact that $\mathbb{G}$ has only one argument, the "history" of $\mathbb{F}$ on the open segment $(a) >$.

Proof In view of Theorem 5.3.7, we only need prove (2). So let $\text{dom}(\mathbb{F}) \neq \mathbb{A}$. Let a in $\mathbb{A}$ be $<$-minimal (also *minimum* here, since $<$ is total) such that

$$\mathbb{F}(a) \uparrow \quad \text{i.e.,} \quad \mathbb{G}(\mathbb{F} \restriction (a) >) \uparrow \tag{3}$$

Thus $(a) > \subseteq \text{dom}(\mathbb{F})$. We will argue that $\text{dom}(\mathbb{F}) = (a) >$. Well, let instead $b \in \text{dom}(\mathbb{F}) - (a) >$ be minimal such that $\mathbb{F}(b) \downarrow$.

By (3) and totalness of $<$, it is $a < b$. By choice of b,

$$(\forall x)(a \leq x < b \to \mathbb{F}(x) \uparrow)$$

Thus,

$$\mathbb{F} \restriction (b) >= \mathbb{F} \restriction (a) > \tag{4}$$

therefore

$$\begin{aligned} \mathbb{F}(b) &= \mathbb{G}(\mathbb{F} \restriction (b) >) \\ &= \mathbb{G}(\mathbb{F} \restriction (a) >) \text{ (by (4))} \\ &= \mathbb{F}(a) \end{aligned}$$

contradicting (3), since $\mathbb{F}(b) \downarrow$. □

5.3.15 Example Let $G : \{0, 1\} \times \mathbb{U} \to \{0, 1\}$ be given as

$$G(x, f) = \begin{cases} 1 & \textbf{if } x = 1 \wedge f = \emptyset \\ \uparrow & \textbf{othw} \end{cases}$$

and $\{0, 1\}$ be equipped with the standard order $<$ on $\mathbb{N}$. Then the recursive definition

$$(\forall a \in \{0, 1\}) F(a) = G(a, F \restriction (a) >)$$

yields the function $F = \{\langle 1, 1\rangle\}$ whose domain is neither $\{0, 1\}$ nor a segment of $\{0, 1\}$. Thus the requirement of *pure* recursion in 5.3.14 is essential.[14] □

5.3.16 Remark In "practice", recursive definitions with respect to a $\mathbb{P}$ that has MC (IC) have often the form

$$\mathbb{F}(s, x) = \begin{cases} \mathbb{H}(s) & \textbf{if } x \text{ is } \mathbb{P}\text{-minimal} \\ \mathbb{G}(s, x, \{\langle s, y, \mathbb{F}(s, y)\rangle : y \mathbb{P} x\}) & \textbf{othw} \end{cases}$$

This reduces to the case considered in 5.3.12 with a $\mathbb{G}$-function, $\widetilde{\mathbb{G}}$, given by

$$\widetilde{\mathbb{G}}(s, x, f) = \begin{cases} \mathbb{H}(s) & \textbf{if } x \text{ is } \mathbb{P}\text{-minimal} \\ \mathbb{G}(s, x, f) & \textbf{othw} \end{cases}$$

A similar remark holds —regarding making the "basis" of the recursion explicit— for all the forms of recursion that we have considered. □

[14] It was tacitly taken advantage of in the last step of the proof. Imagine what would happen if $\mathbb{F}$'s argument were explicitly present in $\mathbb{G}$: We would get $\mathbb{G}(b, \mathbb{F} \restriction (b) >) = \mathbb{G}(b, \mathbb{F} \restriction (a) >)$ but *not necessarily* $\mathbb{G}(b, \mathbb{F} \restriction (a) >) = \mathbb{G}(a, \mathbb{F} \restriction (a) >)$.

5.3.17 Example (The support function) The *support function* $sp : \mathbb{U} \to \mathbb{U}$ gives the set of *all* atoms, $sp(x)$, that took part in the formation of some set x.

For example,

$$sp(\emptyset) = \emptyset$$
$$sp(\{\{\{\emptyset\}\}\}) = \emptyset$$
$$sp(\{2, \{\#, !, \{1\}\}\}) = \{2, \#, 1, !\} \text{ for atoms } 2, \#, 1, !$$

The existence and uniqueness of sp is established by the following recursive definition:

$$sp(x) = \begin{cases} \{x\} & \textbf{if } x \text{ is an atom} \\ \bigcup\{sp(y) : y \in x\} & \textbf{othw} \end{cases} \tag{1}$$

That (1) is an appropriate recursion can be seen as follows:

First, $\in$ is left-narrow and has MC.

Next, (1) can be put in "standard" form (Theorem 5.3.9 in this case)

$$(\forall x \in \mathrm{dom}(sp))sp(x) = \mathbb{G}(x, sp \restriction (x) \ni) \tag{2}$$

(of course, for a set x, $(x) \ni = x$) where the *total* $\mathbb{G} : \mathbb{U} \times \mathbb{U} \to \mathbb{U}$ is given by

$$\mathbb{G}(x, f) = \begin{cases} \{x\} & \textbf{if } x \text{ is an atom} \\ \uparrow & \textbf{othw, if } f \text{ is not a relation} \\ \bigcup_{y \in f} \delta(y) & \textbf{in all other cases} \end{cases}$$

In (2) the middle case for $\mathbb{G}$ above never applies. Note that, for a set x, $\bigcup_{y \in sp \restriction x} \delta(y) = \bigcup\{sp(y) : y \in x\}$. □

5.3.18 Definition A set with empty support is called a *pure* set. □

5.3.1 Examples on Inductive Function Definitions

5.3.19 Lemma *Let $n \geq 1$. If we define the order $\prec$ on $\mathbb{N}^{n+1}$ by $(a, \vec{b}) \prec (a', \vec{b'})$ iff $a < a'$ and $\vec{b} = \vec{b'}$, then $\prec$ is an order that has MC on $\mathbb{N}^{n+1}$.*

Proof 1. $\prec$ is an order:

- Indeed, if $(a, \vec{b}) \prec (a, \vec{b})$, then $a < a$ which is absurd.

- If $(a, \vec{b}) \prec (a', \vec{b}') \prec (a'', \vec{b}'')$, then $\vec{b} = \vec{b}' = \vec{b}''$ and $a < a' < a''$. Thus $a < a''$ and hence $(a, \vec{b}) \prec (a'', \vec{b}'')$.

2. $\prec$ has MC: So let $\emptyset \neq A \subseteq \mathbb{N}^{n+1}$. Let a be $<$-minimal in $S = \{x : (\exists \vec{b})(x, \vec{b}) \in A\} \subseteq \mathbb{N}$.
Pause. Why is $S \neq \emptyset$?◀
Let $\vec{c}$ be such that $(a, \vec{c}) \in A$. This $(a, \vec{c})$ is $\prec$-minimal in A. Otherwise for some d, $A \ni (d, \vec{c}) \prec (a, \vec{c})$. Hence $d < a$, but this is a contradiction since $d \in S$ (why?). □

The minimal elements of $\prec$ in $\mathbb{N}^{n+1}$ are of the form $(0, \vec{b})$, $(0, \vec{b}')$, $(0, \vec{b}'')$, …, which are not comparable if they have distinct "$\vec{b}$-parts". Thus they are infinitely many.

We can now state the important (for *computability*, e.g., cf. Tourlakis (2012, 2022)).

5.3.20 Definition (Primitive Recursive Schema) The following inductive definition is the schema of *primitive recursion* due to Dedekind. Define $f : \mathbb{N}^{n+1} \to \mathbb{N}$ via given functions $h : \mathbb{N}^n \to \mathbb{N}$ and $g : \mathbb{N}^{n+2} \to \mathbb{N}$ by

$$\begin{aligned} f(0, \vec{y}) &= h(\vec{y}) \\ f(x+1, \vec{y}) &= g(x, \vec{y}, f(x, \vec{y})) \end{aligned} \tag{1}$$

□

5.3.21 Theorem *The schema (1) of 5.3.20 defines inductively a unique function* $f : \mathbb{N}^{n+1} \to \mathbb{N}$.

Proof Using the relation $\prec$ of 5.3.19 that has MC (and thus IC) on $\mathbb{N}^{n+1}$, we rewrite (1) of 5.3.20 as follows:

- First, let G be given by

$$G(x, \vec{y}, \psi) \stackrel{Def}{=} \begin{cases} h(\vec{y}) & \textbf{if } x = 0 \\ g\Big(x-1, \vec{y}, \psi\big(x-1, \vec{y}\big)\Big) & \textbf{if } x > 0 \text{ and } \psi \text{ is a function } \mathbb{N}^{n+1} \to \mathbb{N} \\ \uparrow & \textbf{othw} \end{cases}$$

- Thus we can rewrite (1) as

$$\big(\forall (x, \vec{y}) \in \mathbb{N}^{n+1}\big) f(x, \vec{y}) = G\Big(x, \vec{y}, f \restriction \big(x, \vec{y}\big) \succ\Big) \tag{2}$$

- Noting that

$$f \upharpoonright \left(x, \vec{y}\right) \succ = \left\{\left(x-1, \vec{y}, f(x-1, \vec{y})\right), \left(x-2, \vec{y}, f(x-2, \vec{y})\right), \ldots, \left(0, \vec{y}, f(0, \vec{y})\right)\right\} \tag{3}$$

we see that the function in (3) applied to input $\left(x-1, \vec{y}\right)$ yields $f(x-1, \vec{y})$ as needed. □

5.3.22 Exercise Prove by induction on x (and using $\vec{y}$ as a parameter) that the f defined by (1) is total provided h and g are. □

Let us see some examples of primitive recursions:

5.3.23 Example We know that 2^n means

$$\overbrace{2 \times 2 \times 2 \times \ldots \times 2}^{n\ 2s}$$

But "…", or "etc.", is *not* mathematics! That is why we gave at the outset of this section the definition 5.3.1.

Applied to the case $a = 2$ we have

$$\begin{aligned} 2^0 &= 1 \\ 2^{n+1} &= 2 \times 2^n \end{aligned} \tag{1}$$

From 5.3.21 we have at once that 5.3.1 and in particular 5.3.23 defines a unique function, each satisfying its defining equations.

For the function that for each n outputs 2^n we can give an alternative definition that uses "+" rather than "×" in the "g-function" part of the definition:

$$\begin{aligned} 2^0 &= 1 \\ 2^{n+1} &= 2^n + 2^n \end{aligned}$$ □

5.3.24 Example Let $f : \mathbb{N}^{n+1} \to \mathbb{N}$ be given. How can I define $\sum_{i=0}^{m} f(i, \vec{b})$ —for any $\vec{b} \in \mathbb{N}^n$— other than by the sloppy

$$f(0, \vec{b}) + f(1, \vec{b}) + f(2, \vec{b}) + \ldots + f(i, \vec{b}) + \ldots + f(m, \vec{b})?$$

By induction/recursion, of course:

$$\begin{aligned} \textstyle\sum_{i=0}^{0} f(i, \vec{b}) &= f(0, \vec{b}) \\ \textstyle\sum_{i=0}^{m+1} f(i, \vec{b}) &= \left(\textstyle\sum_{i=0}^{m} f(i, \vec{b})\right) + f(m+1, \vec{b}) \end{aligned} \tag{1}$$

□

5.3.25 Example Let $f : \mathbb{N}^{n+1} \to \mathbb{N}$ be given. How can I define $\prod_{i=0}^{n} f(i, \vec{b})$ —for any $\vec{b} \in \mathbb{N}^n$— other than by the sloppy

$$f(0, \vec{b}) \times f(1, \vec{b}) \times f(2, \vec{b}) \times \ldots \times f(i, \vec{b}) \times \ldots \times f(n, \vec{b})?$$

By induction/recursion:

$$\begin{aligned} \textstyle\prod_{i=0}^{0} f(i, \vec{b}) &= f(0, \vec{b}) \\ \textstyle\prod_{i=0}^{n+1} f(i, \vec{b}) &= \left(\textstyle\prod_{i=0}^{n} f(i, \vec{b}) \right) \times f(n+1, \vec{b}) \end{aligned} \tag{2}$$

Again, by 5.3.21, (2) defines a unique function named $\lambda n \vec{b}. \prod_{i=0}^{n} f(i, \vec{b})$ that behaves as required. □

5.3.26 Example Here is a function with huge output! Define $f : \mathbb{N} \to \mathbb{N}$ by

$$\begin{aligned} f(0) &= 1 \\ f(n+1) &= 2^{f(n)} \end{aligned} \tag{3}$$

What does (3) look like in the notation of 5.3.21?

It is

$$\begin{aligned} f(0) &= 1 \\ f(n+1) &= G(n, f(n)) \end{aligned} \tag{3'}$$

where for all n and z, $G(n, z) = 2^z$.

What does the output $f(n)$ look like in mathematical notation? Well,

$$f(0) = 1, \quad f(1) = 2^{f(0)} = 2, \quad f(2) = 2^{f(1)} = 2^2, \quad f(3) = 2^{f(2)} = 2^{2^2}$$

Hmm! Is the guess that $f(n)$ is a ladder of n 2s correct? Yes! Let's verify by induction:

1. *Basis.* $f(0) = 1$. A ladder of zero 2s. Correct!
2. *I.H.* Fix n and assume that
$$f(n) = 2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}} \Big\} n \text{ 2s}$$
A ladder of n 2s.
3. I.S. Thus $f(n+1) = 2^{f(n)}$, so we put the ladder of n 2s of the I.H. as the exponent of 2 —forming a ladder of $n+1$ 2s— to obtain $f(n+1)$. Done! □

5.3.2 Fibonacci-like Inductive Definitions; Course-of-Values Recursion

5.3.27 Definition (Course-of-Values-Recursion) The general case of Fibonacci-like recursive definitions is based on 5.3.9 or 5.3.12 and uses the order $\prec$ (and $\succ = \prec^{-1}$).

$$\begin{aligned} f(0,\vec{y}) &= h(\vec{y}) \\ f(n,\vec{y}) &= \text{if } n>0, \text{ then } g\Big(n-1,\vec{y},f\restriction(n,\vec{y})\succ\Big) \end{aligned} \qquad \text{Fibonacci-like}$$

One often refers to the part "$f\restriction(n,\vec{y})\succ$", that is,[15]

$$\Big(n-1,\vec{y},f(n-1,\vec{y})\Big),\Big(n-2,\vec{y},f(n-2,\vec{y})\Big),\ldots,\Big(0,\vec{y},f(0,\vec{y})\Big)$$

as the *history* of f at $(n-1,\vec{y})$.

In computability theory Fibonacci-like recursions are called *Course-of-Values Recursions* (*CVR*) as they depend for their recursive calls, *in general*, on the entire history of the under definition function.

The above CVR has the form of 5.3.9 or 5.3.12. □

5.3.28 Example (Fibonacci again; with a comment re Basis case) Thus if want to fit the Fibonacci definition into the general schemata of 5.3.9 or 5.3.27 —without a parameter "$\vec{y}$"— we would choose a "g" like this

$$g(n,f)=\begin{cases} \textbf{if } f\not\subseteq\mathbb{N}^2 \textbf{ then} & \uparrow \\ \textbf{else if } n=0 \textbf{ then} & 0 \\ \textbf{else if } n=1 \textbf{ then} & 1 \\ \textbf{else if } n>1 \textbf{ then} & f(n-1)+f(n-2) \end{cases} \tag{1}$$

Thus the recurrence

$$F(n)=g(n,F\restriction(n)\succ) \tag{2}$$

shows the *uniqueness* and *existence* of the Fibonacci definition via Theorem 5.3.7 or 5.3.9.

In (1) above "$f(1)=1$" is *not* a "Basis case" because 1 is *not* minimal in $\mathbb{N}$. ("$f(0)=0$" *is* the Basis case since 0 is $<$-minimal, indeed least, in $\mathbb{N}$).

So what is "$f(1)=1$"? It is a *boundary* case of the g-definition since $n-2$ makes no sense in the Fibonacci recurrence if $n=1$. Display (2) (via (1)) yields $F(0)=0$, $F(1)=1$ and, for $n>1$, $F(n)=F(n-1)+F(n-2)$. □

[15] In the expanded version below it is understood that the tuple $(x,\vec{y},f(x,\vec{y}))$ is missing if $f(x,\vec{y})\uparrow$.

5.4 Exercises

1. Use induction to prove that $1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1$.
2. Use induction to prove that $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$.
3. Use induction to prove that $\sum_{i=1}^{n} i^3 = \left(\sum_{i=1}^{n} i\right)^2$.
   Hint. You may use the well-known $\sum_{i=1}^{n} i = n(n+1)/2$.
4. Use induction to prove that $5^n - 3^n$ is even, for $n \geq 0$.
5. Use a direct proof (no induction!) to prove that $5^n - 3^n$ is even, for $n \geq 0$.
6. Use induction to prove that $11^n - 4^n$ is divisible by 7, for $n \geq 1$.
7. Use induction to prove that $5^{n+1} + 2 \times 3^n + 1$ is divisible by 8, for $n \geq 0$.
8. Use induction to prove that $n^3 - 4n + 6$ is divisible by 3, for $n \geq 0$.
9. Use induction to prove that $n^2 - n$ is divisible by 2, for $n \geq 0$.
10. Prove that $n^3 - n$ is divisible by 3, for $n \geq 0$.
11. Using induction prove that $\frac{1}{\sqrt{1}} + \frac{1}{\sqrt{2}} + \ldots + \frac{1}{\sqrt{n}} \leq 2\sqrt{n} - 1$, for $n \geq 1$
12. Use induction to prove that $n! \geq 2^{2n}$, for $n \geq 9$.
13. Prove 9 without induction, using a one-(short)line direct proof.
14. Use induction to prove that $\sum_{i=1}^{n} (3i - 2) = (3n^2 - n)/2$.
15. *This time do not use induction*. Prove directly —using the well-known $\sum_{i=1}^{n} i = n(n+1)/2$— that $\sum_{i=1}^{n} (3i - 2) = (3n^2 - n)/2$.
16. Use induction to prove that

$$\sum_{i=1}^{n} \frac{1}{(4n-3)(4n+1)} = \frac{n}{4n+1}$$

17. Can you prove that

$$\sum_{i=1}^{n} \frac{1}{i \cdot (i+1)} = \frac{n}{n+1}$$

    without induction?
    Hint. $1/i(i+1) = \frac{1}{i} - \frac{1}{i+1}$.
18. Use induction to prove that

$$\sum_{i=1}^{n} \frac{1}{i \cdot (i+1)} = \frac{n}{n+1}$$

19. Can you prove that

$$\sum_{i=1}^{n} \frac{1}{(4n-3)(4n+1)} = \frac{n}{4n+1}$$

    without induction?

20. Let

$$b_1 = 3,\ b_2 = 6$$
$$b_k = b_{k-1} + b_{k-2}, \text{ for } k \geq 3$$

Prove by induction that b_n is divisible by 3 for $n \geq 1$. (Be careful to distinguish between what is *basis* and what are *cases* arising from the **induction step**!)

21. Prove that

$$\sum_{0 \leq k \leq n} (-2)^k = (1/3)(1 - 2^{n+1})$$

for all *odd positive* n.

22. Prove that $2^{2n+1} + 3^{2n+1}$ is divisible by 5 for all $n \geq 0$.

23. Let

$$F_0 = 0,\ \ F_1 = 1$$
$$F_k = F_{k-1} + F_{k-2}, \text{ for } k \geq 2$$

Let ϕ stand for the number $\dfrac{1+\sqrt{5}}{2}$. Prove by induction that $F_n > \phi^{n-2}$ for all $n \geq 3$.

24. Let A be a set of n elements. Prove that 2^A has 2^n elements using the binomial theorem.

Hint. By the binomial theorem (5.2.25) $2^n = (1+1)^n = \sum_{i=0}^{n} \binom{n}{i}$. Do not mix this up with the methodology suggested in Exercise 25 below.

25. Use induction on n to prove that if A has n elements, that is, $A \sim \{0, 1, \ldots, n-1\}$ if $n \geq 1$ —that is, A has the form $\{a_0, a_1, \ldots, a_{n-1}\}$— or $A = \emptyset$, then 2^A has 2^n elements.

