Discrete Mathematics for Computer Science, Some Notes

Book · June 2008

DOI: 10.1007/978-1-4419-8047-2 · Source: arXiv

Jean Gallier
University of Pennsylvania

Discrete Mathematics, Some Notes

Jean H. Gallier
University of Pennsylvania, jean@cis.upenn.edu

Discrete Mathematics
Some Notes

Jean Gallier
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
e-mail: `jean@cis.upenn.edu`

January 26, 2009

Discrete Mathematics
Some Notes

Jean Gallier

Abstract: These are notes on discrete mathematics for computer scientists. The presentation is somewhat unconventional. Indeed I begin with a discussion of the basic rules of mathematical reasoning and of the notion of proof formalized in a natural deduction system "a la Prawitz". The rest of the material is more or less traditional but I emphasize partial functions more than usual (after all, programs may not terminate for all input) and I provide a fairly complete account of the basic concepts of graph theory.

Preface

The curriculum of most undergraduate programs in computer science includes a course untitled *Discrete Mathematics*. These days, given that many students who graduate with a degree in computer science end up with jobs where mathematical skills seem basically of no use,[1] one may ask why these students should take such a course. And if they do, what are the most basic notions that they should learn?

As to the first question, I strongly believe that *all* computer science students should take such a course and I will try justifying this assertion below.

The main reason is that, based on my experience of more than twenty five years of teaching, I have found that the majority of the students find it very difficult to present an argument in a rigorous fashion. The notion of a proof is something very fuzzy for most students and even the need for the rigorous justification of a claim is not so clear to most of them. Yet, they will all write complex computer programs and it seems rather crucial that they should understand the basic issues of program correctness. It also seems rather crucial that they should possess some basic mathematical skills to analyse, even in a crude way, the complexity of the programs they will write. Don Knuth has argued these points more eloquently that I can in his beautiful book, *Concrete Mathematics*, and I will not elaborate on this anymore.

On a scholarly level, I will argue that some basic mathematical knowledge should be part of the scientific *culture* of any computer science student and more broadly, of any engineering student.

Now, if we believe that computer science students should have some basic mathematical knowledge, what should it be?

There no simple answer. Indeed, students with an interest in algorithms and complexity will need some discrete mathematics such as combinatorics and graph theory but students interested in computer graphics or computer vision will need some geometry and some continuous mathematics. Students interested in data bases will need to know some mathematical logic and students interested in computer architecture will need yet a different brand of mathematics. So, what's the common core?

As I said earlier, most students have a very fuzzy idea of what a proof is. This is actually true of most people! The reason is simple: It is quite difficult to define precisely what a proof

[1] In fact, some people would even argue that such skills constitute a handicap!

is. To do this, one has to define precisely what are the “rules of mathematical reasoning” and this is a lot harder than it looks. Of course, defining and analyzing the notion of proof is a major goal of mathematical logic.

Having attempted some twenty years ago to “demystify” logic for computer scientists and being an incorrigible optimist, I still believe that there is great value in attempting to teach people the basic principles of mathematical reasoning in a precise but not overly formal manner. In these notes, I define the notion of proof as a certain kind of tree whose inner nodes respect certain proof rules presented in the style of a natural deduction system “a la Prawitz”. Of course, this has been done before (for example, in van Dalen [44]) but our presentation has more of a “computer science” flavor which should make it more easily digestible by our intended audience. Using such a proof system, it is easy to describe very clearly what is a proof by contradiction and to introduce the subtle notion of “constructive proof”. We even question the “supremacy” of classical logic, making our students aware of the fact that there isn’t just one logic, but different systems of logic, which often comes as a shock to them.

Having provided a firm foundation for the notion of proof, we proceed with a quick and informal review of the first seven axioms of Zermelo-Fraenkel set theory. Students are usually surprised to hear that axioms are needed to ensure such a thing as the existence of the union of two sets and I respond by stressing that one should always keep a healthy dose of skepticism in life!

What next? Again, my experience has been that most students do not have a clear idea of what a function is, even less of a partial function. Yet, computer programs may not terminate for all input, so the notion of partial function is crucial. Thus, we define carefully relations, functions and partial functions and investigate some of their properties (being injective, surjective, bijective).

One of the major stumbling blocks for students is the notion of proof by induction and its cousin, the definition of functions by recursion. We spend quite a bit of time clarifying these concepts and we give a proof of the validity of the induction principle from the fact that the natural numbers are well-ordered. We also discuss the pigeonhole principle and some basic facts about equinumerosity, without introducing cardinal numbers.

We introduce some elementary concepts of combinatorics in terms of counting problems. We introduce the binomial and multinomial coefficients and study some of their properties and we conclude with the Inclusion-Exclusion Principle.