Hint. For the induction step —*going from* $A = \{a_0, \ldots, a_{n-1}\}$ *to* $A' = \{a_0, \ldots, a_{n-1}, a_n\}$— argue that the added member a_n is in as many new subsets (of A') as A has in total.

26. Show, for any $0 < n \in \mathbb{N}$, that $(\mathbb{P}^n)^{-1} = (\mathbb{P}^{-1})^n$.

27. Let $\mathbb{P}$ on $\mathbb{A}$ be left-narrow. Show that, for any $a \in \mathbb{A}$, $(a)(\mathbb{P}^{-1})^n$ is a set.

Hint. $(y)\mathbb{P}^{-1} = \{x : x\mathbb{P}y\}$ is a set for any y by left narrowness. What values go into x in an expression like $y \underbrace{(\mathbb{P}^{-1}) \circ (\mathbb{P}^{-1}) \circ \cdots \circ (\mathbb{P}^{-1})}_{n} x$ for any y?

28. Prove that every natural number ≥ 2 is a product of primes, where 2 is the "trivial" product of *one factor*.

Hint. Use CVI in conjunction with 5.2.20.

29. Supplement the above problem to add a proof that every natural number ≥ 2 is a product of primes in a *unique* way, if we disregard permutations of the factors, which is permissible by the commutativity of $\times$.

30. Code any finite set $S = \{a_0, a_1, \ldots, a_n\}$ by a code we will name c_S ("c" for "code") given by

$$c_S \stackrel{Def}{=} \prod_{a \in S} p_a$$

where "p_a" is the "a-th prime" in the sequence of primes

position	$= 0$	1	2	3	4	...
prime name $=$	p_0	p_1	p_2	p_3	p_4	...
prime value $=$	2	3	5	7	11	...

Prove

a. The function that assigns to every finite $\emptyset \neq S$ the number c_S —and assigns 1 to $\emptyset$— is a 1-1 correspondence onto its range.
b. Use that fact to show that the set of *all finite subsets* of $\mathbb{N}$ is *enumerable*.

31. Reprove the previous problem with a different coding: To every set $S = \{a_0, a_1, \ldots, a_n\}$ —where the a_i are distinct— this time we assign the natural number (we assign 0 to $\emptyset$).

$$bc_S = 2^{a_0} + 2^{a_1} + 2^{a_2} + \cdots + 2^{a_n}$$

"bc" stands for "binary code".[16] That is, show

a. The function that assigns to every finite $\emptyset \neq S$ the number bc_S —and assigns 0 to $\emptyset$— is a 1-1 correspondence onto its range.
b. If we are given a binary code of a set, we easily find the set if we convert the number in binary notation. The positions of the 1-bits (terminology for binary digits) in the code are the values of the members of the set.
c. Now argue again that the set of finite subsets of $\mathbb{N}$ is enumerable.

32. Prove that if $A \subseteq B$ and A and C are enumerable, then $B \cup C \sim B$.
Hint. You can actually construct the 1-1 correspondence.

33. Prove that if B is infinite then it has an enumerable subset.
Hint. Construct a sequence of *infinitely* many *distinct* members of B,

$$b_0, b_1, b_2, b_3, \ldots$$

and argue that the set $\{b_0, b_1, b_2, b_3, \ldots\}$ *is enumerable*.
In more detail,

[16] The literature on *computability* refers to this *finite set* code as the "*canonical index*" (Rogers (1967), Tourlakis (2022)).

a. "Define" the sequence above by induction (recursion) by

$$\begin{aligned}&\text{Pick (and fix) any } b \text{ in } B \text{ and call it } b_0\\&\text{Pick (and fix) any } b \text{ in } B - \{b_0, \ldots, b_n\} \text{ and call it } b_{n+1}\end{aligned}$$

b. Well, "define" is a bit of an exaggeration as this implies an infinite length proof since we have/give no clue on how these infinitely many choices are done. Let's use the *axiom of choice* that guarantees a function C exists such that for each $S \in 2^B - \{\emptyset\}$ it is $C(S) \in S$: We now *really define*

$$\begin{aligned}b_0 &= C(B)\\ b_{n+1} &= C\Big(B - \{b_0, \ldots, b_n\}\Big)\end{aligned}$$

From our work on inductive definitions, the sequence (function of n, right?) exists. Next

c. Prove by induction on n that the function $\lambda n.b_n$[17] is total (on $\mathbb{N}$), 1-1 and onto the set

$$T = \{b_0, b_1, \ldots\} \tag{1}$$

(the onto part is trivial; why?).

34. Prove that if B is infinite and C is enumerable, then $B \cup C \sim B$.
Hint. Use 33 and 32.

35. (*Dedekind Infinite*) Dedekind gave this alternative definition of *infinite set*, namely,

A is infinite iff for some proper subset of A —let's call it S— we have $A \sim S$.

Prove that his definition is equivalent to the one we introduced in Definition 3.6.1. *There are two directions in the equivalence!*
Hint. Use 33 and 34. Note that if $A \subseteq B$ is enumerable, then $B = (B - A) \cup A$.

The following few exercises expand our topics.

36. (Upper bounds in POsets) Let $(A, <)$ be some PO*set*, and let $\emptyset \neq B \subseteq A$.
We call a $u \in A$ an *upper bound* of B iff, for all $x \in B$, we have $x \leq u$ —where you will recall that $x \leq u$ means $x < u \vee x = u$.
We say that u is *the least upper bound* of B (in A) —in symbols, $u = lub(B)$— or also *the supremum* or "*the sup*", in symbols, $u = sup(B)$, iff for all upper bounds u' of B we have that $u \leq u'$.
Determine upper bounds and the lub (if any) for the following two sets in the POset

$$\Big(\{1, 2, 3\}, \{(1, 2), (1, 3)\}\Big)$$

[17] λ-notation was defined in 3.5.10.

- {1} and
- {1, 2, 3}.

37. Prove that $\sqrt{2}$ is not rational. We say that a *real number* that is not rational is *irrational*.
Hint. The statement means that $\sqrt{2}$ cannot equal m/n for any integers m, n ($n \neq 0$, of course). Well, *assume* that $\sqrt{2} = m/n$, for some m, n, where we also assume that we have reduced the fraction m/n to the lowest possible numerator and denominator.
Now $2n^2 = m^2$ says that 2 divides m^2. Does it also divide m? Can you now reach a contradiction to the assumption $\sqrt{2} = m/n$?

38. (**A non Constructive Proof!**) Prove that there are irrational numbers a and b such that a^b is *rational* (fraction of two integers).
Hint. Logic tells us that $A \vee \neg A$ is true for any *sentence* A —see Exercises 4.2.8 for a definition of "sentence".
So, consider *cases*.

a. Case where $\sqrt{2}^{\sqrt{2}}$ is rational. Done!
b. Negation of case above: $\sqrt{2}^{\sqrt{2}}$ is *irrational*. Take it from here.

39. Recall the notation "(a, b)" for the *open interval* of *reals* or *rationals* as the case may be, that is, $(a, b) \stackrel{Def}{=} \{x \in \mathbb{R} : a < x < b\}$ or $(a, b) \stackrel{Def}{=} \{x \in \mathbb{Q} : a < x < b\}$ respectively.[18]
A real —such as $\sqrt{2}$ that is not *rational* is called *irrational*.

40. Start with the POset $(\mathbb{R}, <)$. Consider next the sets of rationals $S = \{x \in \mathbb{Q} : x < \sqrt{2}\}$ and $T = \{x \in \mathbb{Q} : x > \sqrt{2}\}$. Prove two things:

a. T has no smallest (rational, of course, element).
Hint. Use the "calculus 101" fact that every interval of reals (a, b) contains some rational number (hence, infinitely many; why "hence"?).
b. While $\sqrt{2}$ is trivially the lub of S *if we include irrational numbers in the set of upper bounds*, on the other hand *if we insist on rational upper bounds only* (all those are in T), then there is NO least.

41. Prove that if $(A, <)$ is a LOset (linearly ordered set), then any pair a, b of members of A has a least upper bound. Explain how such an lub can be found/constructed in each case.

42. (Knaster-Tarski Fixpoint theorem) Let $(A, <)$ be a POset with a minimum element m and $f : A \to A$ be a total *continuous function*, which in the context of POsets *means* that lub and function applications (calls) *commute*, that is, whenever $lub(X)$ exists for $\emptyset \neq X \subseteq A$, then so does $lub(f[X])$ and $f(lub(X)) = lub(f[X])$.

[18] We are reminded that $\mathbb{R}$ stands for the set of all real numbers and $\mathbb{Q}$ for the set of all rational numbers. Of course, $\mathbb{Q} \subsetneq \mathbb{R}$ as we know from "calculus 101".

Assume now that *every nonempty totaly ordered by* "$<$" *subset of* A —such a subset is often called a "*chain*"— has a lub.
Prove that there is an $a \in A$ such that $f(a) = a$. Such an a is called a *fixed point* or *fixpoint* of f.
Hints.

- Prove that f is increasing on A, that is, if $f(x) \downarrow$ and $f(y) \downarrow$ and $x \leq y$, then $f(x) \leq f(y)$ (*Hint*. What is $lub(\{x, y\})$?).
- Define inductively the sequence —that is, function $\lambda n.a_n$ from $\mathbb{N}$ to A— below:

$$a_0 = m$$
$$a_{n+1} = f(a_n)$$

- Prove by induction that $\lambda n.a_n$ is total and increasing, that is, $a_n \leq a_{n+1}$, for $n \geq 0$.
- Let a be defined to stand for $lub\Big(\{a_0, a_1, a_2, \ldots\}\Big)$, that is, $lub(\text{ran}(\lambda n.a_n))$.

  Prove $f(a) = a$. *Hint* for this bullet. Show that $lub\Big(\{a_0, a_1, a_2, \ldots\}\Big) = lub\Big(\{a_1, a_2, \ldots\}\Big)$.

43. Prove that the fixpoint you found above is *least*. That is, if c is any other member of A such that $f(c) = c$, then $a \leq c$.

44. (*An Application of* 42 *to Computability*) Let $\mathcal{P}(\mathbb{N} : \mathbb{N})$ denote the set of all 1-argument functions from $\mathbb{N}$ to $\mathbb{N}$.
Let $\mathscr{F}$ be a *total* function $\mathscr{F} : \mathcal{P}(\mathbb{N} : \mathbb{N}) \to \mathcal{P}(\mathbb{N} : \mathbb{N})$. Such a function is called an *operator*.
Now equip $P(\mathbb{N} : \mathbb{N})$ with the order $\subset$ to obtain the POset of unary functions under the *inclusion* (subset) order.

Prove

a. $\Big(P(\mathbb{N} : \mathbb{N}), \subset\Big)$ has the properties given *abstractly* to $(A, <)$ in Exercise 42 above.
b. *Assume* that the total operator $\mathscr{F} : \mathcal{P}(\mathbb{N} : \mathbb{N}) \to \mathcal{P}(\mathbb{N} : \mathbb{N})$ is continuous. Prove that $\mathscr{F}$ has a least fixpoint $\alpha \in \mathcal{P}(\mathbb{N} : \mathbb{N})$.

   Note. In computability theory $\mathscr{F}$ is assumed to be computable (in some mathematically appropriate sense). Then, provably, so is its least fixpoint. This result, due to Kleene, known as (a special case of his) first recursion theorem (cf. Tourlakis (2022)) has extensive applications in computability, but also in the area of *program semantics*.

45. The *greatest common divisor* —*acronym gcd*— let us call it "d", of two nonzero integers a and b is the *largest* positive *common divisor* of the two. We write $d = gcd(a, b)$.
Prove that if $d = gcd(a, b)$, then for some *integers* (members of $\mathbb{Z} = \{\ldots, -1, 0, 1, \ldots\}$) x, y we have $d = ax + by$.
Hint. Prove that the set $S = \{ax + by : x \in \mathbb{Z} \wedge y \in \mathbb{Z}\}$ has positive members. Call d the *smallest such positive member* and *prove* $d = gcd(a, b)$. To this end,

a. Prove that we may write $d = ax + by$ for the smallest positive member of S. (Trivial)
b. Every common divisor of a and b divides d (Trivial)
c. d divides a and b. If not it will be, say, $a = dq + r$ for some q and $0 < r < d$. Derive a contradiction by showing that $r = aX + bY$ for some X and Y in $\mathbb{Z}$. Similarly for b.

46. If $ab \neq 0$ and $1 = gcd(a, b)$, then we say that a and b are *relatively prime*.
Prove that if $1 = gcd(a, b)$ and $a \mid bc$ —where as before, $x \mid y$ means that $y = xq$ for some q— then $a \mid c$.
Hint. Use 45 above.

47. Refer to Example 5.2.21. Generalise said example to any base. Thus, let $1 < b \in \mathbb{N}$. Prove that every natural number $n \geq 0$ is expressible base-b as an expression

$$n = a_m b^m + a_{m-1} b^{m-1} + \cdots + a_1 b + a_0 \tag{1}$$

$$\text{where each } a_i \text{ satisfies } 0 \leq a_i < b \tag{2}$$

Hint. Use CVI.

48. Write an algorithm as a, say, *pseudo* C program,[19] which will convert a number $n \geq 0$ given base-10 to base-b.
Hint. Due to (1) above,

$$\begin{aligned} n &= a_m b^m + a_{m-1} b^{m-1} + \cdots + a_1 b + a_0 \\ &= \left(a_m b^{m-1} + a_{m-1} b^{m-2} + \cdots + a_1\right) b + a_0 \end{aligned}$$

thus we can obtain a_0 by noting the remainder of the division of n by b. Your algorithm ought to work from right to left to get the sequence $a_0, a_1, \ldots a_m$ by repeating the preceding observation.

49. Demonstrate your algorithm above by converting 131 (this is expressed in *decimal* or base-10 notation) to *binary* (or base-2) notation.

[19] "Pseudo" means to *not* be too faithful to programming language syntax, and shortcuts in notation are allowed if they do not introduce ambiguities.

50. (**An ancient theorem**[20]) Prove that there are infinitely many primes. Note that this will be a proof by contradiction. Induction is not relevant.

Hint. Suppose instead that there only finitely many primes, *exactly these* $n + 1$:

$$p_0, p_1, \ldots, p_n$$

Consider as Euclid did their product plus 1:

$$Q = p_0 \times p_1 \times \ldots \times p_n + 1$$

We have *two cases*: One, Q is prime and Two Q is not prime.
Trivially show that the first case implies a contradiction right away and then invoke 5.2.20 for the second case to get, again, a contradiction. Done. **Right?** *Fill in all the "blanks".*

[20] Euclid.

Inductively Defined Sets; Structural Induction

6

Overview

This chapter introduces a generalisation of the *definitions by induction* (recursion) of the last section. Here we define *sets* inductively, not functions. The associated proof tool —*induction along an inductive definition*, or *structural induction* — of *properties* of inductively defined sets is introduced and validated.

We also connect the *inductive definition* of sets with an appropriate *iterative* construction *by stages* and also we connect it (in the chapter's Exercises section) with the definition of sets as *monotone operator fixpoints* (see 3.8.1).

6.1 Set Closures

An example of an inductively defined set is the following.

Suppose you want to define *by finite means*, and do so *precisely*, the set of all *"simple" arithmetical expressions* that use the numbers 1, 2, 3, the operations $+$ and $\times$, and round brackets (but nothing else). Then you would do it like this:

The set of said *simple arithmetical expressions* is the *smallest* set ($\subseteq$-smallest) that

1. Contains each of 1, 2 and 3.
2. If it contains expressions E and E', then it also contains $(E + E')$ and $(E \times E')$.

Some folks would add a 3rd requirement "*nothing else is in the set unless so demonstrated using 1. 2. above*" and omit "smallest". *Really*?

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0_6

How *exactly* would you so "demonstrate"? In a recursive definition you ought to be able to make your recursive calls and not have to *trace back* why the object you constructed *exists*!

We will prove in Sect. 6.3.5 that indeed there *is* an iterative way to show that a *particular* simple arithmetic expression was formed correctly by our recursion, but that defeats the beauty of recursion.

> Besides, until we reach said section we don't even *know* what "nothing else is in the set unless so demonstrated using 1. 2. above" *means* or *how* to "use" 1. and 2. to do it!

So it is nonsense to stick such a statement in the bottom of the definition as a (redundant) afterthought.

Before we get to the general definitions, let us finesse our construction and propose some terminology.

(a) First off, in step 1. above we say that 1, 2 and 3 are *the initial objects* of our recursive/inductive definition.

(b) In step 2. we say that $(E + E')$ is obtained by an *operation* (on strings) that is available to us, depicted as a "blackbox" below, which we named "+".

$$\begin{array}{l} E \\ \longrightarrow \\ \longrightarrow \\ E' \end{array} \boxed{+} \longrightarrow (E + E')$$

In words, the operation *concatenates from left to right the strings*

"(", *the string named by* "E", "+", *the string named by* "E'", and ")"

Similar comments for the operation "×".

$$\begin{array}{l} E \\ \longrightarrow \\ \longrightarrow \\ E' \end{array} \boxed{\times} \longrightarrow (E \times E')$$

(c) Both operations in this example are single-valued, that is, functions. It is preferable to be slightly more general and allow *operations* that are just relations, but not necessarily functions. Such an operation $O(x_1, \ldots, x_n, y)$ is n-ary —n inputs, $x_1, \ldots, x_n$— with output variable y.

(d) We say that a set of objects S is *closed under* a *relation* (operation) —it could also be a function— $O(x_1, \ldots, x_n, y)$ meaning that for *all* input values $x_1, \ldots, x_n$ in S, *all* the obtained values y are also in S.

We are ready for the general definition:

6.1.1 Definition Given a set of *initial objects* $\mathcal{I}$ and a *set* of *operations* $\mathcal{O} = \{O_1, O_2, O_2, \ldots\}$, the object $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ is called the *closure of* $\mathcal{I}$ *under* $\mathcal{O}$ —or the set *inductively defined by the pair* $(\mathcal{I}, \mathcal{O})$— and *denotes* the $\subseteq$-*smallest set*[1] S that satisfies

1. $\mathcal{I} \subseteq S$.
2. S is *closed under all operations in* $\mathcal{O}$, or simply, *closed under* $\mathcal{O}$ or even $\mathcal{O}$-closed.
3. The "smallest" part: Any *set* T that satisfies 1. and 2. also satisfies $S \subseteq T$.

The set $\mathcal{O}$ may be infinite. *Each operation* O_i *is a set.* □

Nice definition, but does the *set* $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ *exist* given *any* $\mathcal{I}$ and $\mathcal{O}$? Yes. But first,

6.1.2 Theorem *For any choice of* $\mathcal{I}$ *and* $\mathcal{O}$*, if* $Cl(\mathcal{I}, \mathcal{O})$ *exists, then it is unique.*

Proof Say the definition of $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ *ambiguously* —i.e., may have more than one value— leads to two classes, S and T.

Then, letting S pose as closure, we get $S \subseteq T$ from 6.1.1, 3.

Then, letting T pose as closure, we get $T \subseteq S$, again from 6.1.1, 3. Thus $S = T$. □

6.1.3 Theorem *For any choice of* $\mathcal{I}$ *and* $\mathcal{O}$ *with the restrictions of Definition 6.1.1 the* set $Cl(\mathcal{I}, \mathcal{O})$ exists.

Proof We have to check and note a few things.

1. By 3.1.5, for each O_i, $\mathrm{ran}(O_i)$ is a set (because the O_i is).
2. The class $F = \{\mathrm{ran}(O_i) : i = 1, 2, 3 \ldots\}$ is a set. This is so by **Principle** 3, since I can index all members of F by assigning unique indices from $\mathbb{N}$ to each of its members (and $\mathbb{N}$ is a set by **Principle** 0).
3. By 2. above and 2.4.17, $\bigcup F$ is a set, and so is $T = \mathcal{I} \cup \bigcup F$ □
4. T contains $\mathcal{I}$ as a subset (by the way T was defined) and is $\mathcal{O}$-closed since any O_i-output —no matter where the *inputs* come from— is in $\mathrm{ran}(O_i) \subseteq \bigcup F$.
5. The family $\mathbb{G} = \{S : \mathcal{I} \subseteq S \wedge S \text{ is } \mathcal{O}\text{-closed}\}$ contains the set T as a member. Thus (cf. 2.4.18)
$$C \stackrel{Def}{=} \left(\bigcap \mathbb{G}\right) \subseteq T$$
is a *set* by the *subclass theorem* (2.3.6).