Next, we introduce partial orders, well-founded sets and complete induction. This way, students become aware of the fact that the induction principle applies to sets with an ordering far more complex that the ordering on the natural numbers. As an application, we prove the unique prime factorization in $\mathbb{Z}$ and discuss GCD’s.

Another extremely important concept is that of an equivalence relation and the related notion of a partition.

We have included some material on lattices, Tarski's fixed point Theorem, distributive lattices, boolean algebras and Heyting algebras. These topics are somewhat more advanced and can be omitted from the "core".

The last topic that we consider crucial is graph theory. We give a fairly complete presentation of the basic concepts of graph theory: directed and undirected graphs, paths, cycles, spanning trees, cocycles, cotrees, flows and tensions, Eulerian and Hamiltonian cycles, matchings, coverings, and planar graphs. We also discuss the network flow problem and prove the Max-Flow Min-Cut Theorem in an original way due to M. Sakarovitch.

These notes grew out of lectures I gave in 2005 while teaching CIS260. There is more material than can be covered in one semester and some choices have to made as to what to omit. Unfortunately, when I taught this course, I was unable to cover any graph theory. I also did not cover lattices and boolean algebras.

My unconventional approach of starting with logic may not work for everybody, as some individuals find such material too abstract. It is possible to skip the chapter on logic and proceed directly with sets functions, *etc.* I admit that I have raised the bar perhaps higher than the average compared to other books on discrete maths. However, my experience when teaching CIS260 was that 70% of the students enjoyed the logic material, as it reminded them of programming. I hope that these notes will inspire and will be useful to motivated students.

Contents

Chapter 1

Mathematical Reasoning, Proof Principles and Logic

1.1 Introduction

Mathematicians write proof; most of us write proofs. This leads to the question: Which principles of reasoning do we use when we write proofs?

The goal of this Chapter is to try answering this question. We do so by formalizing the basic rules of reasoning that we use, most of the time unconsciously, in a certain kind of formalism known as a *natural deduction system*. We give a (very) quick introduction to *mathematical logic*, with a very deliberate *proof-theoretic* bent, that is, neglecting almost completely all semantic notions, except at a very intuitive level. We still feel that this approach is fruitful because the mechanical and rules-of-the-game flavor of proof systems is much more easily grasped than semantic concepts. In this approach, we follow Peter Andrew's motto [1]:

"To truth through proof".

We present various natural deduction systems due to Prawitz and Gentzen (in more modern notation), both in their intuitionistic and classical version. The adoption of natural deduction systems as proof systems makes it easy to question the validity of some of the inference rules, such as the *principle of proof by contradiction*. In brief, we try to explain to our readers the difference between *constructive* and *classical* (i.e., not necessarily constructive) proofs. In this respect, we plant the seed that there is a deep relationship between *constructive proofs* and the notion of *computation* (the "Curry-Howard isomorphism" or "formulae–as–types principle", see Section 1.9 and Howard [30]).

1.2 Inference Rules, Deductions, The Proof Systems $\mathcal{N}_m^{\Rightarrow}$ and $\mathcal{NG}_m^{\Rightarrow}$

In this section, we review some basic proof principles and attempt to clarify, at least informally, what constitutes a mathematical proof.

In order to define the notion of proof rigorously, we would have to define a formal language in which to express statements very precisely and we would have to set up a proof system in terms of axioms and proof rules (also called inference rules). We will not go into this; this would take too much time and besides, this belongs to a logic course, which is not what CIS260 is! Instead, we will content ourselves with an intuitive idea of what a statement is and focus on stating as precisely as possible the rules of logic that are used in constructing proofs. Readers who really want to see a thorough (and rigorous) introduction to logic are referred to Gallier [19] van Dalen [44] or Huth and Ryan [31], a nice text with a Computer Science flavor. A beautiful exposition of logic (from a proof-theoretic point of view) is also given in Troelstra and Schwichtenberg [43], but at a more advanced level. Frank Pfenning has also written an excellent and more extensive introduction to constructive logic. This is available on the web at

http://www.andrew.cmu.edu/course/15-317/handouts/logic.pdf

You should also be aware of CIS482, a very exciting course about logic and its applications in Computer Science. By the way, my book has been out of print for some time but you can get it free (as pdf files) from my logic web site

http://www.cis.upenn.edu/˜jean/gbooks/logic.html

In mathematics, we **prove statements.** Statements may be *atomic* or *compound*, that is, built up from simpler statements using *logical connectives*, such as, *implication* (if–then), *conjunction* (and), *disjunction* (or), *negation* (not) and (existential or universal) *quantifiers*.