[1] We will learn that it *is* actually a *set*.

Since all sets S in $\mathbb{G}$ contain $\mathcal{I}$ and are $\mathcal{O}$-closed, so is C (Verify). That is, C satisfies 1.–2. of 6.1.1. But also $C \subseteq S$ *for all such sets* S the way it is defined. So it satisfies 6.1.1, 3. as well; it is $\subseteq$-smallest.

We proved existence: $C = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. □

6.2 Induction Over a Closure

6.2.1 Definition Let a pair $(\mathcal{I}, \mathcal{O})$ be given as above.

We say that a property $P[x]$ *propagates with* $\mathcal{O}$ *iff* for each $O_i(x_1, \ldots, x_n, y) \in \mathcal{O}$, *if whenever all the inputs in the* x_i *satisfy* $P[x]$ (i.e., $P[x_i]$ is true for each argument x_i), then all output values returned by y —for said inputs— satisfy $P[x]$ as well. Recall that for each assignment of values to the inputs $x_1, \ldots, x_n$ we may have more than one output values in y; for all such values $P[y]$ is true. □

6.2.2 Lemma *For all* $(\mathcal{I}, \mathcal{O})$ *and a property* $P[x]$*, if the latter propagates with* $\mathcal{O}$*, then the class* $\mathbb{A} = \{x : P[x]\}$ *is closed under* $\mathcal{O}$ *(is* $\mathcal{O}$*-closed).*

Proof So let $O_i(x_1, \ldots, x_n, y) \in \mathcal{O}$. Let $a_1, \ldots, a_n$ be all in $\mathbb{A}$. Thus

$$P[a_i], \text{ for all } i = 1, \ldots, n$$

By assumption, if $O_i(a_1, \ldots, a_n, b)$, then $P[b]$ is true, hence $b \in \mathbb{A}$. □

6.2.3 Theorem (Induction Over a Closure Principle) *Let* $Cl(\mathcal{I}, \mathcal{O})$ *and a property* $P[x]$ *be given. Suppose we have done the following steps:*

1. *We showed that for each* $a \in \mathcal{I}$, $P[a]$ *is true.*
2. *We showed that* $P[x]$ *propagates with* $\mathcal{O}$.

Then every $a \in Cl(\mathcal{I}, \mathcal{O})$ *has property* $P[x]$.

Naturally, the technique encapsulated by 1. and 2. of 6.2.3 is called "induction over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$" or "structural induction" over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$.

Note that for *each* $O_i \in \mathcal{O}$ the "propagation of property $P[x]$" will take the form of an I.H. followed by an I.S.:

- **Assume** for the unspecified fixed inputs $a_1, \ldots, a_n$ of O_i that *all* satisfy $P[x]$. This is the *I.H.* for O_i.

- Then **prove** that any output b of O_i *caused by said input* also satisfies the property.

Proof (of 6.2.3) Let us write

$$\mathbb{A} \stackrel{Def}{=} \{x : P[x]\}$$

Thus, 1. in 6.2.3 translates to

$$\mathcal{I} \subseteq \mathbb{A} \qquad (*)$$

2. in 6.2.3 yields by the Lemma

$$\mathbb{A} \text{ is } \mathcal{O}\text{-closed} \qquad (**)$$

Now we cannot directly apply 6.1.1 and say "by $(*)$ and $(**)$ we have"

$$\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \subseteq \mathbb{A}$$

because in 6.1.1 the "*sets* T" that fulfil "1. and 2." must be, well, *sets*; not proper classes.

Here is the workaround: $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ contains $\mathcal{I}$ and is $\mathcal{O}$-closed. By $(*)$ and $(**)$ so does

$$T = \mathrm{Cl}(\mathcal{I}, \mathcal{O}) \cap \mathbb{A} \qquad (***)$$

But T is a set by 2.3.6 and thus

$$\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \stackrel{6.1.1}{\subseteq} T \stackrel{(***)}{\subseteq} \mathbb{A}$$

The last inclusion immediately translates to

$$\boxed{x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O}) \text{ implies } P[x] \text{ is true}} \qquad \square$$

6.2.4 Example Let $S = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where $\mathcal{I} = \{0\}$ and $\mathcal{O}$ contains just one operation, $x + 1 = y$, where y is the output variable. That is,

$$n \longrightarrow \boxed{x + 1 = y} \longrightarrow n + 1 \qquad (1)$$

is our only operation. By induction over S, I can show $S \subseteq \mathbb{N}$.

The "$P[x]$" here is "$x \in \mathbb{N}$".

So $P[0]$ is true. I verified the property for $\mathcal{I}$. That the property propagates with our operation is captured by (1) above (if $n \in \mathbb{N}$, then $n + 1 \in \mathbb{N}$). Done!

Can we show also $\mathbb{N} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$? **Yes**: In this direction I do SI over $\mathbb{N}$ on variable n. The property, let's call it $Q[x]$, now is "$x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$".

For $n = 0$, $n \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ since $0 \in \mathcal{I} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ by 6.1.1.

Now, say (*I.H.*) $n \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. Since $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ is closed under the operation $x + 1 = y$, we have $n + 1 \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ by 6.1.1.

So,

$$\mathrm{Cl}(\mathcal{I}, \mathcal{O}) = \mathbb{N} \qquad \square$$

Thus the induction over a closure generalises *SI*. The direction $\mathbb{N} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ can be also proved directly by a result in the new section.

6.3 Closure Versus Definition by Stages

We will see in this section that there is also a *by-stages* or *by-steps* way to obtain $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$.

6.3.1 Definition (Derivations) An $(\mathcal{I}, \mathcal{O})$-*derivation* —or just *derivation* if we know which $(\mathcal{I}, \mathcal{O})$ we are talking about— is a *finite sequence of objects*

$$d_1, d_2, d_3, \ldots, d_i, \ldots, d_n \tag{1}$$

satisfying:

Each d_i is

1. A member of $\mathcal{I}$,
   or
2. For some j, one of the results of $O_j(x_1, \ldots, x_k, y)$ with inputs $a_1, \ldots, a_k$ that are found in the derivation (1) *to the* left *of* d_i.

n is called the *length of the derivation*. Every d_i in (1) is called an $(\mathcal{I}, \mathcal{O})$-*derived* object, or just *derived*, if the $(\mathcal{I}, \mathcal{O})$ is understood. $\square$

Clearly, the concept of a derivation abstracts, thus generalises, the concept of *proof*, while a derived object abstracts the concept of a *theorem*.

6.3.2 Example For the $(\mathcal{I}, \mathcal{O})$ of 6.2.4, here are some derivations:

$$0$$

$$0, 0, 0$$

$$0, 1, 0, 1, 0, 1, 1, 1, 1, 0$$

Nothing says we cannot repeat a d_i in a derivation! Lastly here is an "efficient" derivation with no redundant steps: 0, 1, 2, 3, 4, 5. $\square$

6.3.3 Proposition *If*

$$d_1, d_2, d_3, \ldots, d_i, \ldots, d_n, d_{n+1}, \ldots, d_m$$

is a $(\mathcal{I}, \mathcal{O})$*-derivation, then so is*

$$d_1, d_2, d_3, \ldots, d_i, \ldots, d_n$$

Proof Each d_i is validated in a derivation either outright (i.e., is in $\mathcal{I}$) or by looking to the left! What we may want to remove to the *right* of d_i does not affect the validity of that entry. □

6.3.4 Proposition *If* $d_1, d_2, \ldots, d_n$ *and* $e_1, e_2, \ldots, e_m$ *are* $(\mathcal{I}, \mathcal{O})$*-derivations, then so is*

$$d_1, d_2, \ldots, d_n, e_1, e_2, \ldots, e_m$$

Proof Traversing $d_1, d_2, \ldots, d_n$ and $e_1, e_2, \ldots, e_m$ in

$$d_1, d_2, \ldots, d_n, e_1, e_2, \ldots, e_m$$

from left to right we validate each d_i and each e_j giving precisely the same validation *reason* as we would in each sequence $d_1, d_2, \ldots, d_n$ and $e_1, e_2, \ldots, e_m$ separately. These reasons are local to each sequence. □

We now prove that defining a set S as a $(\mathcal{I}, \mathcal{O})$-closure is equivalent with defining S as the set of all $(\mathcal{I}, \mathcal{O})$-derived objects.

6.3.5 Theorem *For any initial sets of objects and operations on objects* ($\mathcal{I}$ *and* $\mathcal{O}$) *we have that* $Cl(\mathcal{I}, \mathcal{O}) = \{x : x \text{ is } (\mathcal{I}, \mathcal{O})\text{-derived}\}$.

Proof Let us write $D = \{x : x \text{ is } (\mathcal{I}, \mathcal{O})\text{-derived}\}$ and prove that $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) = D$. We have two directions:

1. $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \subseteq D$: By induction over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$. The property to prove is "$x \in D$".

   - Let $x \in \mathcal{I}$. Then x is derived via the one-member derivation

     $$x$$

     So $x \in D$. Thus all $x \in \mathcal{I}$ have the property.
   - The property "$x \in D$" propagates with each $O_k(\vec{x}_n, y) \in \mathcal{O}$: So let each of the x_i have a derivation $\boxed{\ldots, x_i}$. We show that so does y.
     Concatenating all these derivations we get a derivation (6.3.4)

$$\boxed{\ldots, x_1}, \ldots, \boxed{\ldots, x_i}, \ldots, \boxed{\ldots, x_n} \tag{1}$$

But then so is

$$\boxed{\ldots, x_1}, \ldots, \boxed{\ldots, x_i}, \ldots, \boxed{\ldots, x_n},\ y \tag{2}$$

by 6.3.1, case 2. That is, y is *derived*, hence $y \in D$ is proved (I.S.).

2. Conversely, prove that $D \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$: Let $x \in D$. This time we do good old-fashioned CVI over $\mathbb{N}$ on the length n of a derivation of x, toward showing that $x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ —this is the "property of x" that we prove.
Basis. $n = 1$. The only way to have a 1-element derivation is that $x \in \mathcal{I}$.
Thus, $x \in \mathcal{I} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ by 6.1.1.
I.H. Assume the claim for x derived with length $k < n$.
I.S. Prove that the claim holds when x has a derivation of length n.
Consider such a derivation

$$\begin{array}{r} a_n \\ \| \\ a_1, \ldots a_i, \ldots, a_k, \ldots, \ x \end{array}$$

If $x \in \mathcal{I}$, then we are done by the *Basis*. Otherwise, say x is the result of an operation (relation) $O_r \in \mathcal{O}$, *applied on entries to the left of* x, that is, say that $O_r(\ldots, x)$ is true —where we did not (have to) specify the inputs.
By the I.H. the inputs of O_r all are *all* in $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$. Now, since this closure is closed under $O_r(\ldots, x)$, we have that the output x is in $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ too. □

6.3.6 Remark So now we have two *equivalent* (6.3.5) approaches to defining inductively defined sets S: As $S = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ or as $S = \{x : x \text{ is } (\mathcal{I}, \mathcal{O})\text{-derived}\}$.

The first approach is best when you want to prove properties of all members of the set S. The second is best when you want to show $x \in S$, for some specific x. □

6.3.7 Example Let us revisit Example 6.2.4, second half of the proof. To prove $\mathbb{N} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ we prove that each $n \in \mathbb{N}$ has a $(\mathcal{I}, \mathcal{O})$-derivation.

Indeed, such a derivation for n is

$$0, 1, 2, \ldots, n-1, n \tag{1}$$

where the above is $(n) \geq = \{x \in \mathbb{N} : x \leq n\}$ where all entries in (1) are in ascending order without repetitions. □

6.3.8 Example Let $A = \{a, b\}$. We call A an "alphabet".

Let $\mathcal{I} = \{\lambda\}$, λ being (the name of) the *empty string*. Let us denote string concatenations by putting the strings we want to concatenate next to each other. E.g., concatenate *aaa* and *bbbaa* to obtain *aaabbbaa*. Also, if X denotes a string, and so does Y, then XY denotes the concatenation of the strings (denoted by) X and Y in that order. Similarly, Xa means

the result of concatenating string named X with the (length-1) string a, in that order. The *length* of a string over A is the *number of occurrences* in the string (counting repetitions) of a and b.

We denote by A^+ the set of all strings of non zero length formed using the symbols a and b. A^* is defined to be $A^+ \cup \{\lambda\}$. Let $\mathcal{O}$ consist of the operations O_a and O_b:

$$X \longrightarrow \boxed{O_a} \longrightarrow Xa \tag{1}$$

and

$$X \longrightarrow \boxed{O_b} \longrightarrow Xb \tag{2}$$

We claim that $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) = A^*$.

1. For $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \subseteq A^*$ we do induction over the closure to prove that any $x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ satisfies $x \in A^*$ ("the property").

   - Well, if $x \in \mathcal{I}$ then $x = \lambda$. But $\lambda \in A^*$.
   - The property propagates with each of O_a and O_b. For example, if $X \in A^*$, then since Xa is also a string over the alphabet A, we have $Xa \in A^*$. Similarly for O_b. Done.

2. For $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \supseteq A^*$ we do induction over $\mathbb{N}$ on $n = |Y|$ —the length of Y— to prove that any $Y \in A^*$ satisfies $Y \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ ("the property").

   - Basis. $n = 0$. Then $Y = \lambda \in \mathcal{I} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. Done.
   - *I.H.* **Assume** claim for fixed n.
   - *I.S.* **Prove** for $n + 1$. If $|Y| = n + 1$ then $Y = Xa$ or $Y = X'b$ for some X or X' of length n. Say, it is $Y = Xa$. By I.H. $X \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. But since $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ is $\mathcal{O}$-closed, we have $Y = Xa \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ by (1). The $Y = X'b$ case is entirely similar. □

6.3.9 Example Let $A = \{a, b\}$ again.

Let $\mathcal{I} = \{\lambda\}$, let $\mathcal{O}$ consist of one operation R:

$$X \longrightarrow \boxed{R} \longrightarrow aXb \tag{3}$$

We claim that $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) = \{a^n b^n : n \geq 0\}$, where for any string X,

$$X^n \stackrel{Def}{=} \underbrace{XX \ldots X}_{n \text{ copies of } X}$$

If $n = 0$, "0 copies of X" means λ.

Let us write $S = \{a^n b^n : n \geq 0\}$.

1. For $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \subseteq S$ we do induction over the closure to prove that any $x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ satisfies $x \in S$ ("the property").

   - Well, if $x \in \mathcal{I}$ then $x = \lambda = a^0 b^0$. Done.
   - The property propagates with each of R. For example, say $x = a^n b^n \in S$. Using (3) we see that the output, axb, is $a^{n+1}b^{n+1} \in S$. The property does propagate! Done.

2. For $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \supseteq S$ we do induction over $\mathbb{N}$ on n of $x = a^n b^n$ (arbitrary member of S) to prove that any $x \in S$ satisfies $x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ ("the property").

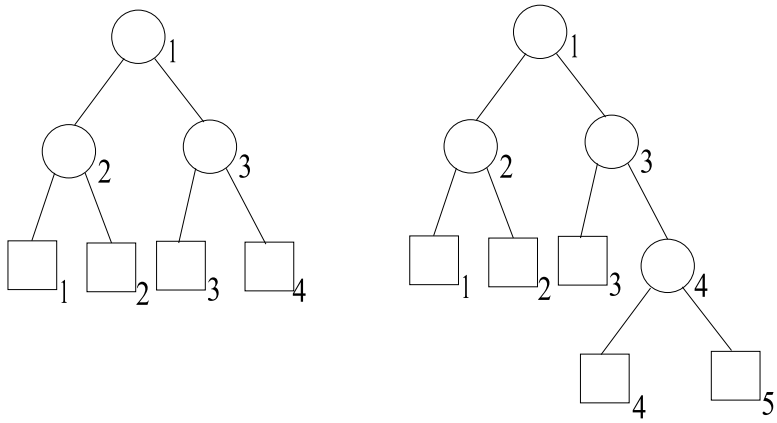
   - Basis. $n = 0$. Then $x = \lambda \in \mathcal{I} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. Done.
   - *I.H.* **Assume** claim for fixed n.
   - *I.S.* **Prove** for $n + 1$. Thus $x = a^{n+1}b^{n+1} = aa^n b^n b$. By the I.H., $a^n b^n \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. By (3) —recall that $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ is $\mathcal{O}$-closed— we get the output $aa^n b^n b = a^{n+1}b^{n+1} \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. □

6.3.10 Example (Extended Binary Trees) This is a longish example with some preliminary discussion up in front. We want to define the mathematical (and computer science) term known as "Tree".

This term refers to a structure, which uses as building blocks —called **nodes**— the members of the enumerable set below

$$A = \{\bigcirc_0, \bigcirc_1, \bigcirc_2, \ldots; \square_0, \square_1, \square_2, \ldots\}$$

Trees look something like this:



The qualifier "extended" is due to the presence of square nodes. We will not define simple trees (they have round nodes only).

These *nodes* are made *distinct* by the use of *subscripts*. The symbols in the set A are *distinguished* by their *type*, "round" versus "square", and *within each type* by their natural

number index. Thus, $\bigcirc_i \neq \bigcirc_j$ iff $i \neq j$, $\square_i \neq \square_j$ iff $i \neq j$, and $\bigcirc_i \neq \square_j$, *for all* i, j.

One feature in both of the above drawings is essential to note:

Circular or *square nodes* are connected by line segments. Walking in the vertical direction from the top of the page towards the bottom, *no nodes are ever shared.* In particular, in all the examples above where we have more than one node, you will notice that

the two *sets* of nodes that "hang below" the top node (left and right of it) are *disjoint*.

We need to include this requirement in our definition.

But clearly these sets of notes have "geometric structure" (*position*: left/right; and *connections*: via line segments)! They are not "flat" sets like $\{\bigcirc_5, \square_{11}\}$.

And yet, in the *mathematical definition below* we will need to *state* the boxed condition: the left and right, when you "forget" the lines and positions, *become disjoint flat sets*. This observation is what *imposes* some complexity in the definition, which defines the "structure" *and* the "flat" set that supports the structure (the set of nodes in the tree) *simultaneously*.

We define an *extended binary tree* as a *member* of the inductively defined *set of e-trees*. It is intended that each e-tree of the inductively defined set of all trees is an *ordered pair*:

(flat set of its nodes, geometric tree structure)

The "geometric tree structure" can be *mathematically pictured* in a *one-dimensional depiction* of the trees.

For example, the first tree in the figure above is linearly represented by the (ordered) triple below, whose first and third components are also ordered triples.

$$\Big(\big(\square_1, \bigcirc_2, \square_2\big), \bigcirc_1, \big(\square_3, \bigcirc_3, \square_4\big)\Big)$$

The "flat set" of (round) nodes of the above is $\{\bigcirc_1, \bigcirc_2, \bigcirc_3\}$.

Thus our definition below builds the flat set —called the *support* of the tree— of nodes of a tree *at the same time as it builds the structure of the tree*.

6.3.11 Definition We define the *set* of *all extended trees* —or just *trees*— ET, as $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where:

1. First, chose as the set of initial objects

$$\mathcal{I} = \Big\{(\emptyset, \square_0), (\emptyset, \square_1), (\emptyset, \square_2), \ldots\}$$

2. $\mathcal{O}$ has just one rule with a constraint on the input: If $F_X \cap F_Y = \emptyset$ **and** $\bigcirc_i \notin F_X \cup F_Y$, then

$$\left.\begin{array}{c}(F_X, X) \longrightarrow \\ \bigcirc_i \longrightarrow \\ (F_Y, Y) \longrightarrow\end{array}\right\} \boxed{\text{form tree}} \longrightarrow \Big(F_X \cup F_Y \cup \{\bigcirc_i\},\ (X, \bigcirc_i, Y)\Big)$$

3. For *each* $(S, T) \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ we say that T *is an extended tree*, and S *is its* support, that is, the "flat" set of round nodes used to build T.[2]

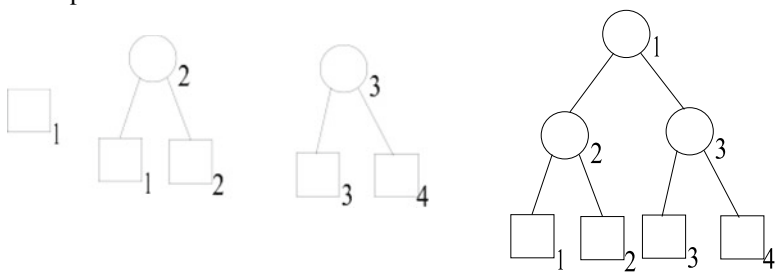
   We indicate this relationship by

$$S = \mathrm{sup}(T)^{3}$$

If $T = (X, \bigcirc_i, Y)$, then we say that $\bigcirc_i$ *is the* **root** *of* T, *while* X *is its* **left** *and* Y *is its* **right subtree**.