As examples of atomic statements, we have:

1. “a student is eager to learn”.
2. “a students wants an A”.
3. “an odd integer is never 0”
4. “the product of two odd integers is odd”

Atomic statements may also contain “variables” (standing for abitrary objects). For example

1. human(x): “x is a human”
2. needs-to-drink(x): “x” needs to drink

An example of a compound statement is

$$\text{human}(x) \Rightarrow \text{needs-to-drink}(x).$$

In the above statement, $\Rightarrow$ is the symbol used for logical implication. If we want to assert that every human needs to drink, we can write

$$\forall x(\text{human}(x) \Rightarrow \text{needs-to-drink}(x));$$

This is read: "for every x, if x is a human then x needs to drink".

If we want to assert that some human needs to drink we write

$$\exists x(\text{human}(x) \Rightarrow \text{needs-to-drink}(x));$$

This is read: "for some x, if x is a human then x needs to drink".

We often denote statements (also called *propositions* or *(logical) formulae*) using letters, such as A, B, P, Q, *etc.*, typically upper-case letters (but sometimes greek letters, φ, ψ, *etc.*).

If P and Q are statements, then their *conjunction* is denoted $P \wedge Q$ (say: P and Q), their *disjunction* denoted $P \vee Q$ (say: P or Q), their *implication* $P \Rightarrow Q$ or $P \supset Q$ (say: if P then Q). Some authors use the symbol $\rightarrow$ and write an implication as $P \rightarrow Q$. We do not like to use this notation because the symbol $\rightarrow$ is already used in the notation for functions ($f\colon A \rightarrow B$). We will mostly use the symbol $\Rightarrow$.

We also have the atomic statements $\perp$ (*falsity*), which corresponds to **false** (think of it as the statement which is false no matter what), and the atomic statement $\top$ (*truth*), which corresponds to **true** (think of it as the statement which is always true). The constant $\perp$ is also called *falsum* or *absurdum*. Then, it is convenient to define the *negation* of P as $P \Rightarrow \perp$ and to abbreviate it as $\neg P$ (or sometimes $\sim P$). Thus, $\neg P$ (say: not P) is just a shorthand for $P \Rightarrow \perp$.

Whenever necessary to avoid ambiguities, we add matching parentheses: $(P \wedge Q)$, $(P \vee Q)$, $(P \Rightarrow Q)$. For example, $P \vee Q \wedge R$ is ambigous; it means either $(P \vee (Q \wedge R))$ or $((P \vee Q) \wedge R)$.

Another important logical operator is *equivalence*. If P and Q are statements, then their *equivalence*, denoted $P \equiv Q$ (or $P \Longleftrightarrow Q$), is an abbreviation for $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$. We often say "$P$ if and only if Q" or even "P iff Q" for $P \equiv Q$. As we will see shortly, to prove a logical equivalence, $P \equiv Q$, we have to prove **both** implications $P \Rightarrow Q$ and $Q \Rightarrow P$.

An implication $P \Rightarrow Q$ should be understood as an if–then statement, that is, if P is true then Q is also true. So, the meaning of negation is that if $\neg P$ holds then P must be false. Otherwise, as $\neg P$ is really $P \Rightarrow \perp$, if P were true, then $\perp$ would have to be true, but this is absurd.

Of course, there are problems with the above paragraph. What does truth have to do with all this? What do we mean when we say "P is true"? What is the relationship between truth and provability?

These are actually deep (and tricky!) questions whose answers are not so obvious. One of the major roles of logic is to clarify the notion of truth and its relationship to provability. We will avoid these fundamental issues by dealing exclusively with the notion of proof. So, the big question is: What is a proof?

Typically, the statements that we prove depend on some set of *hypotheses*, also called *premises* (or *assumptions*). As we shall see shortly, this amounts to proving implications of the form

$$(P_1 \wedge P_2 \wedge \cdots \wedge P_n) \Rightarrow Q.$$

However, there are certain advantages in defining the notion of *proof* (or *deduction*) of a proposition from a set of premises. Sets of premises are usually denoted using upper-case greek letters such as Γ or Δ.

Roughly speaking, a *deduction* of a proposition Q from a set of premises Γ is a finite labeled tree whose root is labeled with Q (the *conclusion*), whose leaves are labeled with premises from Γ (possibly with multiple occurrences), and such that every interior node corresponds to a given set of *proof rules* (or *inference rules*). Certain simple deduction trees are declared as obvious proofs, also called *axioms*.