□

Some examples of trees are



We verify the example above: Using 6.3.5, the leftmost example is a tree since it is the right component of the pair $(\emptyset, \square_1)$. The next tree is built via the derivation —written linearly,

$$(\emptyset, \square_1),\ (\emptyset, \square_2),\ \Big(\{\bigcirc_2\},\ (\square_1, \bigcirc_2, \square_2)\Big)$$

The next derivation builds both the 2nd and 3rd trees:

$$(\emptyset, \square_1),\ (\emptyset, \square_2),\ \Big(\{\bigcirc_2\},\ (\square_1, \bigcirc_2, \square_2)\Big),\ (\emptyset, \square_3),\ (\emptyset, \square_4),\ \Big(\{\bigcirc_3\},\ (\square_3, \bigcirc_3, \square_4)\Big)$$

The 4th tree has this as a derivation:[4]

$$(\emptyset, \square_1),\ (\emptyset, \square_2),\ \Big(\{\bigcirc_2\}, (\square_1, \bigcirc_2, \square_2)\Big),\ (\emptyset, \square_3),\ (\emptyset, \square_4),\ \Big(\{\bigcirc_3\},\ (\square_3, \bigcirc_3, \square_4)\Big),$$
$$\Big(\{\bigcirc_1, \bigcirc_2, \bigcirc_3\},\ \Big(\{\bigcirc_2\},\ (\square_1, \bigcirc_2, \square_2)\Big),\ \bigcirc_1,\ \Big(\{\bigcirc_3\},\ (\square_3, \bigcirc_3, \square_4)\Big)\Big)$$

The support of the 4th tree is the flat set $\{\bigcirc_1, \bigcirc_2, \bigcirc_3\}$. □

[2] We may fix *a priori* a supply (set) A of acceptable round nodes.

[3] As for many other symbols, "sup" means something else in the context of POsets.

[4] Derivations are not unique as is clear from Example 6.3.2.

6.3.12 Example (Trees —continued) Hmm! Seems like we are not including square nodes in the support. See how the *support* of all nodes in $\mathcal{I}$ is $\emptyset$ for each entry. Why so?

In the words of Knuth (Knuth (1973)) "trees is the most important nonlinear structure arising in computing algorithms". The extended tree is an abstraction of trees that we implement with computer programs, where round nodes are the *only ones that can carry data*. The lines are (implicitly) pointing downwards. They are *pointers*, in computer jargon. For example, the topmost leftmost line in the fourth tree above points to the node $\bigcirc_2$. Practically it means that if your program is processing node $\bigcirc_1$, then it can transfer to and process node $\bigcirc_2$ if it wishes. It knows the address of $\bigcirc_2$. The pointer holds this address as value.

Which brings me to square nodes! Together with the line planted on them, they are notation for *null* pointers! They point nowhere. So square nodes cannot hold information, *that is why they do not contribute to the support of the tree*.

The computer scientist calls round nodes "internal" and calls square nodes "external".

Finally, how do the lines —called *edges*— get inserted? We defined "root" for trees, as well as "left subtree" and "right subtree". So, to draw lines and draw a tree that is given mathematically as $(X, \bigcirc_r, Y)$, we *call* recursively the process that does the "drawing" on (inputs) X and Y.

Then add two more edges: One from $\bigcirc_r$ to the root of X and one from $\bigcirc_r$ to the root of Y.

How does the recursion terminate? Well, if your tree is just $\square_j$, then there is nothing to draw. $\square_j$ is the root. This is the basis of the recursive procedure: *do nothing*. □

Here is something interesting about all extended trees:

6.3.13 Proposition *In any extended tree, the number of square nodes exceeds by one the number of round nodes.*

Proof Induction over the set of all trees (6.3.11) $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$.

1. *Basis*. For any $(\emptyset, \square_i)$, the tree-part (structure-part) is just $\square_i$. One square node, 0 round nodes. Done.
2. The property propagates with the only tree-builder operation:

$$\left.\begin{array}{c}(F_X, X) \longrightarrow \\ \bigcirc_i \longrightarrow \\ (F_Y, Y) \longrightarrow\end{array}\right\} \boxed{\text{form tree}} \longrightarrow \Big(F_X \cup F_Y \cup \{\bigcirc_i\},\ (X, \bigcirc_i, Y)\Big)$$

Indeed, *suppose* that X has ϕ internal (round) and ε external (square) nodes. Let also Y have ϕ' internal and ε' external nodes.

The assumption *on the input side* is then (I.H.) that

$$\phi + 1 = \varepsilon \tag{1}$$

and

$$\phi' + 1 = \varepsilon' \tag{2}$$

The *output side* of the operation has the tree $(X, \bigcirc_i, Y)$. This has $\Phi = \phi + \phi' + 1$ internal nodes and $E = \varepsilon + \varepsilon'$ external ones. Using (1) and (2) we have

$$\Phi = \varepsilon + \varepsilon' - 1 = E - 1$$

Seeing that this is the property we want to prove on the output side, indeed the property propagates with the rule. Done. □

We will have more to say about trees in Chap. 8.

6.4 Exercises

1. (**Long but Easy Exercise**) Below we *simultaneously* define the *syntax* of **a set of names** of certain sets of strings and the *semantics* of said names —that is, *what* sets they *name*.
The set of names is given *as a closure* while the semantics of those names is given along the definition of the closure, inductively. See below.
For the names we need an alphabet of symbols. As such we take the alphabet $\Sigma = \{0, 1, (,), \cdot, +, *\}$ by the inductive definition:

Names Form	**Semantics (as *subsets* of** $\{0, 1\}^*$**)**
$\emptyset$	$\emptyset$
0	$\{0\}$
1	$\{1\}$

If α and β are strings among those we are defining here, with meanings $A \subseteq \{0, 1\}^*$ and $B \subseteq \{0, 1\}^*$ respectively, then so are

$(\alpha + \beta)$	$A \cup B$
$(\alpha \cdot \beta)$	$A \cdot B (= \{x \cdot y : x \in A \wedge y \in B\})$
(α^*)	A^*

These strings are called *regular expressions*, and the sets they are "naming" are called *regular sets*. For example, $(0 + 1)$ is a regular expression for the regular set $\{0, 1\}$, while $(\emptyset^*)$ is a regular expression for $\{\lambda\}$, where λ denotes the empty string (in this context "λ" is *not* related to λ-notation). We *informally* omit brackets (so that we can write $0 + 1, \emptyset^*$), by the rules:

a. Omit outermost brackets
b. The strength of operations is (from strongest to weakest), $*$, $\cdot$, $+$.

Prove by induction on the definition of *regular expressions*, **that if** A **is a** *regular set*, **then so is** $\{x : x \cdot y \in A$ **for some y** $\in \{\mathbf{0}, \mathbf{1}\}^*\}$.

Hint.
You need to show (by induction on regular expressions) that any such set of prefixes, $pref(A)$ to use a name for convenience, can be **named** by a regular expression.
Basis. $\emptyset$ names the *set* $\emptyset$. Clearly, $pref(\emptyset) = \emptyset$, hence the prefix has a "name", $\emptyset$. 0 names $\{0\}$. $pref(\{0\}) = \{\lambda, 0\}$. This has a name (using simplified notation, without all brackets) $\emptyset^* + 0$, so is a regular set. Similarly for regular set named "1".
I.H. Assume that if α, β name A, B respectively, that $pref(A)$, $pref(B)$ have names (i.e., are regular) γ, δ respectively.
I.S. What about $\alpha + \beta$ that names $A \cup B$? Well, $pref(A \cup B) = pref(A) \cup pref(B)$ which has name $\gamma + \delta$ hence is regular.
Your work starts here:

▶ Now work out the cases where $\alpha \cdot \beta$ names $A \cdot B$ and α^* names A^*.

2. Prove that for each choice of $\mathcal{I}, \mathcal{O}$ we can define a monotone operator (cf. 3.8.1) Φ such that $\Phi^\infty = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. The notation "$\Phi^\infty$" was introduced in Definition 3.8.4.

3. Show that the transitive closure P^+ of a *set* relation P is $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ for appropriate $\mathcal{I}, \mathcal{O}$. Specify $\mathcal{I}$ and $\mathcal{O}$ and prove they work for P.

4. Show that, for each $\mathcal{I}$ and $\mathcal{O}$, the part $\mathcal{I}$ can be absorbed by $\mathcal{O}$ —its members viewed as zero-ary operations without premises (inputs). Thus an $\mathcal{O}'$ exists such that $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) = \bigcap \left\{ S : S \text{ is } \mathcal{O}'\text{-closed} \right\}$.
Note. Authors adopting 0-premise rules thus write their closures as "$\mathrm{Cl}(\mathcal{O})$" where $\mathcal{O}$ contains 0-premise rules.

5. (*The Syntax of Boolean Formulas,* #1) *Boolean formulas* are defined as $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where $\mathcal{I} = \{\bot, \top, p, p', p'', p''', \ldots\}$ and $\{p, p', p'', \ldots\}$ is the *enumerable* set of Boolean variables while $\bot, \top$ are the two Boolean *constants* which, for simplicity, we have identified in the body of this book (in Chap. 4) with their *intended values* in the metatheory, namely, **f**, **t**.
The members of $\mathcal{I}$ are called *atomic Boolean formulas*.
There are two operations on *strings* in $\mathcal{O}$, namely,

$$A \longrightarrow \boxed{\neg} \longrightarrow (*\neg * A*)$$

$$\begin{matrix} A \longrightarrow \\ B \longrightarrow \end{matrix} \boxed{\vee} \longrightarrow (*A * \vee * B*)$$

where "$*$" denotes *string concatenation*.

a. How many brackets do we need to write $\bot$ or p' correctly?
b. Prove by induction over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ that every Boolean formula has as many left as it has right brackets.

6. (*The Syntax of Boolean Formulas,* #2) Prove by induction over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ (5 above) that every Boolean formula has as many left brackets as *Boolean connectives*.

7. (*The Syntax of Boolean Formulas, #3*) Prove by induction over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ (5 above) that every Boolean formula contains an atomic one as a substring.

8. (*The Syntax of Boolean Formulas, #4*) A *prefix* of a string of symbols X is a string U for which we have a string V such that $X = U * V$ (or, simply, $X = UV$). A prefix U of X is *proper* by definition iff $X \neq U$.
Prove by induction over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ (5 above) that, *for every* Boolean formula A, *every one* of its nonempty ($\neq \lambda$) *proper prefixes* contains an excess of left brackets.

9. (*Lack of Ambiguity or "Unique Readability"; Immediate Predecessors*) A Boolean formula A is either *atomic* or $(\neg B)$ or $(B \vee C)$ where B, C are formulas (cf. 5 above). In the first of the latter two cases we call B an *immediate predecessor* (or *i.p.*) of A, while in the second case we say that both B and C are i.p. of A. On the other hand, *an atomic formula has no i.p.*
If *for every formula A* the *i.p. are uniquely determined*, then we say that the pair $(\mathcal{I}, \mathcal{O})$ is *unambiguous*, else it is *ambiguous*.
Prove that the rule set in 5 is unambiguous. We say that this is the "unique readability" *metatheorem* for Boolean formulas.
Hint. Use Problem 8.

10. Let $S = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where $\mathcal{I} = \{2\}$ and $\mathcal{O}$ contains only $\lambda xy.x + y$.
Prove that $S = \{2n : n \geq 1\}$. There are two directions: $\subseteq$ and $\supseteq$.

11. Let $T = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where $\mathcal{I} = \{2\}$ and $\mathcal{O}$ contains only $\lambda xy.x + y$ and $\lambda xy.x - y$.
Prove that $S = \{2n : n \in \mathbb{Z}\}$. There are two directions: $\subseteq$ and $\supseteq$.

12. Consider $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where $\mathcal{I} = \{(0, 0)\}$ and $\mathcal{O}$ contains only $\lambda(x, y).(x + 3, y + 2)$ where both x and y are in $\mathbb{N}$.
Prove that for all $(n, m) \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$, 5 divides $n + m$.

13. Consider $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where $\mathcal{I} = \{(0, 0)\}$ and $\mathcal{O}$ contains only the two rules $\lambda(x, y).(x + 1, y)$ and $\lambda(x, y).(x, y + 1)$ where both x and y are in $\mathbb{N}$.
Prove that $\mathbb{N} \times \mathbb{N} = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$.
Caution. Do not forget the $\subseteq$ direction.

14. Use Simple Induction to prove that if h and g are total and if f is defined from them by primitive recursion, that is,

For all $x, \vec{y}$,

$$f(0, \vec{y}) = h(\vec{y})$$
$$f(x + 1, \vec{y}) = g(x, \vec{y}, f(x, \vec{y}))$$

then f is total as well. Incidentally, we use the notation $f = prim(h, g)$ for the f defined above and call h the *basis function* and g the *iteration function*.
Hint. Prove by simple induction on x that, for all $\vec{y}$, we have $f(x, \vec{y}) \downarrow$.

Very Important! Some of the variables in f, g and h above may be missing! This is alright. Permutation of the variables is also alright. We have indicated all the variables as place-holders in the general case. The following function (Exercise 15) has a primitive recursive definition where the last variable of "g" is missing so there is no "recursive call"!

15. (*The switch function*) Prove that $\lambda xyz.if\ x = 0\ then\ y\ else\ z$ is in $\mathcal{PR}$.

16. Let $f = prim(h, g)$. Imagine a programming language that allows the assignment statements $z \leftarrow h(\vec{y})$ and $z \leftarrow g(x, \vec{y}, w)$. Program in this programming language, using a *single do loop*, the function $\lambda x\vec{y}.f(x, \vec{y})$ given by the primitive recursion in 14.
Hint. You will obviously use *pseudo-programming*, as details of the programming language are not essential. The crucial part is that it supports the above mentioned assignment statement and it can do "loops":
do $i = 0$ **to** n
$\left\{\right.$
$\vdots$
$\left.\right\}$

17. True or false? In the schema defining f as $f = prim(h, g)$ the recursive call can be eliminated.

18. Define the set of *all* primitive recursive functions, that we denote by $\mathcal{PR}$, as a closure $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where

a. $\mathcal{I}$ is the set of initial functions. These are precisely $S = \lambda x.x + 1$, $Z = \lambda x.0$, and U_i^n, for $1 \leq i \leq n > 0$, given by $U_i^n = \lambda \vec{x}_n x_i$.
b. There are just two operations in $\mathcal{O}$:

   i. From functions $\lambda \vec{x} z \vec{y}.f(\vec{x}, z, \vec{y})$ and $\lambda \vec{w}.g(\vec{w})$ obtain —as we say *by substitution*

$$H = \lambda \vec{x} \vec{w} \vec{y}.F(\vec{x}, g(\vec{w}), \vec{y})$$

   ii. From functions h, g obtain $f = prim(h, g)$

Prove that all the functions of $\mathcal{PR}$ are total.
Hint. Towards proving "$f \in \mathcal{PR}$ implies that f is total" (by induction over the closure), the case that an application of the operation $prim$ propagates totalness relies on the previous exercise.

19. (*Easy*) True or false and why?

a. If h and g are in $\mathcal{PR}$ then so is $f = prim(h, g)$
b. $\lambda\vec{x}z\vec{y}.f(\vec{x}, z, \vec{y})$ and $\lambda\vec{w}.g(\vec{w})$ are in $\mathcal{PR}$ then so is $H = \lambda\vec{x}\vec{w}\vec{y}.F(\vec{x}, g(\vec{w}), \vec{y})$.

20. Prove that

a. $\lambda x.x + 3 \in \mathcal{PR}$.
b. $\lambda xy.x + y \in \mathcal{PR}$.
c. $\lambda x.x + x \in \mathcal{PR}$.
d. $\lambda x.2^x \in \mathcal{PR}$.
e.

$$\lambda x.2^{2^{\cdot^{\cdot^{2}}}} \Big\} x \text{ 2's}$$

is in $\mathcal{PR}$.

Hint. The last function is obtained trivially as $prim(h, g)$ where h is the constant function that outputs 1 and g is one of the previous functions in this exercise (of course, you *must* provide justification).

21. (**The Formal Natural Numbers**) Let $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ be such that $\mathcal{I} = \{\emptyset\}$ and $\mathcal{O}$ contains only the following operation on sets

$$\lambda x.x \cup \{x\}^{5}$$

Modern set theorists call this closure ω, the set of *formal natural numbers*. Correspondingly, they call the members of ω *formal natural numbers*.

We will see (actually you will prove) below (30) that $\emptyset \in \omega$ is the counterpart of $0 \in \mathbb{N}$

and $n \cup \{n\} \in \omega$ is the counterpart of $n + 1 \in \mathbb{N}$. Naturally, "$n \cup \{n\}$" is called the (formal) *successor* of $n \in \omega$.

Let us now discover properties of the formal natural numbers —the first three of which

are the *sets* $\emptyset$, $\{\emptyset\}$, $\{\emptyset, \{\emptyset\}\}$, etc.—

Pause. Can you better *specify* the "etc." above? ◀

that mirror exactly those of the members of $\mathbb{N}$.

Now, *Prove* (easy) that we can *do* induction over ω on the variable n, namely:

If $P(n)$ is a property of sets n, then if one verifies $P(\emptyset)$ and also, for the arbitrary n, verifies that $P(n)$ implies $P\Big(n \cup \{n\}\Big)$, then one has proved that $P(n)$ is true for all $n \in \omega$.

22. Prove that $n \cup \{n\} \neq \emptyset$ for all $n \in \omega$.

---

[5] λ-notation was introduced in 3.5.10.


This corresponds to "$n + 1 \neq 0$" for $\mathbb{N}$.


23. Prove that $n \cup \{n\} = m \cup \{m\}$ implies $m = n$ for all n, m in ω.
Hint. Say instead that m, n exist such that $m \neq n$ and yet $n \cup \{n\} = m \cup \{m\}$.


This corresponds to "$n + 1 = m + 1$ implies $n = m$" on $\mathbb{N}$.
The reader will note that a proof along the Hint is valid for *all* sets n, m not just those in ω.


24. Prove if $n \in \omega$, then either $n = \emptyset$ or $n = m \cup \{m\}$ for some $m \in \omega$. We say, in the latter case, that n is a successor.
Hint. Do induction over ω on $n \in \omega$ to prove $(\forall n)P(n)$, where $P(n)$ stands for $n = \emptyset \vee (\exists m \in \omega)(n = m \cup \{m\})$.

25. (*Transitive classes*) A class $\mathbb{A}$ is *transitive* iff $x \in y \in \mathbb{A}$ implies $x \in \mathbb{A}$ for all x, y. This can be also said as $y \in \mathbb{A}$ implies $y \subseteq \mathbb{A}$. Prove that every member of ω is a transitive set.
Hint. Use induction over ω on the variable z to prove $(\forall z)(\forall x)(\forall y)(x \in y \in z \rightarrow x \in z)$.

26. Prove that ω is a transitive set.
Hint. Use induction over ω on the variable y to prove $(\forall y)(\forall x)(x \in y \in \omega \rightarrow x \in \omega)$.

27. Prove that ω is *not* a successor.

28. Prove that if n is a formal natural number then every one of its members is a natural number.

29. Prove that the (proper) subset relation $n \subset m$ on formal natural numbers is a well ordering on ω.
Hint. Organise your thoughts on this by listing first what exactly "well ordering" entails as a property of $\subset$.

30. Prove that $\mathbb{N} \sim \omega$.
Hint. Define by induction (recursion) $f : \mathbb{N} \rightarrow \omega$ by $f(0) = \emptyset$ and $f(n + 1) = f(n) \cup \{f(n)\}$. Show that f is total, 1-1 and onto.