There are many kinds of proofs systems: Hilbert-style systems, Natural-deduction systems, Gentzen sequents systems, *etc.* We describe a so-called *natural-deduction system* invented by G. Gentzen in the early 1930's (and thoroughly investigated by D. Prawitz in the mid 1960's). The major advantage of this system is that it captures quite nicely the "natural" rules of reasoning that one uses when proving mathematical statements. This does not mean that it is easy to find proofs in such a system or that this system is indeed very intuitive! We begin with the inference rules for implication.

In the definition below, the expression Γ, P stands for the union of Γ and P. So, P may already belong to Γ. A picture such as

$$\frac{\Delta}{P}$$

represents a deduction tree whose root is labeled with P and whose leaves are labeled with propositions from Δ (possibly with multiples occurrences). Some of the propositions in Δ may be tagged be variables. The list of untagged propositions in Δ is the list of *premises* of the deduction tree. For example, in the deduction tree below,

$$\dfrac{\dfrac{P \Rightarrow (R \Rightarrow S) \qquad P}{R \Rightarrow S} \qquad \dfrac{Q \Rightarrow R \qquad \dfrac{P \Rightarrow Q \qquad P}{Q}}{R}}{S}$$

no leaf is tagged, so the premises form the set

$$\Delta = \{P \Rightarrow (R \Rightarrow S), P, Q \Rightarrow R, P \Rightarrow Q\},$$

with two occurrences of P, and the conclusion is S.

Certain inferences rules have the effect that some of the original premises may be discarded; the traditional jargon is that some premises may be *discharged* (or *closed*). This this the case for the inference rule whose conclusion is an implication. When one or several occurrences of some proposition, P, are discharged by an inference rule, these occurrences (which label some leaves) are tagged with some new variable not already appearing in the deduction tree. If x is a new tag, the tagged occurrences of P are denoted P^x and we indicate the fact that premises were discharged by that inference by writing x immediately to the right of the inference bar. For example,

$$\cfrac{\cfrac{P^x, Q}{Q}}{P \Rightarrow Q}\; x$$

is a deduction tree in which the premise P is discharged by the inference rule. This deduction tree only has Q as a premise, since P is discharged.

What is the meaning of the horizontal bars? Actually, nothing really! Here, we are victims of an old habit in logic. Observe that there is always a single proposition immediately under a bar but there may be several propositions immediately above a bar. The intended meaning of the bar is that the proposition below it is obtained as the result of applying an inference rule to the propositions above it. For example, in

$$\frac{Q \Rightarrow R \qquad Q}{R}$$

the proposition R is the result of applying the $\Rightarrow$-elimination rule (see Definition 1.2.1 below) to the two premises $Q \Rightarrow R$ and Q. Thus, the use of the bar is just a convention used by logicians going back at least to the 1900's. Removing the bar everywhere would not change anything to our trees, except perhaps reduce their readability! Since most logic books draw proof trees using bars to indicate inferences, we also use bars in depicting our proof trees.

Since propositions do not arise from the vacuum but instead are built up from a set of atomic propositions using logical connectives (here, $\Rightarrow$), we assume the existence of an "official set of atomic propositions", $\mathbf{PS} = \{\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \cdots\}$. So, for example, $\mathbf{P}_1 \Rightarrow \mathbf{P}_2$ and $\mathbf{P}_1 \Rightarrow (\mathbf{P}_2 \Rightarrow \mathbf{P}_1)$ are propositions. Typically, we will use upper-case letters such as P, Q, R, S, A, B, C, *etc.*, to denote arbitrary propositions formed using atoms from $\mathbf{PS}$.

Definition 1.2.1 The axioms and inference rules for *implicational logic* are:

$$\frac{\Gamma, P}{P}$$

The above is a concise way of denoting a tree whose leaves are labeled with P and the propositions in Γ, each of these proposition (including P) having possibly multiple occurrences but at least one, and whose root is labeled with P. A more explicit form is

$$\frac{\overbrace{P_1, \cdots, P_1}^{k_1}, \cdots, \overbrace{P_i, \cdots, P_i}^{k_i}, \cdots, \overbrace{P_n, \cdots, P_n}^{k_n}}{P_i}$$

where $k_1, \ldots, k_n \geq 0$, $n \geq 1$ and $k_i \geq 1$ for some i with $1 \leq i \leq n$. This axiom says that we always have a deduction of P_i from any set of premises including P_i. Some (or all) of the occurrences of the premises $P_1, \ldots, P_n$ may be tagged with distinct variables.