Recurrence Equations and Their Closed-Form Solutions

7

Overview

In so-called "divide and conquer" algorithms one usually ends up with a recurrence relation (i.e., *inductive or recursive definition*!) that defines the "timing function", $T(n)$ —such timing indicating worst case upper bound on run time or average run time as the case may be. For example, the recurrence might look like

$$T(n) = \begin{cases} 1 & \textbf{if } n = 1 \\ T(n/2) + 1 & \textbf{otherwise} \end{cases}$$

In order to assess the "goodness" of the proposed algorithm by comparison to either our expectations or to another algorithm, we need to know $T(n)$ in "closed" form, that is, in terms of known functions, for example, n^r for $r > 0$, c^n for $c > 1$, $\log_b n$ for some integer $b > 1$.

Often, a preliminary analysis need only worry about the "asymptotic behaviour" of the algorithm, i.e., the behaviour for *large* inputs (n is the input *size*).

What *is* input size? Since many algorithms of interest —e.g., they may manipulate trees— are *non numerical*, "size" is not the numerical value of the input normally. Moreover, even numerical algorithms are often expressed in terms of the digit-structure of the inputs thus it makes sense to assess their "goodness" with respect to the *number of digits* in the input or the *length of the input*, not its value. Does it matter? It does in the context of the so-called *efficient* (or "feasible") algorithms which is defined to mean that their run time is bounded by a polynomial function of the input size!

It turns out that —due to the exponential relation between value and length of a natural number— an algorithm that runs in polynomial time with respect to the input numerical *value*

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0_7

will run in exponential time with respect to input *length* and thus be termed "inefficient" or worse: "unfeasible".

"Big-O" notation —introduced in this chapter— is an excellent tool in gauging upper bounds of run times of algorithms, therefore the solution of recurrences is often sought in such notation. On occasion one requires an "exact" solution (this is much harder to achieve in general).

There is a big variety of recurrence relations and an equally big variety of solution techniques. Some restricted cases are handled well by packages such as *Mathematica* or *Maple*.

In this chapter we restrict attention to simple classes of recurrences taken from both the "additive" and "multiplicative" cases. These characterisations in quotes refer to the manner of handling the argument of the recurrence. E.g., the recurrence above is multiplicative as the recursive call is to an argument obtained by *halving* the original argument n.

For the solution of the Fibonacci recurrence and other "Fibonacci-like" recurrences in closed form we introduce the topic of *generating functions*.

7.1 Big-O, Small-o, and the "Other" ∼

This notation is due to the mathematician E. Landau and is in wide use in number theory, but also in computer science in the context of measuring (bounding above) computational complexity of algorithms for all "very large inputs".

7.1.1 Definition Let f and g be two total functions of one variable, where $g(x) > 0$, for all x. Then

1. $f = O(g)$ —also written as $f(x) = O(g(x))$— read "f is big-oh g", means that there are positive constants C and K in $\mathbb{N}$ such that

$$x > K \text{ implies } |f(x)| \leq Cg(x)$$

2. $f = o(g)$ —also written as $f(x) = o(g(x))$— read "f is small-oh g", means that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0$$

3. $f \sim g$ —also written as $f(x) \sim g(x)$— read "f is of the same order as g", means that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$$

□

"$\sim$" between two sets A and B, as in $A \sim B$, means that there is a 1-1 correspondence $f : A \to B$. Obviously, the context will protect us from confusing this $\sim$ with the one introduced just now, in 7.1.1.

Both definitions 2. and 3. require some elementary understanding of differential calculus. Case 2. says, intuitively, that as x gets extremely large, then the fraction $f(x)/g(x)$ gets extremely small, infinitesimally close to 0. Case 3. says, intuitively, that as x gets extremely large, then the fraction $f(x)/g(x)$ gets infinitesimally close to 1; that is, the function outputs are infinitesimally close to each other.

7.1.2 Example 1. $x = O(x)$ since $x \leq 1 \cdot x$ for $x \geq 0$.
2. $x \sim x$, since $x/x = 1$, and stays 1 as x gets very large.
3. $x = o(x^2)$ since $x/x^2 = 1/x$ which trivially goes to 0 as x goes to infinity.
4. $2x^2 + 1000^{1000}x + 10^{350000} = O(x^2)$. Indeed

$$\frac{2x^2 + 1000^{1000}x + 10^{350000}}{3x^2} = 2/3 + 1000^{1000}/3x + 10^{350000}/3x^2 < 1$$

for $x > K$ for some well chosen K. Note that $1000^{1000}/3x$ and $10^{350000}/3x^2$ will each be $< 1/6$ for all sufficiently large x-values: we will have $2/3 + 1000^{1000}/3x + 10^{350000}/3x^2 < 2/3 + 1/6 + 1/6 = 1$ for all such x-values. Thus $2x^2 + 1000^{1000}x + 10^{350000} < 3x^2$ for $x > K$ as claimed.
In many words, in a polynomial, the order of magnitude is determined by the highest power term. □

The last example motivates

7.1.3 Proposition *Suppose that g is as in 7.1.1 and $f(x) \geq 0$ for all $x > L$, hence $|f(x)| = f(x)$ for all $x > L$. Now, if $f(x) \sim g(x)$, then $f(x) = O(g(x))$.*

Proof The assumption says that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$$

From "calculus 1" (1st year differential calculus) we learn that this implies that for some K, $x > K$ entails

$$\left| \frac{f(x)}{g(x)} - 1 \right| < 1$$

hence

$$-1 < \frac{f(x)}{g(x)} - 1 < 1$$

therefore, $x > \max(K, L)$ implies $f(x) < 2g(x)$. □

7.1.4 Proposition *Suppose that g is as in 7.1.1 and $f(x) \geq 0$ for all $x > L$, hence $|f(x)| = f(x)$ for all $x > L$. Now, if $f(x) = o(g(x))$, then $f(x) = O(g(x))$.*

Proof The assumption says that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0$$

From calculus 1 we learn that this implies that for some K, $x > K$ entails

$$\left|\frac{f(x)}{g(x)}\right| < 1$$

hence

$$-1 < \frac{f(x)}{g(x)} < 1$$

therefore, $x > \max(K, L)$ implies $f(x) < g(x)$. □

These two propositions enrich our toolbox:

7.1.5 Example 1. $\ln x = o(x^r)$ for any positive real r. Here "ln" stands for $\log_e$ where e is the Euler constant

$$2.71828182845904523536028747135266249775724709 37\ldots$$

Seeing that both numerator and denominator

$$\lim_{x\to\infty} \frac{\ln x}{x^r}$$

go to ∞, we have here (if we do not do anything to mitigate) an impasse: We have a "limit" that is "indeterminate":

$$\frac{\infty}{\infty}$$

So, we will use "l'Hôpital's rule" (the limit of the fraction is equal to the limit of the fraction of the derivatives):

$$\lim_{x\to\infty} \frac{\ln x}{x^r} = \lim_{x\to\infty} \frac{1/x}{rx^{r-1}} = \lim_{x\to\infty} \frac{1}{rx^r} = 0$$

2. $\ln x = O(\log_{10}(x))$. In fact, you can go from one log-base to the other:

$$\log_e(x) = \frac{\log_{10}(x)}{\log_{10}(e)}$$

The claim follows from 7.1.3 since trivially $\ln x \sim \log_{10}(x)/\log_{10}(e)$. For that reason —and since multiplicative constants are hidden in big-O notation— complexity- and

algorithms-practitioners omit the base of the logarithm and write things like $O(\log n)$ and $O(n \log n)$. □

7.2 Solving Recurrences; the Additive Case

The general case here is of the form[1]

$$\begin{aligned} T_0 &= k \\ s_n T_n &= v_n T_{n-1} + f(n) \textbf{ if } n > 0 \end{aligned}$$

a recurrence defining the *sequence* T_n, or equivalently, the *function* $T(n)$ (both jargons and notations spell out the same thing), in terms of the *known* functions (sequences) s_n, v_n, $f(n)$.

For the general case see Knuth (1973). Here we will restrict attention to the case $s_n = 1$, for all n, and also $v_n = a$ (a constant), for all n.

Subcase 1. ($a = 1$) Solve

$$\begin{aligned} T_0 &= k \\ T_n &= T_{n-1} + f(n) \textbf{ if } n > 0 \end{aligned} \tag{1}$$

From (1), $T_n - T_{n-1} = f(n)$, thus

$$\sum_{i=1}^{n} (T_i - T_{i-1}) = \sum_{i=1}^{n} f(i)$$

the lower summation value dictated by the lowest valid value of $i - 1$ according to (1).

7.2.1 Remark The summation in the lhs above is called a "*telescoping (finite) series*" because the terms $T_1, T_2, \ldots, T_{n-1}$ appear both positively and negatively and pairwise cancel. Thus the series "contracts" into $T_n - T_0$ like a (hand held) telescope. □

Therefore

$$\begin{aligned} T_n &= T_0 + \textstyle\sum_{i=1}^{n} f(i) \\ &= k + \textstyle\sum_{i=1}^{n} f(i) \end{aligned} \tag{2}$$

If we know how to get the sum in (2) in closed form, then we solved the problem!

7.2.2 Example Solve

$$p_n = \begin{cases} 2 & \textbf{if } n = 1 \\ p_{n-1} + n & \textbf{otherwise} \end{cases} \tag{3}$$

Here

[1] Note the "additivity" in the relation between indices/arguments: n versus $n - 1$.

$$\sum_{i=2}^{n}(p_i - p_{i-1}) = \sum_{i=2}^{n} i$$

Note the lower bound of the summation: It is here 2, to allow for the lowest $i - 1$ value possible. That is 1 according to 3, hence $i = 2$.

Thus,

$$p_n = 2 + \frac{(n+2)(n-1)}{2}$$

(Where did I get the $(n + 2)(n - 1)/2$ from?) The above answer is the same as (verify!)

$$p_n = 1 + \frac{(n+1)n}{2}$$

obtained by writing

$$2 + \sum_{i=2}^{n} i = 1 + \sum_{i=1}^{n} i$$

Subcase 2. ($a \neq 1$) Solve

$$\begin{aligned} T_0 &= k \\ T_n &= aT_{n-1} + f(n) \textbf{ if } n > 0 \end{aligned} \tag{4}$$

(4) is the same as

$$\frac{T_n}{a^n} = \frac{T_{n-1}}{a^{n-1}} + \frac{f(n)}{a^n}$$

To simplify notation, set

$$t_n \stackrel{\text{Def}}{=} \frac{T_n}{a^n}$$

thus the recurrence (4) becomes

$$\begin{aligned} t_0 &= k \\ t_n &= t_{n-1} + \frac{f(n)}{a^n} \textbf{ if } n > 0 \end{aligned} \tag{5}$$

By subcase 1, this yields

$$t_n = k + \sum_{i=1}^{n} \frac{f(i)}{a^i}$$

from which

$$T_n = ka^n + a^n \sum_{i=1}^{n} \frac{f(i)}{a^i} \tag{6}$$

7.2.3 Example As an illustration solve the recurrence below.

$$T_n = \begin{cases} 1 & \textbf{if } n = 1 \\ 2T_{n-1} + 1 & \textbf{otherwise} \end{cases} \tag{7}$$

To avoid trouble, note that the lowest term here is T_1, hence its "translation" to follow the above methodology will be "$t_1 = T_1/2^1 = 1/2$". So, the right hand side of (6) applied here will have "ka^{n-1}" instead of "ka^n" (Why?) and the indexing in the summation will start at $i = 2$ (Why?)

Thus, by (6),

$$\begin{aligned} T_n &= 2^n(1/2) + 2^n \sum_{i=2}^{n} \frac{1}{2^i} \\ &= 2^{n-1} + 2^n\left(\frac{(2^{-1})^{n+1} - 1}{2^{-1} - 1} - 1 - \frac{1}{2}\right) \\ &= 2^{n-1} + 2^n(2 - 2^{-n} - 1 - \frac{1}{2}) \\ &= 2^n - 1 \end{aligned}$$

In the end you will probably agree that it is easier to redo the work with (7) directly, first translating it to

$$t_n = \begin{cases} 1/2 & \textbf{if } n = 1 \\ t_{n-1} + 1/2^n & \textbf{if } n > 1 \end{cases} \tag{8}$$

rather than applying (6)!

We immediately get from (8)

$$T_n = 2^n t_n = 2^n\left(1/2 + \sum_{i=2}^{n} 1/2^i\right) = 2^n\left(1/2 + \frac{(2^{-1})^{n+1} - 1}{2^{-1} - 1} - 1 - 1/2\right)$$

etc.

The red terms are subtracted as they are missing from our $\sum$. The blue formula used is for

$$\sum_{i=0}^{n} 1/2^i$$

□

7.3 Solving Recurrences; the Multiplicative Case

Subcase 1.

$$T(n) = \begin{cases} k & \textbf{if } n = 1 \\ aT(n/b) + c & \textbf{if } n > 1 \end{cases} \tag{1}$$

were a, b are positive integer constants ($b > 1$) and k, c are any constants. Recurrences like (1) above arise in *divide and conquer* solutions to problems. For example, *binary search* has timing governed by the above recurrence with $b = 2, a = c = k = 1$.

Why does (1) with the above-mentioned parameters —$b = 2, a = c = k = 1$— capture the run time of binary search? First off, regarding "run time" let us be *specific*: we mean number of comparisons.

OK, to do such a search on a sorted (in ascending order, say) array of length n, you first check the mid point (for a match with what you are searching for). If you found what you want, exit. If not, you know (due to the ordering) whether you should next search the left half or the right half.

So you call the procedure recursively on an array of length about $n/2$.

This decision *and* call took $T(n/2) + 1$ comparisons. This equals $T(n)$. If the array has length 1, then you spend just one comparison, $T(1) = 1$.

We seek a general solution to (1) in big-O notation.

First convert to an "additive case" problem: To this end, seek a solution in the *restricted* set $\{n \in \mathbb{N} : n = b^m \text{ for some } m \in \mathbb{N}\}$. Next, set

$$t(m) = T(b^m) \tag{2}$$

so that the recurrence becomes

$$t(m) = \begin{cases} k & \textbf{if } m = 0 \\ at(m-1) + c & \textbf{if } m > 0 \end{cases} \tag{3}$$

hence, from the work in the previous section,

$$\sum_{i=1}^{m} \left(\frac{t(i)}{a^i} - \frac{t(i-1)}{a^{i-1}} \right) = c \sum_{i=1}^{m} a^{-i}$$

therefore

$$t(m) = a^m k + ca^m \begin{cases} m & \textbf{if } a = 1 \\ a^{-1} \dfrac{(a^{-1})^m - 1}{a^{-1} - 1} & \textbf{if } a \neq 1 \end{cases}$$

or, more simply,

$$t(m) = \begin{cases} k + cm & \textbf{if } a = 1 \\ a^m k + c \dfrac{a^m - 1}{a - 1} & \textbf{if } a \neq 1 \end{cases}$$

Using O-notation, and going back to T we get:

$$T(b^m) = \begin{cases} O(m) & \textbf{if } a = 1 \\ O(a^m) & \textbf{if } a \neq 1 \end{cases} \tag{4}$$

or, *provided we remember that this solution relies on the assumption that n* has the form b^m:

$$T(n) = \begin{cases} O(\log n) & \textbf{if } a = 1 \\ O(a^{\log_b n}) & \textbf{if } a \neq 1 \end{cases} = \begin{cases} O(\log n) & \textbf{if } a = 1 \\ O(n^{\log_b a}) & \textbf{if } a \neq 1 \end{cases} \tag{5}$$


If $a > b$ then we get slower than linear "run time" $O(n^{\log_b a})$ with $\log_b a > 1$. If on the other hand $b > a > 1$ then we get a sublinear run time, since then $\log_b a < 1$.



Now a very important observation. For functions $T(n)$ that are *increasing*,[2] i.e., $T(i) \leq T(j)$ if $i < j$ the restriction of n to have form b^m proves to be *irrelevant* in obtaining the solution. The solution is still given by (5) *for all n*. Here's why:

In the general case, n satisfies

$$b^{m-1} < n \leq b^m \text{ for some } m \geq 0 \tag{6}$$

Suppose now that $a = 1$ (upper case in (4)). We want to establish that $T(n) = O(\log n)$ for the general n (of (6)). By monotonicity of T and the second inequality of (6) we get

$$T(n) \overset{\text{by (6) right}}{\leq} T(b^m) \overset{\text{by (4)}}{=} O(m) \overset{\text{by (6) left}}{=} O(\log n)$$

The case where $a > 1$ is handled similarly. Here we found an answer $O(n^r)$ (where $r = \log_b a > 0$) *provided* $n = b^m$ (some m). Relax this proviso, and assume (6).

Now

$$T(n) \overset{\text{by (6) right}}{\leq} T(b^m) \overset{\text{by (4)}}{=} O(a^m) = O((b^m)^r) \overset{\text{Why?}}{=} O((b^{m-1})^r) \overset{\text{by (6) left}}{=} O(n^r)$$

Subcase 2.

$$T(n) = \begin{cases} k & \textbf{if } n = 1 \\ aT(n/b) + cn & \textbf{if } n > 1 \end{cases} \tag{1'}$$

were a, b are positive integer constants ($b > 1$) and k, c any constants. Recurrences like (1′) above also occur in divide and conquer solutions to problems. For example, *two-way merge sort* has timing governed by the above recurrence with $a = b = 2$ and $c = 1/2$. Quicksort has *average* run time governed, essentially, by the above with $a = b = 2$ and $c = 1$. Both lead to $O(n \log n)$ solutions. Also, *Karatsuba's "fast" integer multiplication* has a run time recurrence as above with $a = 3, b = 2$.

[2] Such are the "complexity" or "timing" functions of algorithms.

These examples are named for easy look up, in case they trigger your interest or curiosity. It is not in the design of this course to expand on them. Merge Sort and Quicksort you might see in a course on data structures while Karatsuba's "fast multiplication" of natural numbers might appear in a course on algorithms.

Setting at first (our famous *initial* restriction on n) $n = b^m$ for some $m \in \mathbb{N}$ and using (2) above we end up with a variation on (3):

$$t(m) = \begin{cases} k & \textbf{if } m = 0 \\ at(m-1) + cb^m & \textbf{if } m > 0 \end{cases} \tag{3'}$$

thus we need do

$$\sum_{i=1}^{m} \left(\frac{t(i)}{a^i} - \frac{t(i-1)}{a^{i-1}} \right) = c \sum_{i=1}^{m} (b/a)^i$$

therefore

$$t(m) = a^m k + ca^m \begin{cases} m & \textbf{if } a = b \\ (b/a)\dfrac{(b/a)^m - 1}{b/a - 1} & \textbf{if } a \neq b \end{cases}$$

Using O-notation, and using cases according as to $a < b$ or $a > b$ we get:

$$t(m) = \begin{cases} O(b^m m) & \textbf{if } a = b \\ a^m O(1) = O(a^m) & \textbf{if } b < a \\ O(b^m - a^m) = O(b^m) & \textbf{if } b > a \end{cases} \qquad /^* \ (b/a)^m \to 0 \text{ as } m \to \infty \ ^*/$$

or, in terms of T and n, which is *restricted* to form b^m (using same calculational "tricks" as before):

$$T(n) = \begin{cases} O(n \log n) & \textbf{if } a = b \\ O(n^{\log_b a}) & \textbf{if } b < a \\ O(n) & \textbf{if } b > a \end{cases} \tag{4'}$$

The above solution is valid for *any* n without restriction, *provided* T is increasing. The proof is as before, so we will not redo it (you may wish to check the "new case" $O(n \log n)$ as an exercise).

In terms of complexity of algorithms, the above solution says that in a divide and conquer algorithm (governed by $(1')$) we have the following cases:

- The total size of all subproblems we solve (recursively) is *equal* to the original problem's size. Then we have a $O(n \log n)$ algorithm (e.g., merge sort).
- The total size of all subproblems we solve is *more* than the original problem's size. Then we go worse than linear ($\log_b a > 1$ in this case). An example is Karatsuba multiplication

that runs in $O(n^{\log_2 3})$ time —still better than the $O(n^2)$ so-called "school method" integer multiplication, which justifies the nickname "fast" for Karatsuba's multiplication.[3]

- The total size of all subproblems we solve is *less* than the original problem's size. Then we go in linear time (e.g., the problem of finding the k-th smallest in a *set* of n elements).

7.4 Generating Functions

We saw some simple cases of recurrence relations with additive and multiplicative index structure (we reduced the latter to the former). Now we turn to a wider class of additive index structure problems where our previous technique of utilizing a "telescoping sum"

$$\sum_{i=1}^{n}(t(i) - t(i-1))$$

does not apply because the right hand side still refers to $t(i)$ for some $i < n$. Such is the case of the well known Fibonacci sequence F_n given by

$$F_n = \begin{cases} 0 & \textbf{if } n = 0 \\ 1 & \textbf{if } n = 1 \\ F_{n-1} + F_{n-2} & \textbf{if } n > 1 \end{cases}$$

The method of *generating functions* that solves this harder problem also solves the previous problems we saw.

Here's the method in *outline*. We will then embark on a number of fully worked out examples.

Given a recurrence relation

$$t_n = \ldots t_{n-1} \ldots t_{n-2} \ldots t_{n-3} \ldots \tag{1}$$

with the appropriate "starting" (initial) conditions. We want t_n in "closed form" in terms of known functions. Here are the steps:

1. Define a *generating function* of the *sequence* $t_0, t_1, \ldots, t_n, \ldots$

$$\begin{aligned} G(z) &= \textstyle\sum_{i=0}^{\infty} t_i z^i \\ &= t_0 + t_1 z + t_2 z^2 + \cdots + t_n z^n + \cdots \end{aligned} \tag{2}$$

[3] But there are even faster integer multiplication algorithms!

(2) is a *formal power series*, where *formal* means that we only are interested in the *form* of the "infinite sum" and *not* in any issues of convergence[4] (therefore "meaning") of the sum. It is stressed that our disinterest in convergence matters is *not* a simplifying *convenience* but it is due to the fact that convergence issues are *irrelevant* to the problem in hand!


In particular, we will *never* have to consider values of z or make substitutions into z.


2. Using the recurrence (1), find a *closed form* of $G(z)$ as a function of z (this *can* be done *prior* to knowing the t_n in closed form!)
3. Expand the closed form $G(z)$ back into a power series

$$\begin{aligned} G(z) &= \textstyle\sum_{i=0}^{\infty} a_i z^i \\ &= a_0 + a_1 z + a_2 z^2 + \cdots + a_n z^n + \cdots \end{aligned} \tag{3}$$

But now we *do have* the a_n's in terms of known functions, because we know $G(z)$ in closed form! We only need to compare (2) and (3) and proclaim

$$t_n = a_n \quad \text{for } n = 0, 1, \ldots$$

The problem has been solved.

Steps 2. and 3. embody all the real work. We will illustrate by examples how this is done in practice, but first we need some "tools":

Let us concentrate below on the "boxed" results, which we will be employing —not being too interested in the arithmetic needed to obtain them!


The Binomial Expansion. For our purposes in this volume we will be content with just one tool, the "binomial expansion theorem" of *calculus* (the *finite* version of it we proved by induction here Example 5.2.25):

For any $m \in \mathbb{R}$ (where $\mathbb{R}$ is the set of real numbers) we have

$$\begin{aligned} (1+z)^m &= \sum_{r=0}^{\infty} \binom{m}{r} z^r \\ &= \cdots + \binom{m}{r} z^r + \cdots \end{aligned} \tag{4}$$

where for any $r \in \mathbb{N}$ and $m \in \mathbb{R}$

[4] In Calculus one learns that power series converge in an interval like $|z| < r$ for some real $r \geq 0$. The $r = 0$ case means the series diverges for *all* z.

$$\binom{m}{r} \stackrel{\text{def}}{=} \begin{cases} 1 & \textbf{if } r = 0 \\ \dfrac{m(m-1)\cdots(m-[r-1])}{r!} & \textbf{otherwise} \end{cases} \tag{5}$$

The expansion (4) terminates with last term

$$\binom{m}{m} z^m \stackrel{\text{by (5)}}{=} z^m$$

as the "binomial theorem of *Algebra*" says, and that is so **iff** m is a *positive integer*. In *all* other cases (4) is non-terminating (infinitely many terms) and the formula is then situated in Calculus. As we remarked before, we will not be concerned with when (4) converges.

Note that (5) gives the familiar

$$\begin{aligned} \binom{m}{r} &= \frac{m(m-1)\cdots(m-[r-1])}{r!} \\ &= \frac{m(m-1)\cdots(m-[r-1])(m-r)\cdots 2\cdot 1}{r!(m-r)!} \\ &= \frac{m!}{r!(m-r)!} \end{aligned}$$

when $m \in \mathbb{N}$. *In all other cases we use* (5) because if $m \notin \mathbb{N}$, then "$m!$" is meaningless.

Let us record the very useful special case when m is a *negative integer*, $-n$ $(n > 0)$.

$$\begin{aligned} (1+z)^{-n} &= \cdots + \frac{-n(-n-1)\cdots(-n-[r-1])}{r!} z^r + \cdots \\ &= \cdots + (-1)^r \frac{n(n+1)\cdots(n+[r-1])}{r!} z^r + \cdots \\ &= \cdots + (-1)^r \frac{(n+[r-1])\cdots(n+1)n}{r!} z^r + \cdots \\ &= \cdots + (-1)^r \binom{n+r-1}{r} z^r + \cdots \end{aligned} \tag{6}$$

$$(1-z)^{-n} = \cdots + \binom{n+r-1}{r} z^r + \cdots \tag{7}$$

Finally, let us record in "boxes" some important special cases of (6) and (7)

$$\begin{aligned} (1-z)^{-1} = \frac{1}{1-z} &= \cdots + \binom{r}{r} z^r + \cdots \\ &= \cdots + z^r + \cdots \end{aligned} \tag{8}$$

The above is the familiar "converging geometric progression" (converging for $|z| < 1$, that is, *but this is the last time I'll raise* irrelevant *convergence issues*). Two more special cases of (6) will be helpful:

$$\begin{aligned}(1-z)^{-2} = \frac{1}{(1-z)^2} &= \cdots + \binom{r+1}{r} z^r + \cdots \\ &= 1 + 2z + \cdots + (r+1)z^r + \cdots\end{aligned} \tag{9}$$

and

$$\begin{aligned}(1-z)^{-3} = \frac{1}{(1-z)^3} &= \cdots + \binom{r+2}{r} z^r + \cdots \\ &= 1 + 3z + \cdots + \frac{(r+2)(r+1)}{2} z^r + \cdots\end{aligned} \tag{10}$$

7.4.1 Example Solve the recurrence

$$\begin{aligned}a_0 &= 1 \\ a_n &= 2a_{n-1} + 1 \qquad \textbf{if } n > 0\end{aligned} \tag{i}$$

Write (i) as

$$a_n - 2a_{n-1} = 1 \tag{ii}$$

Next, form the generating function for a_n, and a "shifted" copy of it (multiplied by $2z$; z does the shifting) underneath it (this was "inspired" by (ii)):

$$\begin{aligned}G(z) &= a_0 + a_1 z + a_2 z^2 + \cdots + a_n z^n + \cdots \\ 2zG(z) &= \quad 2a_0 z + 2a_1 z^2 + \cdots + 2a_{n-1} z^n + \cdots\end{aligned}$$

Subtract the above term-by-term to get

$$\begin{aligned}G(z)(1-2z) &= 1 + z + z^2 + z^3 + \cdots \\ &= \frac{1}{1-z}\end{aligned}$$

Hence

$$G(z) = \frac{1}{(1-2z)(1-z)} \tag{iii}$$

(iii) is $G(z)$ in closed form. To expand it back to a (known) power series we first use the "*partial fractions*" method (familiar to students of calculus) to write $G(z)$ as the sum of *two* fractions with *linear* denominators. I.e., find constants A and B such that (iv) below is true

for all z:

$$\frac{1}{(1-2z)(1-z)} = \frac{A}{(1-2z)} + \frac{B}{(1-z)}$$

or

$$1 = A(1-z) + B(1-2z) \qquad (iv)$$

Setting in turn $z \leftarrow 1$ and $z \leftarrow 1/2$ we find $B = -1$ and $A = 2$, hence

$$\begin{aligned} G(z) &= \frac{2}{1-2z} - \frac{1}{1-z} \\ &= 2\big(\cdots (2z)^n \cdots\big) - \big(\cdots z^n \cdots\big) \\ &= \cdots (2^{n+1} - 1)z^n \cdots \end{aligned}$$

Comparing this known expansion with the original power series above, we conclude that

$$a_n = 2^{n+1} - 1$$

Of course, we solved this problem much more easily in Sect. 7.2. However due to its simplicity it was worked out here again to illustrate this new method. *Normally, you apply the method of generating functions when there is no other obviously simpler way to do it.* □

7.4.2 Example Solve

$$\begin{aligned} p_1 &= 2 \\ p_n &= p_{n-1} + n \qquad \textbf{if } n > 1 \end{aligned} \qquad (i)$$

Write (i) as

$$p_n - p_{n-1} = n \qquad (ii)$$

Next, form the generating function for p_n, and a "shifted" copy of it underneath it (this was "inspired" by (ii)).

Note how this sequence starts with p_1 (rather than p_0). Correspondingly, the constant term of the generating function is p_1.

$$\begin{array}{lllllll} G(z) & = p_1 & +p_2 z & +p_3 z^2 & +\cdots & +p_{n+1} z^n & +\cdots \\ zG(z) & = & p_1 z & +p_2 z^2 & +\cdots & +p_n z^n & +\cdots \end{array}$$

Subtract the above term-by-term to get

$$\begin{aligned} G(z)(1-z) &= 2 + 2z + 3z^2 + 4z^3 + \cdots + (n+1)z^n + \cdots \\ &= 1 + \frac{1}{(1-z)^2} \qquad \text{by (9)} \end{aligned}$$

Hence

$$\begin{aligned} G(z) &= \frac{1}{1-z} + \frac{1}{(1-z)^3} \\ &= \left(\cdots z^n \cdots\right) + \left(\cdots \frac{(n+2)(n+1)}{2} z^n \cdots\right) \qquad \text{by (10)} \\ &= \cdots \left(1 + \frac{(n+2)(n+1)}{2}\right) z^n \cdots \end{aligned}$$

Comparing this known expansion with the original power series above, we conclude that

$$p_{n+1} = 1 + \frac{(n+2)(n+1)}{2}, \text{ the coefficient of } z^n$$

or

$$p_n = 1 + \frac{(n+1)n}{2}$$

□

7.4.3 Example Here is one that cannot be handled by the techniques of Sect. 7.2.

$$\begin{aligned} s_0 &= 1 \\ s_1 &= 1 \\ s_n &= 4s_{n-1} - 4s_{n-2} \quad \textbf{if } n > 1 \end{aligned} \tag{i}$$

Write (i) as

$$s_n - 4s_{n-1} + 4s_{n-2} = 0 \tag{ii}$$

to "inspire"

$$\begin{array}{lrrrrrr} G(z) & = s_0 & +s_1 z & +s_2 z^2 & +\cdots & +s_n z^n & +\cdots \\ 4zG(z) & = & 4s_0 z & +4s_1 z^2 & +\cdots & +4s_{n-1} z^n & +\cdots \\ 4z^2 G(z) & = & & 4s_0 z^2 & +\cdots & +4s_{n-2} z^n & +\cdots \end{array}$$

By (ii),

$$\begin{aligned} G(z)(1 - 4z + 4z^2) &= 1 + (1-4)z \\ &= 1 - 3z \end{aligned}$$

Since $1 - 4z + 4z^2 = (1-2z)^2$ we get

$$\begin{aligned} G(z) &= \frac{1}{(1-2z)^2} - 3z\frac{1}{(1-2z)^2} \\ &= \left(\cdots (n+1)(2z)^n \cdots\right) - 3z\left(\cdots (n+1)(2z)^n \cdots\right) \\ &= \left(\cdots \left[(n+1)2^n - 3n2^{n-1}\right] z^n \cdots\right) \end{aligned}$$

Thus,

$$\begin{aligned} s_n &= (n+1)2^n - 3n2^{n-1} \\ &= 2^{n-1}(2n + 2 - 3n) \\ &= 2^n(1 - n/2) \end{aligned}$$

□

7.4.4 Example Here is another one that cannot be handled by the techniques of Sect. 7.2.

$$\begin{aligned} s_0 &= 0 \\ s_1 &= 8 \\ s_n &= 2s_{n-1} + 3s_{n-2} \quad \textbf{if } n > 1 \end{aligned} \tag{i}$$

Write (i) as

$$s_n - 2s_{n-1} - 3s_{n-2} = 0 \tag{ii}$$

Next,

$$\begin{array}{lll} G(z) & = s_0 + s_1 z + s_2 z^2 & + \cdots + s_n z^n \quad + \cdots \\ 2zG(z) & = \quad 2s_0 z + 2s_1 z^2 & + \cdots + 2s_{n-1} z^n + \cdots \\ 3z^2 G(z) & = \qquad 3s_0 z^2 & + \cdots + 3s_{n-2} z^n + \cdots \end{array}$$

By (ii),

$$G(z)(1 - 2z - 3z^2) = 8z$$

The roots of $1 - 2z - 3z^2 = 0$ are

$$z = \frac{-2 \pm \sqrt{4+12}}{6} = \frac{-2 \pm 4}{6} = \begin{cases} -1 \\ 1/3 \end{cases}$$

hence $1 - 2z - 3z^2 = -3(z+1)(z-1/3) = (1-3z)(1+z)$, therefore

$$G(z) = \frac{8z}{(1-3z)(1+z)} = \frac{A}{1-3z} + \frac{B}{1+z} \text{ splitting into partial fractions}$$

By a calculation as in Example 7.4.1, $A = 2$ and $B = -2$, so

$$\begin{aligned} G(z) &= \frac{2}{1-3z} - \frac{2}{1+z} \\ &= 2\big(\cdots (3z)^n \cdots\big) - 2\big(\cdots (-z)^n \cdots\big) \\ &= (\cdots [2 \cdot 3^n - 2(-1)^n] z^n \cdots) \end{aligned}$$

hence $s_n = 2 \cdot 3^n - 2(-1)^n$ □

7.4.5 Example The Fibonacci recurrence.

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \quad \textbf{if } n > 1 \end{aligned} \tag{i}$$

Write (i) as

$$F_n - F_{n-1} - F_{n-2} = 0 \tag{ii}$$

Next,

$$\begin{array}{lllll} G(z) & = F_0 & +F_1 z & +F_2 z^2 & +\cdots +F_n z^n \quad +\cdots \\ zG(z) & = \quad F_0 z & +F_1 z^2 & +\cdots +F_{n-1} z^n & +\cdots \\ z^2 G(z) & = \qquad F_0 z^2 & +\cdots & +F_{n-2} z^n & +\cdots \end{array}$$

By (ii),

$$G(z)(1 - z - z^2) = z$$

The roots of $1 - z - z^2 = 0$ are

$$z = \frac{-1 \pm \sqrt{1+4}}{2} = \begin{cases} \dfrac{-1+\sqrt{5}}{2} \\ \dfrac{-1-\sqrt{5}}{2} \end{cases}$$

For convenience of notation, set

$$\phi_1 = \frac{-1+\sqrt{5}}{2}, \qquad \phi_2 = \frac{-1-\sqrt{5}}{2} \tag{iii}$$

Hence

$$\begin{aligned} 1 - z - z^2 &= -(z - \phi_1)(z - \phi_2) \\ &= -(\phi_1 - z)(\phi_2 - z) \end{aligned} \tag{iv}$$

therefore

$$G(z) = \frac{z}{1 - z - z^2} = \frac{A}{\phi_1 - z} + \frac{B}{\phi_2 - z} \text{ splitting into partial fractions}$$

from which (after some arithmetic that I will not display),

$$A = \frac{\phi_1}{\phi_1 - \phi_2}, \qquad B = \frac{\phi_2}{\phi_2 - \phi_1}$$

so

$$\begin{aligned} G(z) &= \frac{1}{\phi_1 - \phi_2}\left[\frac{\phi_1}{\phi_1 - z} - \frac{\phi_2}{\phi_2 - z}\right] \\ &= \frac{1}{\phi_1 - \phi_2}\left[\frac{1}{1 - z/\phi_1} - \frac{1}{1 - z/\phi_2}\right] \\ &= \frac{1}{\phi_1 - \phi_2}\left(\left(\cdots\left[\frac{z}{\phi_1}\right]^n \cdots\right) - \left(\cdots\left[\frac{z}{\phi_2}\right]^n \cdots\right)\right) \end{aligned}$$

therefore

$$F_n = \frac{1}{\phi_1 - \phi_2}\left(\frac{1}{\phi_1^n} - \frac{1}{\phi_2^n}\right) \tag{v}$$

Let's simplify (v):

First, by brute force calculation, or by using the "known" relations between the roots of a 2nd degree equation, we find

$$\phi_1\phi_2 = -1, \qquad \phi_1 - \phi_2 = \sqrt{5}$$

so that (v) gives

$$\begin{aligned} F_n &= \frac{1}{\sqrt{5}}\left(\frac{\phi_2^n}{(\phi_1\phi_2)^n} - \frac{\phi_1^n}{(\phi_1\phi_2)^n}\right) \\ &= \frac{1}{\sqrt{5}}\left((-1)^n\frac{\left((1+\sqrt{5})/2\right)^n}{(-1)^n} - (-1)^n\frac{\left((1-\sqrt{5})/2\right)^n}{(-1)^n}\right) \\ &= \frac{1}{\sqrt{5}}\left(\left[\frac{1+\sqrt{5}}{2}\right]^n - \left[\frac{1-\sqrt{5}}{2}\right]^n\right) \end{aligned}$$

In particular, we find that

$$F_n = O\left(\left[\frac{1+\sqrt{5}}{2}\right]^n\right)$$

since

$$\left[\frac{1-\sqrt{5}}{2}\right]^n \to 0 \text{ as } n \to \infty$$

due to $(1-\sqrt{5})/2$ being about -0.62.

That is, F_n grows exponentially with n, since $|\phi_2| > 1$.

7.5 Exercises

1. Given the recurrence below.

$$T(n) \leq \begin{cases} T(n/9) + T(63n/72) + Cn & \textbf{if } n \geq 90 \\ Cn & \textbf{if } n < 90 \end{cases}$$

Prove that $T(n) \leq 72Cn$, for $n \geq 1$.

2. *Not* using generating functions, solve in closed form employing O-notation, the following recurrence. State clearly what assumptions you need to make on T in order to have a solution that is valid for all n.

$$\begin{aligned} T(1) &= 1 \\ T(n) &= 2T(n/2) + n^2 \end{aligned}$$

3. Solve in closed form the following recurrence, and express the answer in Big-O notation. Do not use generating functions.

$$T(1) = a$$
$$T(n) = 3T(n/2) + n$$

4. In this exercise you are asked to use the method of *generating functions* —the telescoping method is not acceptable.
Solve in closed form the following recurrence.

$$a_0 = 0$$
$$a_n = a_{n-1} + 1, \text{ for } n \geq 1$$

5. Design a divide-and-conquer recursive function procedure $F(n)$ (give the pseudo-code in pseudo-C) which returns the n-th Fibonacci number, F_n. Ensure that it runs in $O(\log n)$ arithmetic operations (multiplications/additions).
In particular, (a) prove the correctness of your algorithm, *and* (b) prove that indeed it runs in "time" $O(\log n)$, by setting and solving the appropriate recurrence relation that defines the run-time.
Hint. It is useful to approach the problem by *proving* first that

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n-1} \\ F_{n-2} \end{pmatrix}$$

and then conclude that

$$\begin{pmatrix} F_n \\ F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^{n-1} \begin{pmatrix} F_1 \\ F_0 \end{pmatrix}$$

6. The *Euclidean algorithm* towards finding the *greatest common divisor* (gcd) of two natural numbers $a > b > 0$ —denoted as $gcd(a, b)$— notes that $gcd(a, b) = gcd(b, r)$, where $a = bq + r$ with $0 \leq r < b$. Argue that the process of shifting the answer — finding $gcd(a, b)$, that is— from the pair (a, b) to the pair (b, r) is terminating.
Estimate the number of *steps* of that process. Then gauge an *upper bound* of this roughly implied algorithm in big-O notation in terms of the digit-length of a.
Hint. Relate this problem to the generation of the Fibonacci sequence.