The $\Rightarrow$-*introduction rule*:

$$\cfrac{\cfrac{\Gamma, P^x}{Q}}{P \Rightarrow Q}\; x$$

This inference rule says that if there is a deduction of Q from the premises in Γ and from the premise P, then there is a deduction of $P \Rightarrow Q$ from Γ. Note that this inference rule has the additional effect of discharging some occurrences of the premise P. These occurrences are tagged with a new variable, x, and the tag x is also placed immediately to the right of the inference bar. This is a reminder that the deduction tree whose conclusion is $P \Rightarrow Q$ no longer has the occurrences of P labeled with x as premises.

The $\Rightarrow$-*elimination rule*:

$$\cfrac{\cfrac{\Gamma}{P \Rightarrow Q} \qquad \cfrac{\Delta}{P}}{Q}$$

This rule is also known as *modus ponens*.

In the above axioms and rules, Γ or Δ may be empty and P, Q denote arbitrary propositions built up from the atoms in **PS**. A *deduction tree* is a tree whose interior nodes correspond to applications of the above inference rules. A *proof tree* is a deduction tree such that *all its premises are discharged*. The above proof system is denoted $\mathcal{N}_m^{\Rightarrow}$ (here, the subscript m stands for *minimal*, referring to the fact that this a bare-bone logical system).

In words, the $\Rightarrow$-introduction rule says that in order to prove an implication $P \Rightarrow Q$ from a set of premises Γ, we assume that P has already been proved, add P to the premises in Γ and then prove Q from Γ and P. Once this is done, the premise P is deleted. This rule formalizes the kind of reasoning that we all perform whenever we prove an implication statement. In that sense, it is a natural and familiar rule, except that we perhaps never stopped to think about what we are really doing. However, the business about discharging the premise P when we are through with our argument is a bit puzzling. Most people probably never carry out this "discharge step" consciously, but such a process does take place implicitly.

It might help to view the action of proving an implication $P \Rightarrow Q$ as the construction of a program that converts a proof of P into a proof of Q. Then, if we supply a proof of P as input to this program (the proof of $P \Rightarrow Q$), it will output a proof of Q. So, if we don't give the right kind of input to this program, for example, a "wrong proof" of P, we should not expect that the program return a proof of Q. However, this does not say that the program is incorrect; the program was designed to do the right thing only if it is given the right kind of input. From this functional point of view (also called, constructive), if we take the simplistic view that P and Q assume the truth values **true** and **false**, we should not be shocked that if we give as input the value **false** (for P), then the truth value of the whole implication $P \Rightarrow Q$ is **true**. The program $P \Rightarrow Q$ is designed to produce the output value **true** (for Q) if it is given the input value **true** (for P). So, this program only goes wrong when, given the input **true** (for P), it returns the value **false** (for Q). In this erroneous case, $P \Rightarrow Q$ should indeed receive the value **false**. However, in all other cases, the program works correctly, even if it is given the wrong input (**false** for P).

☡

1. Only the leaves of a deduction tree may be discharged. Interior nodes, including the root, are *never* discharged.

2. Once a set of leaves labeled with some premise P marked with the label x has been discharged, none of these leaves can be discharged again. So, each label (say x) can only be used once. This corresponds to the fact that some leaves of our deduction trees get "killed off" (discharged).

3. A proof is a deduction tree whose leaves are *all discharged* (Γ is empty). This corresponds to the philosophy that if a proposition has been proved, then the validity of the proof should not depend on any assumptions that are still active. We may think of a deduction tree as an unfinished proof tree.

4. When constructing a proof tree, we have to be careful not to include (accidently) extra premises that end up not beeing discharged. If this happens, we probably made a mistake and the redundant premises should be deleted. On the other hand, if we have a proof tree, we can always add extra premises to the leaves and create a new proof tree from the previous one by discharging all the new premises.

5. Beware, when we deduce that an implication $P \Rightarrow Q$ is provable, we **do not** prove that P **and** Q are provable; we only prove that **if** P is provable **then** Q is provable.

The $\Rightarrow$-elimination rule formalizes the use of *auxiliary lemmas*, a mechanism that we use all the time in making mathematical proofs. Think of $P \Rightarrow Q$ as a lemma that has already been established and belongs to some data base of (useful) lemmas. This lemma says if I can prove P then I can prove Q. Now, suppose that we manage to give a proof of P. It follows from the $\Rightarrow$-elimination rule that Q is also provable.

Observe that in an introduction rule, the conclusion contains the logical connective associated with the rule, in this case, $\Rightarrow$; this justifies the terminology "introduction". On the other hand, in an elimination rule, the logical connective associated with the rule is gone (although it may still appear in Q). The other inference rules for $\wedge$, $\vee$, *etc.*, will follow this pattern of introduction and elimination.

Examples of proof trees.

(a)

$$\dfrac{\dfrac{P^x}{P}}{P \Rightarrow P}\; x$$

So, $P \Rightarrow P$ is provable; this is the least we should expect from our proof system!

(b)

$$\dfrac{\dfrac{\dfrac{\dfrac{(Q \Rightarrow R)^y \qquad \dfrac{(P \Rightarrow Q)^z \qquad P^x}{Q}}{R}}{P \Rightarrow R}\; x}{(Q \Rightarrow R) \Rightarrow (P \Rightarrow R)}\; y}{(P \Rightarrow Q) \Rightarrow ((Q \Rightarrow R) \Rightarrow (P \Rightarrow R))}\; z$$

In order to better appreciate the difference between a deduction tree and a proof tree, consider the following two examples:

1. The tree below is a deduction tree, since two its leaves are labeled with the premises $P \Rightarrow Q$ and $Q \Rightarrow R$, that have not been discharged yet. So, this tree represents a deduction of $P \Rightarrow R$ from the set of premises $\Gamma = \{P \Rightarrow Q, Q \Rightarrow R\}$ but it is *not a proof tree* since $\Gamma \neq \emptyset$. However, observe that the original premise, P, labeled x, has been discharged.

$$\dfrac{\dfrac{Q \Rightarrow R \qquad \dfrac{P \Rightarrow Q \qquad P^x}{Q}}{R}}{P \Rightarrow R}\; x$$

2. The next tree was obtained from the previous one by applying the ⇒-introduction rule which triggered the discharge of the premise $Q \Rightarrow R$ labeled y, which is no longer active. However, the premise $P \Rightarrow Q$ is still active (has not been discharged, yet), so the tree below is a deduction tree of $(Q \Rightarrow R) \Rightarrow (P \Rightarrow R)$ from the set of premises $\Gamma = \{P \Rightarrow Q\}$. It is not yet a proof tree since $\Gamma \neq \emptyset$.

$$\dfrac{\dfrac{\dfrac{(Q \Rightarrow R)^y \qquad \dfrac{P \Rightarrow Q \qquad P^x}{Q}}{R}}{P \Rightarrow R}\; x}{(Q \Rightarrow R) \Rightarrow (P \Rightarrow R)}\; y$$

Finally, one more application of the ⇒-introduction rule will discharged the premise $P \Rightarrow Q$, at last, yielding the proof tree in (b).

(c) In the next example, the two occurrences of A labeled x are discharged simultaneously.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(A \Rightarrow (B \Rightarrow C))^z \qquad A^x}{B \Rightarrow C} \qquad \dfrac{(A \Rightarrow B)^y \qquad A^x}{B}}{C}}{A \Rightarrow C}\; x}{(A \Rightarrow B) \Rightarrow (A \Rightarrow C)}\; y}{\big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big)}\; z$$

(d) In contrast to Example (c), in the proof tree below the two occurrences of A are discharged separately. To this effect, they are labeled differently.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(A \Rightarrow (B \Rightarrow C))^z \qquad A^x}{B \Rightarrow C} \qquad \dfrac{(A \Rightarrow B)^y \qquad A^t}{B}}{C}}{A \Rightarrow C}\; x}{(A \Rightarrow B) \Rightarrow (A \Rightarrow C)}\; y}{\big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big)}\; z}{A \Rightarrow \Big(\big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big)\Big)}\; t$$

Remark: How do we find these proof trees? Well, we could try to enumerate all possible proof trees systematically and see if a proof of the desired conclusion turns up. Obviously, this is a very inefficient procedure and moreover, how do we know that all possible proof trees will be generated and how do we know that such a method will terminate after a finite number of steps (what if the proposition proposed as a conclusion of a proof is not provable)? This is a very difficult problem and, in general, it can be shown that there is **no** procedure that will give an answer in all cases and terminate in a finite number of steps for all possible input propositions. We will come back to this point in Section 1.9. However, for the system $\mathcal{N}_m^{\Rightarrow}$, such a procedure exists, but it is not easy to prove that it terminates in all cases and in fact, it can take a very long time.