7. Using generating functions solve the following recurrence exactly in closed form

$$a_0 = 1$$
$$a_1 = 2$$
$$\text{for } n \geq 2, \quad a_n = 2a_{n-1} - a_{n-2}$$

8. a. Prove that if $G(z)$ is the generating function for the sequence (a_n), for $n = 0, 1, \ldots$, that is,

$$G(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + \cdots$$

then $G(z)/(1 - z)$ is the generating function of the sequence $(\sum_{i=0}^{n} a_n)$, for $n = 0, 1, \ldots$.

b. Now using generating functions prove that

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

9. Using generating functions solve the following recurrence in closed form.

$$\begin{aligned} a_0 &= 0 \\ a_1 &= 1 \\ a_2 &= 2 \\ \text{for } n \geq 3,\ a_n &= 3a_{n-1} - 3a_{n-2} + a_{n-3} \end{aligned}$$

10. Using generating functions solve the following recurrence in closed form.

$$\begin{aligned} a_0 &= 0 \\ a_1 &= 1 \\ \text{for } n \geq 2,\ a_n &= -2a_{n-1} - a_{n-2} \end{aligned}$$

11. Consider the recurrence

$$\begin{aligned} a_0 &= 1 \\ a_1 &= 1 \\ a_2 &= 1 \\ \text{for } n \geq 3,\ a_n &= a_{n-1} + a_{n-3} \end{aligned}$$

By induction or in any other manner, prove that $a_n \geq 2a_{n-2}$, for $n \geq 3$ and $a_n \geq 2^{(n-2)/2}$, for $n \geq 2$.

12. This is a tooling exercise for Exercises 13, 14 below and Exercise 8.5.3 in the next chapter. Prove these facts about floors and ceilings.

- $\lceil n/2 \rceil + \lfloor n/2 \rfloor = n$, for all n. *Hint*. Argue the two cases separately, n even and n odd.
- $\lceil n/2 \rceil - 1 = \lfloor (n-1)/2 \rfloor$, for all n. *Hint*. Argue the two cases separately, n even and n odd.
- $\lceil n/2 \rceil = \lfloor (n+1)/2 \rfloor$, for all n. *Hint*. Argue the two cases separately, n even and n odd.
- $\lceil n/2 \rceil \geq n/2 \geq \lfloor n/2 \rfloor$ (trivial).
- $\lfloor n/2 \rfloor + 1 \geq \lceil n/2 \rceil$. *Hint*. Directly from the definitions of $\lfloor \ldots \rfloor$ and $\lceil \ldots \rceil$.
- $\lfloor n/4 \rfloor = \lfloor \lfloor n/2 \rfloor /2 \rfloor$. *Hint*. If $l = \lfloor n/4 \rfloor$, then by definition, $l \leq n/4 < l+1$ hence $2l \leq n/2 < 2l+2$ thus $2l \leq \lfloor n/2 \rfloor < 2l+2$ (explain "$\leq$" but the "$<$" is trivial). It follows that $l \leq \lfloor n/2 \rfloor /2 < l+1$. Checkmate in one very *short* sentence.

- $\lceil n/4 \rceil = \lceil \lceil n/2 \rceil /2 \rceil$. *Hint*. If $l = \lceil n/4 \rceil$, then by definition, $l - 1 < n/4 \leq l$ hence $2l - 2 < n/2 \leq 2l$ thus $2l - 2 < \lceil n/2 \rceil \leq 2l$ (explain "$\leq$" but the "$<$" is trivial). It follows that $l - 1 < \lceil n/2 \rceil /2 \leq l$. Checkmate in one very *short* sentence.

13. Consider standard binary search, where an array $A[1 \ldots n]$ with entries in ascending order is recursively searched for the possible occurrence of K as an $A[i]$ as follows:

a. Check if K matches $A[\lfloor (n+1)/2 \rfloor]$. If yes, *exit* successfully. If not
b. Recursively call the search algorithm to search for K in the array

$$A[1 \ldots \lfloor (n+1)/2 \rfloor - 1]$$

if $K < A[\lfloor (n+1)/2 \rfloor]$
c. Recursively call the search algorithm to search for K in the array

$$A[\lfloor (n+1)/2 \rfloor + 1 \ldots n]$$

if $K > A[\lfloor (n+1)/2 \rfloor]$.

a. Set up the run time (upper bound, worst case) for the run time $\lambda n.T(n)$ —assumed to be a *non-decreasing* function of n— of the algorithm. Preserve $\lfloor \ldots \rfloor$ in the recurrence equations for T, i.e., do *not* approximate with non-integer expressions such as $n/2$, $(n+1)/2$.
Hint. Trivially, the worst case $T(n)$ is 1 (comparing K with the middle entry) plus *maximum* of worst case time for *left* —$T(\lfloor (n+1)/2 \rfloor - 1)$— call and *right* —$T(n - \lfloor (n+1)/2 \rfloor)$— call. Decide which of these two calls is always worst and arrive at $T(n) = 1 + T(\ldots)$. Note the "="! You are equating two sides with the worst case run time in each.
b. Now solve for $T(n)$ in exact closed form by the telescoping trick and using the appropriate tools from 12. *Hint*. Extend the last bullet from 12 with the result $\lfloor \lfloor n/2^m \rfloor /2 \rfloor = \lfloor n/2^{m+1} \rfloor$.

14. Consider a modified "binary search" where the "middle" entry of the array that we compare with the *key*[5] we are searching for, is the one stored in location to $\lfloor n/2 \rfloor$ rather than $\lfloor (n+1)/2 \rfloor$.

a. Formulate the recurrence that expresses the worst case number of comparisons $T(n)$ in this modified binary search.

[5] In storing data —for example in an array— we often store them accompanied by short aliases that we call "keys". Thus instead of repeatedly comparing a possibly unwieldy and large record during our search, we instead compare repeatedly its key against the keys of the stored (in the array) records.

b. Solve your recurrence *exactly*, that is, do not "simplify" the floors, and do not answer in O-notation.

15. Solve the following recurrence and express the solution in O-notation, where it is given that the function T that expresses the run time is increasing.

$$T(n) = \begin{cases} 0 & \textbf{if } n \leq 2 \\ \sqrt{n}\, T(\sqrt{n}) + n & \textbf{if } n > 2 \end{cases}$$

Hint. Solve for $f(n) = T(n)/n$ instead. The simplified equations for f must be solved first for the special case of $n = 2^{2^m}$. Then adapt to any n.
Caution! Show that your solution is valid for all n rather than only for n of restricted forms.

16. Solve the following recurrence in O-notation, where we are told that $T(n)$ is increasing.

$$T(n) = \begin{cases} 1 & \textbf{if } n = 1 \\ aT(n/b) + n^2 & \textbf{if } n > 2 \end{cases}$$

Caution!

a. You need to consider cases according to the values of the natural numbers a kai b.
b. Show that your closed-form solution is valid for *all* n rather than only for those of some restricted forms.

17. Consider the described below algorithm to search a sorted (ascending order) array of length n:

/ * We are searching the array for an element equal to m * /
if $n \leq 7$ **then do** a linear search of $A[1 \ldots n]$
else if $A[7] = m$ **then** successful **exit**
else if $A[7] > m$ **then do** a linear search of $A[1 \ldots 7]$
else call recursively on segment $A[8 \ldots n]$

Formulate and solve *exactly* (not in O-notation) the recurrence relation that defines the worst case number of comparisons $T(n)$.

18. Let the symbol $\Pi(n)$ stand for the number of ways we can write down the sum $a_1 + \cdots + a_n$ with all possible brackets inserted.

Examples:

The trivial "sum" a_1 offers only one way to become "fully parenthesised", namely, (a_1).

Then, $a_1 + a_2 + a_3$ allows two ways to be fully parenthesised, namely, $((a_1) + ((a_2) + (a_3)))$ and $(((a_1) + (a_2)) + (a_3))$.

In terms of Π we have $\Pi(1) = 1$, $\Pi(3) = 2$.

a. Find the correct recurrence that expresses $\Pi(n)$.
b. Find the generating function $G(z)$ of the sequence $\Pi(n)$, $n = 1, 2, \ldots$ *in closed form*, but you are not required to find $\Pi(n)$ itself in closed form.

An Addendum to Trees

8

Overview

This short Addendum uses trees to calculate a sum that arrises in the analysis of algorithms in *exact* (i.e., not in O-notation) closed form. The difficulty with this sum is that its terms involve the ceiling function —in something forbidding like $\lambda x.\lceil \log_2 x \rceil$. In the area of discrete mathematics known as *graph theory*, trees —in particular binary trees— play a central role as special cases of the so-called *directed graphs*. While trees are studied for their own merit in modelling important data structures in computing practise, they have also unexpected applications to discrete mathematics such as the one we will demonstrate in this chapter. The chapter concludes with an application of *generating functions* used to compute a simple expression that computes *the number of all extended trees* that have n internal nodes.

There is a direct graph-theoretic definition of a tree —that is beyond the design of this volume— it is arguably more convenient to go the direct graph-independent route to a definition (Example 6.3.10) that we took in Chapter 6 if for no other reason besides the fact that such definition enables us to prove tree properties by *structural induction*. Our definitive definition has been given in Example 6.3.10.

8.1 Trees: More Terminology

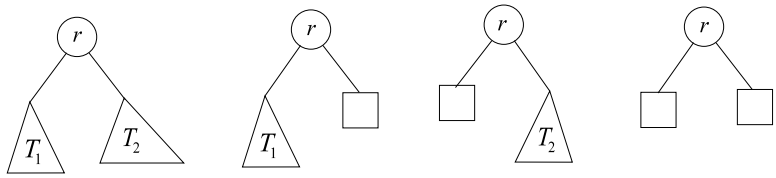
8.1.1 Example This example supplements the discussion started at 6.3.10.

So here are some trees $\emptyset$, $(\emptyset, 1, \emptyset)$, and $((\emptyset, 1, \emptyset), 2, \emptyset)$ where we wrote 1 and 2 for $\bigcirc_1$ and $\bigcirc_2$ respectively.

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0_8

In the figure below the notation shows only the structure part (not the support) and the first example is what we may call "the most general tree" as there is no specific assumptions regarding the left and right *subtrees* of the root $\bigcirc_r$ (which we can, by abuse of notation and language just call "root r"). They are "any" trees drawn as triangles with names "T_1" and "T_2" respectively.

The last tree below has both subtrees of its root empty, while the second tree has an empty right subtree. Thus, the "general tree" is drawn as one of the following, where r is the root.



The leftmost drawing uses the notation of a "triangle" to denote a tree. The other three cases are used when we want to draw attention to the fact that the right (respectively, left, both) subtree(s) is (are) empty. □

8.1.2 Definition (**Simple and Extended Trees**) We agree that we have *two types* of tree-*notations* (abusing language we say that we have two types of *trees*).
Simple Trees are those drawn *only* with "round" nodes (i.e., we do not draw the empty subtrees).
Extended Trees are those that all empty subtrees are drawn as "square nodes" (as in 6.3.10). We call, in this case, the round nodes *internal* and the square nodes *external*. □

Clearly, the "external nodes" of an extended tree cannot hold any information since they are (notations for) empty subtrees.

Alternatively, we may think of them (in a programming-implementation sense) as *notations* for *null pointers*. That is, in the case of *simple trees* we do *not* draw any null links, while in the case of *extended trees* we draw *all* null links as square nodes.

8.1.3 Definition (**Graph-Theoretic Terminology**) We introduce some standard *graph theory terminology*:

If node a points to node b by an edge, then b is *a child* of a and a is *the parent* of b. If two nodes are the children of the same node, then they are *siblings*.

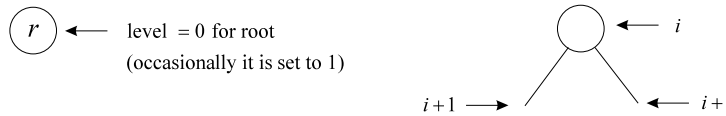
A sequence of nodes $a_1, a_2, \ldots, a_n$ in a tree is a *path* or *chain* iff for *all* $i = 1, 2, \ldots, n-1$, a_i is the parent of a_{i+1}. We say that this is a chain *from* a_1 *to* a_n. We say that a_n is a *descendant* of a_1 and that a_1 is an *ancestor* of a_n.

A node is a *leaf* iff it has no children. □

In an extended tree the only leaves are the external (square) nodes.

8.1.4 Definition We define *levels* of nodes in a tree recursively (inductively):
The root has level 0 (sometimes we assign level 1 to the root, as it may prove convenient).
If b is any child of a and a has level i, then b has level $i + 1$.
The *highest* level in a tree is called the *height* of the tree. □

8.1.5 Example (Assignment of levels)

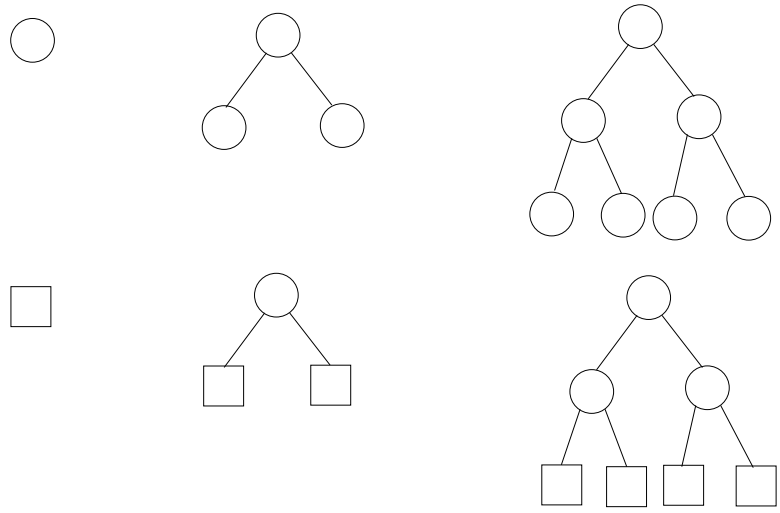


□

8.1.6 Definition A *non leaf node* is *fertile* iff it has two children. A *tree is fertile* iff *all* its non leaves are fertile.
A tree is *full* iff it is fertile *and all* the leaves are at the same level. □


An extended tree is always fertile. The last sentence above then simplifies, if restricted to such trees, to "All square nodes are at the same level".


8.1.7 Example (Full Trees). A "full tree" has all the possible nodes it deserves.

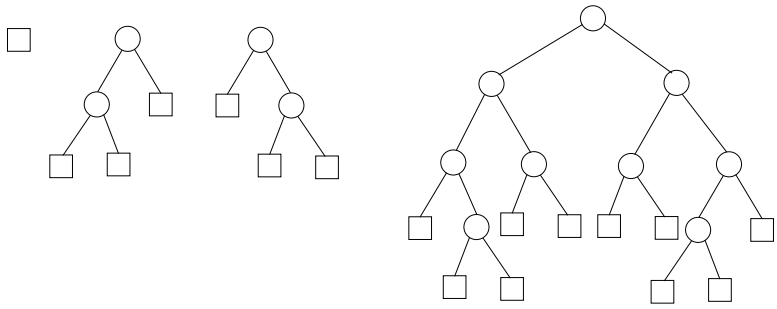


□

8.1.8 Definition A tree is *complete* iff it is fertile *and* all the leaves occupy *at most* two consecutive levels (obviously, one is going to be the last (highest) level). □

Again, for extended trees we need only ask that all square nodes occupy "at most two consecutive levels".

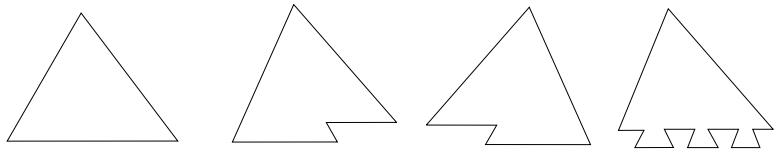
8.1.9 Example (Complete Trees)



Redraw the above so that all the square nodes are "rounded" and you get examples of complete *Simple Trees*.

□

8.1.10 Example There is a variety of complete trees, the general case having the nodes in the highest level scattered about in any manner. In practice we like to deal mostly with complete trees whose highest level nodes are left justified (left-complete) or right-justified (right-complete). See the following, where we drew (abstractly) a full tree (special case of complete!), a left complete, a right complete, and a "general" complete tree.



Example 8.1.9 provides a number of more concrete examples □

8.2 A Few Provable Facts About Trees

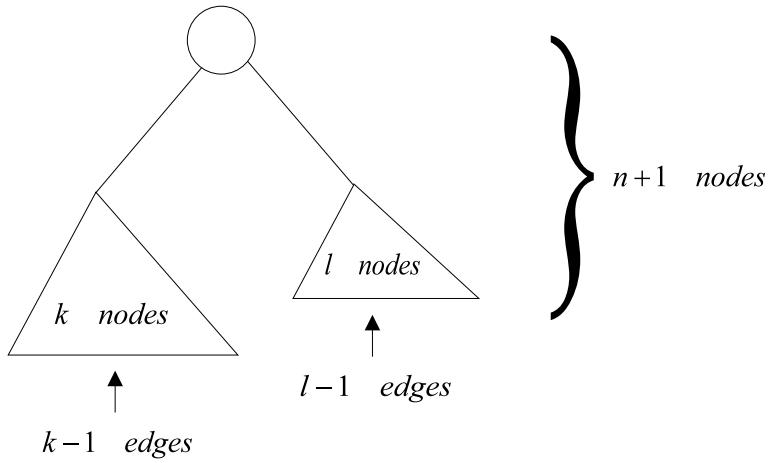
8.2.1 Theorem *An* extended *tree with a total number of nodes equal to n (this accounts for internal* and *external nodes) has* $n - 1$ *edges.*

Proof We do induction with respect to the definition of trees, or as we say in short, *induction on trees*. Cf. 6.2.3.

Basis. The smallest tree is $\emptyset$, i.e., exactly one "square" node. It has no edges, so the theorem verifies in this case.

I.H. Assume the claim for "small" trees.

I.S. Consider the "big" tree below composed of two "small" trees of k and l nodes and a root. Say the total number of nodes is $n+1$.[1]



By I.H., the left and right (small) subtrees have $k-1$ and $l-1$ edges respectively. Thus the total number of edges is $k-1+l-1+2=k+l$ (Note that in an extended tree all round nodes are fertile, so both edges emanating from the root *are* indeed present).

On the other hand, the total number of nodes $n+1$ is $k+l+1$. We rest our case. □

8.2.2 Corollary *An extended tree of* n internal *nodes has* $n+1$ external *nodes.*

Proof This was proved in 6.3.13. Here is another proof.

Let us have ϕ internal and ε external nodes. Given that $\phi = n$.

By the 8.2.1 the tree has $\varepsilon+\phi-1$ edges. That is, accounting differently, 2ϕ edges since all round nodes are fertile, and the square nodes are all leaves. Thus,

$$\varepsilon+\phi-1=2\phi$$

from which, $\phi=\varepsilon-1$. Thus there are $n+1$ square nodes as claimed. □

8.2.3 Corollary *A* simple *tree of* $n\geq 1$ *nodes has* $n-1$ *edges.*

Proof Let E be the original number of edges, still to be computed in terms of n. Add external nodes (two for each "original" leaf). What this does is:

[1] No magic with $n+1$. We could have called the total n, but then we would have to add "where $n\geq 1$" to account for the presence of the root. The "≥ 1" part is built-in if you use $n+1$ instead.

It adds $n+1$ square nodes, by the previous corollary.
It adds $n+1$ new edges (one per square node). Thus,

Total Nodes$= 2n+1$
Total Edges$= E+n+1$

By theorem 8.2.1, $E+n+1=2n$, hence $E=n-1$ as claimed. □

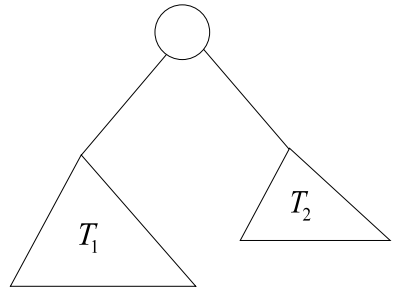
8.2.4 Theorem *In any nonempty* fertile simple *tree we have*

$$\sum_{\substack{l \text{ is a leaf's} \\ \text{level}}} 2^{-l} = 1$$

where we assigned level 0 *to the root.*

Proof Induction on trees.

Basis The smallest tree is one round node. Its level is 0 and $2^{-0}=1$, so we are OK.
I.H. Assume for small trees, and go to the "big" case.
I.S. The big case (recall that the tree is fertile, so even though simple, the root has two children).



Since each of T_1 and T_2 are "small", I.H. applies to give

$$\sum_{\substack{l \text{ is a leaf's} \\ \text{level in } T_1}} 2^{-l} = 1 \tag{1}$$

and

$$\sum_{\substack{l \text{ is a leaf's} \\ \text{level in } T_2}} 2^{-l} = 1 \tag{2}$$

It is understood that (1) and (2) are valid for T_1 and T_2 "free-standing" (i.e., root level is 0 in each). When they are incorporated in the overall tree, call it T, then their roots obtain a level value of 1, so that formulas (1) and (2) need adjustment: All levels now in T_1, T_2 are by one larger than the previous values. Thus,

$$\sum_{\substack{l \text{ is a leaf's} \\ \text{level in } T}} 2^{-l} = \sum_{\substack{l \text{ is a leaf's} \\ \text{level in } T_1 \\ \text{free standing}}} 2^{-(l+1)} + \sum_{\substack{l \text{ is a leaf's} \\ \text{level in } T_2 \\ \text{free standing}}} 2^{-(l+1)} \overset{\text{I.H.}}{=} 1/2 + 1/2 = 1$$

□

8.2.5 Corollary *In an* extended *tree*

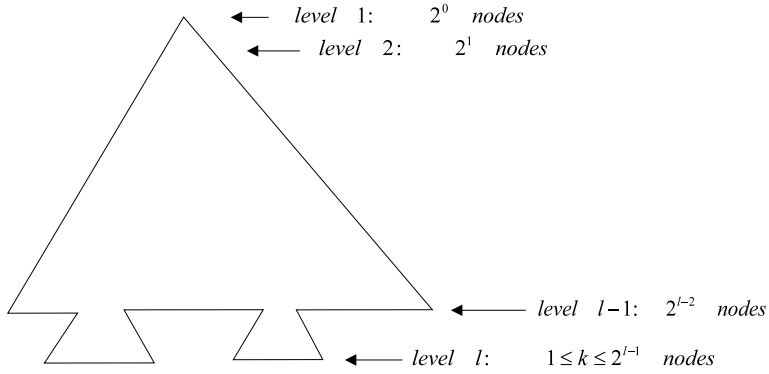
$$\sum_{\substack{l \text{ is a leaf's} \\ \text{level}}} 2^{-l} = 1$$

where we assigned level 0 *to the root.*

8.2.6 Corollary *In both 8.2.4 and 8.2.5, if the root is assigned level 1, then*

$$\sum_{\substack{l \text{ is a leaf's} \\ \text{level}}} 2^{-l} = 1/2$$

Next we address the relation between n, the number of nodes in a simple *complete* tree (8.1.8), with its height l (8.1.4).



Clearly,

$$n = 2^0 + 2^1 + \cdots + 2^{l-2} + k \le 2^l - 1 \qquad (A)$$

thus

$$2^{l-1} - 1 < n \leq 2^l - 1$$

From this follows

$$2^{l-1} < n + 1 \leq 2^l$$

or

$$2^{l-1} \leq n < 2^l$$

leading to

$$\boxed{l = \lceil \log_2(n+1) \rceil = \lfloor \log_2 n \rfloor + 1} \qquad (*)$$

a good formula to remember.

Of course, all this holds when counting levels from 1 up. Check to see what happens if the root level is 0.

How does k, the number of nodes at level l, relate to n, the number of nodes in the tree? From (A),

$$\boxed{k = n + 1 - 2^{l-1}} \qquad (**)$$

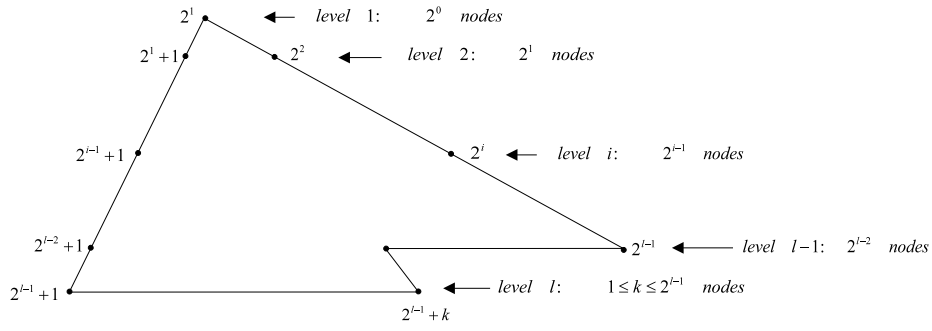
another very important formula this to remember, which can be also written (because of $(*)$) as

$$\begin{aligned} k &= n + 1 - 2^{\lceil \log_2(n+1) \rceil - 1} \\ &= n + 1 - 2^{\lfloor \log_2 n \rfloor} \end{aligned} \qquad (B)$$

Note that $(**)$ or (B) hold even if some or all nodes at level $l-1$ have no more than one child (in which case the tree fails to be complete, or fertile for that matter).

8.3 An Application to Summations

Let us see next what happens if we label the nodes of a left-complete tree by numbers successively, starting with label 2 for the root.



An easy induction shows that at level i we have the labels

$$2^{i-1}+1, 2^{i-1}+2, \ldots, 2^i \tag{1}$$

Note that if t is *any* of the numbers in (1), then $2^{i-1} < t \leq 2^i$, hence

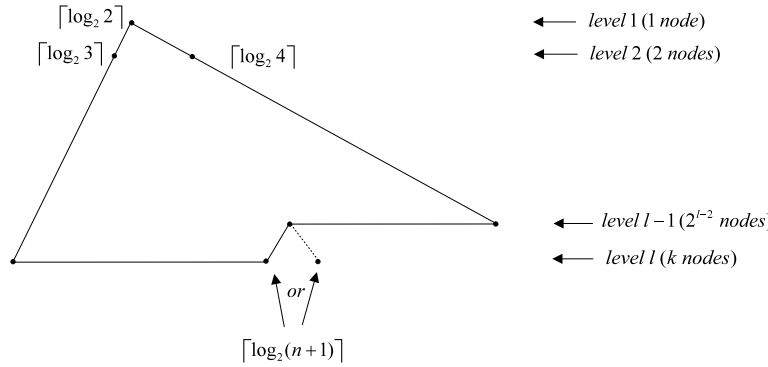
$$\lceil \log_2 t \rceil = i$$

In words, the ceiling of $\log_2$ of *any* node-label at level i equals i.

We are in a position now to evaluate the sum

$$A = \sum_{i=2}^{n+1} \lceil \log_2 i \rceil \tag{2}$$

which arises in the analysis of certain algorithms. The figure below helps to group terms appropriately:



Clearly,

$$A = \sum_{i=1}^{l-1} i2^{i-1} + k\lceil \log_2 (n+1) \rceil \tag{3}$$

To compute (3) we need to find k as a function of n, and to evaluate

$$B = \sum_{i=1}^{l-1} i2^{i-1} \tag{4}$$

There are two cases at level l as in the previous figure. Regardless, k is given in $(**)$ of the previous section (p. 238) as

$$k = n + 1 - 2^{l-1}$$

Thus, we only have to compute (4). Now,

$$\begin{aligned}
B &= \sum_{i=1}^{l-1} i2^{i-1} \\
&= \sum_{i=0}^{l-2} (i+1)2^i \\
&= \sum_{i=0}^{l-2} i2^i + 2^{l-1} - 1 \\
&= 2\sum_{i=1}^{l-2} i2^{i-1} + 2^{l-1} - 1 \\
&= 2\sum_{i=1}^{l-1} i2^{i-1} - (l-2)2^{l-1} - 1 \\
&= 2B - (l-2)2^{l-1} - 1
\end{aligned}$$

Thus, $B = (l-2)2^{l-1} + 1$, and the original becomes (recall $(*)$ and $(**)$!)

$$\begin{aligned}
A &= (l-2)2^{l-1} + 1 + kl \\
&= l2^{l-1} - 2^l + 1 + (n+1-2^{l-1})l \\
&= (n+1)l - 2^l + 1 \\
&= (n+1)\lceil \log_2(n+1) \rceil - 2^{\lceil \log_2(n+1) \rceil} + 1
\end{aligned}$$

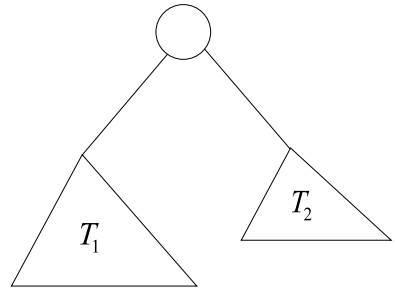
Note. A rough analysis of A would go like this: Each term of the sum is $O\big(\log(n+1)\big)$ and we have n terms. Therefore, $A = O\big(n \log(n+1)\big)$. However we often need the exact answer ...

8.4 How Many Trees?

We want to find the number of *all* extended binary trees of n internal nodes.

Let the sought quantity be called x_n.

Refer to the following figure, where the tree has n internal nodes, while subtree T_1 has m internal nodes and subtree T_2 has r internal nodes.



Thus, $n = m + r + 1$. We can choose T_1 in x_m different ways, and for each way, we can have x_r different versions of T_2. And that is true for each size of T_1. Thus, the recurrence equations for x_n are

$$x_0 = 1(\text{there is only one, empty tree}) \tag{1}$$

$$x_n = \sum_{n-1=m+r} x_m x_r, \text{ for } n > 0 \tag{2}$$

We recognise in (2) the so-called *convolution* resulting from $G(z)^2$, where

$$G(z) = x_0 + x_1 z + x_2 z^2 + \cdots + x_n z^n + \cdots \tag{3}$$

Indeed,

$$\begin{aligned} G(z)^2 &= x_0^2 + (x_0 x_1 + x_1 x_0)z + \cdots + (\textstyle\sum_{n-1=m+r} x_m x_r) z^{n-1} + \cdots \\ &= x_1 + x_2 z + \cdots + x_n z^{n-1} + \cdots \end{aligned} \tag{4}$$

Thus, $zG(z)^2 + x_0 = G(z)$, or

$$zG(z)^2 - G(z) + 1 = 0 \tag{5}$$

We solve (5) for $G(z)$ to find

$$G(z) = \begin{cases} \dfrac{1 + \sqrt{1 - 4z}}{2z} \\ \dfrac{1 - \sqrt{1 - 4z}}{2z} \end{cases} \text{ or} \tag{6}$$

equivalently,

$$zG(z) = \begin{cases} \dfrac{1 + \sqrt{1 - 4z}}{2} \\ \dfrac{1 - \sqrt{1 - 4z}}{2} \end{cases} \text{, or} \tag{6'}$$

The first of (6′) is false for $z = 0$, so we keep and develop the second solution. To this end we expand $\sqrt{1 - 4z} = (1 - 4z)^{1/2}$ by the binomial expansion.

$$(1 - 4z)^{1/2} = 1 + \cdots + \binom{1/2}{n} (-4z)^n + \cdots$$

Let us work with the coefficient $\binom{1/2}{n} (-4)^n$.

$$\begin{aligned}
\binom{1/2}{n}(-4)^n &= \frac{1/2(1/2-1)(1/2-2)\dots(1/2-[n-1])}{n!}(-4)^n \\
&= (-1)^n 2^{2n}\frac{1(1-2\cdot 1)(1-2\cdot 2)\dots(1-2\cdot[n-1])}{2^n n!} \\
&= (-1)^n(-1)^{n-1}2^n\frac{(2\cdot 1-1)(2\cdot 2-1)\dots(2\cdot[n-1]-1)}{n!} \qquad (7)\\
&= -2\frac{(2\cdot 1-1)[2\cdot 1](2\cdot 2-1)[2\cdot 2]\dots(2\cdot[n-1]-1)[2\cdot[n-1]]}{n!(n-1)!} \qquad (8)\\
&= -2\frac{(2n-2)!}{n!(n-1)!} \\
&= -\frac{2}{n}\binom{2n-2}{n-1}
\end{aligned}$$

Going from (7) to (8) above we introduced factors $[2\cdot 1], [2\cdot 2], \dots, [2\cdot[n-1]]$ in order to "close the gaps" and make the numerator be a factorial. This has spent $n-1$ of the n 2-factors in 2^n, but introduced $(n-1)!$ on the numerator, hence we balanced it out in the denominator.

It follows that, according to the second case of (6),

$$x_n = G(z)[z^n] = \frac{1-\left(1-2z-\dots-\frac{2}{n}\binom{2n-2}{n-1}z^n-\dots\right)}{2z}[z^n] \qquad (9)$$

where for any generating function $G(z)$, $G(z)[z^n]$ denotes the coefficient of z^n.

In short,

$$x_n = \frac{1}{n+1}\binom{2n}{n}$$

Don't forget that we want $G(z)[z^n]$; we have adjusted for the division by z.

8.5 Exercises

1. Prove the identity (due to Vandermonde)

$$\binom{n+m}{r} = \sum_{i+j=r}\binom{n}{i}\binom{m}{j}$$

Hint. Multiply the generating functions $(1+z)^n$ and $(1+z)^m$ and look at the r-th coefficient.

2. Do n, m above have to be natural numbers?

3. This exercise is asking you to 1) formulate the recurrence equations and 2) solve them in exact closed form (not in O-notation) for the worst case run time of the "*linear merge sort*" algorithm.
Just like the binary search algorithm, linear merge sort uses a *divide and conquer* — as practitioners in the *analysis of algorithms* call them— technique to devise a fast algorithm.
In this case we want to sort an array $A[1 \ldots n]$ in ascending order. Thus we divide the problem into two almost equal size problems —to sort each of $A[1 \ldots \lfloor (n+1)/2 \rfloor]$ and $A[\lfloor (n+1)/2 \rfloor + 1 \ldots n]$— and we do so by calling the procedure recursively for each half-size array.
We then do a linear —essentially n-comparisons— merge of the two sorted half-size arrays.
Hints and directions.

a. Prove that the recurrence equations for the worst case are

$$T(0) = 0$$
$$T(1) = 0$$

Case of $n > 1$

$$T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n$$

Be sure to prove that the equations above are correct.

b. You will use now the tools from Exercise 7.5.12 and also what we have learnt from our work (your work) on recurrence equations solving. Exercise 7.5.13 will be helpful in the final stages of your solution of the present exercise. Indeed let me show how you can solve the above recurrence by reducing it, essentially, to the binary search case recurrence.

Step 1: Towards the telescoping trick. We are also trying to get all divisions by 2 in $\lceil \ldots \rceil$-notation. So,

$$\begin{aligned} T(n) - T(n-1) &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) - \\ &\quad - T(\lceil (n-1)/2 \rceil) - T(\lfloor (n-1)/2 \rfloor) + 1 \\ &= T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) - \\ &\quad - T(\lfloor n/2 \rfloor) - T(\lfloor (n-1)/2 \rfloor) + 1 \\ &= T(\lceil n/2 \rceil) - T(\lceil n/2 \rceil - 1) + 1 \end{aligned}$$

Step 2: *Rename*: $B(n) \stackrel{Def}{=} T(n) - T(n-1)$ so that $B(n) = B(\lceil n/2 \rceil) + 1$ and $B(n) = 0$, for $n \in \{0, 1\}$.

Step 3: Solve for $B(n)$ similarly to Exrecise 7.5.13 and note that $1 < n/2^m \leq 2$ iff $2^m < n \leq 2^{m+1}$ iff $m < \log_2 n \leq m+1$ iff $\lceil \log_2 n \rceil = m+1$. Thus,

$$\begin{aligned} B(n) &= B(\lceil n/2 \rceil) + 1 \\ &= B(\lceil n/2^2 \rceil) + 2 \\ &= B(\lceil n/2^3 \rceil) + 3 \\ &\vdots \\ &= B(\lceil n/2^{m+1} \rceil) + m + 1 \end{aligned}$$

So (*use the initial (boundary) conditions for B and provide details!*) conclude that $B(n) = \lceil \log n \rceil$, where I wrote "log" for "$\log_2$", and

Step 4: Compute the sum of the telescoping $T(n) - T(n-1) = \lceil \log n \rceil$ using Section 8.3.

4. Given an extended tree. Traverse it and mark its round nodes in the order you encounter them.
Program —as a pseudo program— this traversal so that you first process the left subtree, then the root and then the right subtree. This is called the *inorder* traversal.
Suppose now that the given tree contains one number in each round node. Moreover assume the tree has the property that for every round node the number contained therein is greater than all numbers contained in *the node's* left subtree and less than all the numbers in the right subtree *of the node*.
Prove that the inorder traversal of such a tree will visit all these numbers in *ascending order*.

5. We have introduced edges to trees. Prove that no node (round or square) in a tree has more than one edge coming into it (from above).

References

J. Barwise, *Admissible Sets and Structures* (Springer, New York, 1975)

É. Borel, *Leçons sur la théorie des fonctions*, 3rd edn. (Gauthier-Villars, Paris, 1928)

N. Bourbaki, *Éléments de Mathématique; Théorie des Ensembles* (Hermann, Paris, 1966)

M. Davis, *The Undecidable* (Raven Press, Hewlett, NY, 1965)

R. Dedekind, *Was sind und was sollen die Zahlen? Vieweg, Braunschweig, 1888* (In English translation by W.W. Beman; cf, Dedekind, 1963)

R. Dedekind, *Essays on the Theory of Numbers* (First English edition translated by W.W. Beman and published by Open Court Publishing, 1901) (Dover Publications, New York, 1963)

J.A. Dieudonné, *Foundations of Modern Analysis* (Academic Press, New York, 1960)

R. Graham, D. Knuth, O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2 edn. (Addison-Wesley, 1994)

David Gries, Fred B. Schneider, *A Logical Approach to Discrete Math* (Springer, New York, 1994)

T.J. Jech, *Set Theory* (Academic Press, New York, 1978)

S.C. Kleene, Recursive predicates and quantifiers. Trans. Am. Math. Soc. **53**, 41–73 (1943) (Also in Davis (1965), 255–287)

D.E. Knuth, *The Art of Computer Programming; Fundamental Algorithms*, vol. 1, 2 edn. (Addison-Wesley, 1973)

Kenneth Kunen, Set Theory, *An Introduction to Independence Proofs* (North-Holland, Amsterdam, 1980)

A.G. Kurosh, *Lectures on General Algebra* (Chelsea Publishing Company, New York, 1963)

A. Levy, *Basic Set Theory* (Springer, New York, 1979)

R. Montague, Well-founded relations; generalizations of principles of induction and recursion. Bull. Am. Math. Soc. 61, 442 (1955) (abstract)

Y.N. Moschovakis, Abstract first-order computability. Trans. Am. Math. Soc. **138**(427–464), 465–504 (1969)

H. Rogers, *Theory of Recursive Functions and Effective Computability* (McGraw-Hill, New York, 1967)

J.R. Shoenfield, *Mathematical Logic* (Addison-Wesley, Reading, MA, 1967)

A.L. Tarski, General principles of induction and recursion; the notion of rank in axiomatic set theory and some of its applications. Bull. Am. Math. Soc. 61, 442–443 (1955) (2 abstracts)

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0

G. Tourlakis, *Lectures in Logic and Set Theory, Volume 1: Mathematical Logic* (Cambridge University Press, Cambridge, 2003a)
G. Tourlakis, *Lectures in Logic and Set Theory, Volume 2: Set Theory* (Cambridge University Press, Cambridge, 2003b)
G. Tourlakis, *Mathematical Logic* (Wiley, Hoboken, NJ, 2008)
G. Tourlakis, *Theory of Computation* (Wiley, Hoboken, NJ, 2012)
G. Tourlakis, *Computability* (Springer Nature, New York, 2022)
R.L. Wilder, *Introduction to the Foundations of Mathematics* (Wiley, New York, 1963)

Index

G. Tourlakis, *Discrete Mathematics*, Synthesis Lectures on Mathematics & Statistics,
https://doi.org/10.1007/978-3-031-30488-0