What we did, and we strongly advise our readers to try it when they attempt to construct proof trees, is to construct the proof tree from the bottom-up, starting from the proposition labeling the root, rather than top-down, i.e., starting from the leaves. During this process, whenever we are trying to prove a proposition $P \Rightarrow Q$, we use the $\Rightarrow$-introduction rule backward, i.e., we add P to the set of active premises and we try to prove Q from this new set of premises. At some point, we get stuck with an atomic proposition, say R. Call the resulting deduction $\mathcal{D}_{bu}$; note that R is the only active (undischarged) premises of $\mathcal{D}_{bu}$ and the node labeled R immediately below it plays a special role; we will call it the special node of $\mathcal{D}_{bu}$. The trick is to now switch strategy and start building a proof tree top-down, starting from the leaves, using the $\Rightarrow$-elimination rule. If everything works out well, we get a deduction with root R, say $\mathcal{D}_{td}$, and then we glue this deduction $\mathcal{D}_{td}$ to the deduction $\mathcal{D}_{bu}$ in such a way that the root of $\mathcal{D}_{td}$ is identified with the special node of $\mathcal{D}_{bu}$ labeled R. We also have to make sure that all the discharged premises are linked to the correct instance of the $\Rightarrow$-introduction rule that caused them to be discharged. One of the difficulties is that during the bottom-up process, we don't know how many copies of a premise need to be discharged in a single step. We only find out how many copies of a premise need to be discharged during the top-down process.

Here is an illustration of this method for our third example. At the end of the bottom-up process, we get the deduction tree $\mathcal{D}_{bu}$:

$$\cfrac{\cfrac{\cfrac{\cfrac{(A \Rightarrow (B \Rightarrow C))^z \qquad (A \Rightarrow B)^y \qquad A^x \qquad C}{C}}{A \Rightarrow C}\; x}{(A \Rightarrow B) \Rightarrow (A \Rightarrow C)}\; y}{\big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big)}\; z$$

At the end of the top-down process, we get the deduction tree $\mathcal{D}_{td}$:

$$\cfrac{\cfrac{A \Rightarrow (B \Rightarrow C) \qquad A}{B \Rightarrow C} \qquad \cfrac{A \Rightarrow B \qquad A}{B}}{C}$$

Finally, after gluing $\mathcal{D}_{td}$ on top of $\mathcal{D}_{bu}$ (which has the correct number of premises to be discharged), we get our proof tree:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(A \Rightarrow (B \Rightarrow C))^z \qquad A^x}{B \Rightarrow C} \qquad \dfrac{(A \Rightarrow B)^y \qquad A^x}{B}}{C}}{A \Rightarrow C}\; x}{(A \Rightarrow B) \Rightarrow (A \Rightarrow C)}\; y}{\big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big)}\; z$$

Let us return to the functional interpretation of implication by giving an example. The proposition $P \Rightarrow ((P \Rightarrow Q) \Rightarrow Q)$ has the following proof:

$$\dfrac{\dfrac{\dfrac{(P \Rightarrow Q)^x \qquad P^y}{Q}}{(P \Rightarrow Q) \Rightarrow Q}\; x}{P \Rightarrow ((P \Rightarrow Q) \Rightarrow Q)}\; y$$

Now, say P is the proposition $R \Rightarrow R$, which has the proof

$$\dfrac{\dfrac{R^z}{R}}{R \Rightarrow R}\; z$$

Using $\Rightarrow$-elimination, we obtain a proof of $((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q$ from the proof of $(R \Rightarrow R) \Rightarrow (((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q)$ and the proof of $R \Rightarrow R$:

$$\dfrac{\dfrac{\dfrac{\dfrac{((R \Rightarrow R) \Rightarrow Q)^x \qquad (R \Rightarrow R)^y}{Q}}{((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q}\; x}{(R \Rightarrow R) \Rightarrow (((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q)}\; y \qquad \dfrac{\dfrac{R^z}{R}}{R \Rightarrow R}\; z}{((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q}$$

Note that the above proof is redundant. A more direct proof can be obtained as follows: Undo the last $\Rightarrow$-introduction in the proof of $(R \Rightarrow R) \Rightarrow (((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q)$:

$$\dfrac{\dfrac{((R \Rightarrow R) \Rightarrow Q)^x \qquad R \Rightarrow R}{Q}}{((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q}\; x$$

and then glue the proof of $R \Rightarrow R$ on top of the leaf $R \Rightarrow R$, obtaining the desired proof of $((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q$:

$$\dfrac{\dfrac{((R \Rightarrow R) \Rightarrow Q)^x \qquad \dfrac{\dfrac{R^z}{R}}{R \Rightarrow R}\; z}{Q}}{((R \Rightarrow R) \Rightarrow Q) \Rightarrow Q}\; x$$

In general, one has to exercise care with the label variables. It may be necessary to rename some of these variables to avoid clashes. What we have above is an example of *proof substitution* also called *proof normalization*. We will come back to this topic in Section 1.9.

The process of discharging premises when constructing a deduction is admittedly a bit confusing. Part of the problem is that a deduction tree really represents the last of a sequence of stages (corresponding to the application of inference rules) during which the current set of “active” premises, that is, those premises that have not yet been discharged (closed, cancelled) evolves (in fact, shrinks). Some mechanism is needed to keep track of which premises are no longer active and this is what this business of labeling premises with variables achieves. Historically, this is the first mechanism that was invented. However, Gentzen (in the 1930’s) came up with an alternative solution which is mathematically easier to handle. Moreover, it turns out that this notation is also better suited to computer implementations, if one wishes to implement an automated theorem prover.

The point is to keep a record of all undischarged assumptions at every stage of the deduction. Thus, a deduction is now a tree whose nodes are labeled with expressions of the form $\Gamma \rightarrow P$, called *sequents*, where P is a proposition, and Γ is a record of all undischarged assumptions at the stage of the deduction associated with this node.

During the construction of a deduction tree, it is necessary to discharge packets of assumptions consisting of one or more occurrences of the same proposition. To this effect, it is convenient to tag packets of assumptions with labels, in order to discharge the propositions in these packets in a single step. We use variables for the labels, and a packet labeled with x consisting of occurrences of the proposition P is written as $x\colon P$. Thus, in a sequent $\Gamma \rightarrow P$, the expression Γ is any finite set of the form $x_1\colon P_1, \ldots, x_m\colon P_m$, where the x_i are pairwise distinct (but the P_i need not be distinct). Given $\Gamma = x_1\colon P_1, \ldots, x_m\colon P_m$, the notation $\Gamma, x\colon P$ is only well defined when $x \neq x_i$ for all i, $1 \leq i \leq m$, in which case it denotes the set $x_1\colon P_1, \ldots, x_m\colon P_m, x\colon P$.

Using sequents, the axioms and rules of Definition 1.2.2 are now expressed as follows:

Definition 1.2.2 The axioms and inference rules of the system $\mathcal{NG}_m^{\Rightarrow}$ (*implicational logic, Gentzen-sequent style (the $\mathcal{G}$ in $\mathcal{NG}$ stands for Gentzen)*) are listed below:

$$\Gamma, x\colon P \to P$$

$$\frac{\Gamma, x\colon P \to Q}{\Gamma \to P \Rightarrow Q} \quad (\Rightarrow\textit{-intro})$$

$$\frac{\Gamma \to P \Rightarrow Q \quad \Gamma \to P}{\Gamma \to Q} \quad (\Rightarrow\textit{-elim})$$

In an application of the rule ($\Rightarrow$-*intro*), observe that in the lower sequent, the proposition P (labeled x) is deleted from the list of premises occurring on the left-hand side of the arrow in the upper sequent. We say that the proposition P which appears as a hypothesis of the deduction is *discharged* (or *closed*). It is important to note that the ability to label packets consisting of occurrences of the same proposition with different labels is essential, in order to be able to have control over which groups of packets of assumptions are discharged simultaneously. Equivalently, we could avoid tagging packets of assumptions with variables if we assumed that in a sequent $\Gamma \to C$, the expression Γ, also called a *context*, is a *multiset* of propositions.

Below we show a proof of the third example given above in our new system. Let

$$\Gamma = x\colon A \Rightarrow (B \Rightarrow C), y\colon A \Rightarrow B, z\colon A.$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\Gamma \to A \Rightarrow (B \Rightarrow C) \qquad \Gamma \to A}{\Gamma \to B \Rightarrow C} \qquad \cfrac{\Gamma \to A \Rightarrow B \qquad \Gamma \to A}{\Gamma \to B}}{x\colon A \Rightarrow (B \Rightarrow C), y\colon A \Rightarrow B, z\colon A \to C}}{x\colon A \Rightarrow (B \Rightarrow C), y\colon A \Rightarrow B \to A \Rightarrow C}}{x\colon A \Rightarrow (B \Rightarrow C) \to (A \Rightarrow B) \Rightarrow (A \Rightarrow C)}}{\to \big(A \Rightarrow (B \Rightarrow C)\big) \Rightarrow \big((A \Rightarrow B) \Rightarrow (A \Rightarrow C)\big)}$$

Remark: An attentive reader will have surely noticed that the second version of the $\Rightarrow$-elimination rule,

$$\frac{\Gamma \to P \Rightarrow Q \quad \Gamma \to P}{\Gamma \to Q} \quad (\Rightarrow\textit{-elim}),$$

differs slightly from the first version given in Definition 1.2.1. Indeed, in Prawitz's style, the rule that matches exactly the $\Rightarrow$-elim rule above is

$$\dfrac{\dfrac{\Gamma}{P \Rightarrow Q} \qquad \dfrac{\Gamma}{P}}{Q}$$

where the deductions of $P \Rightarrow Q$ and P have the *same* set of premises, Γ. Equivalently, the rule in sequent-format that corresponds to the $\Rightarrow$-elimination rule of Definition 1.2.1 is

$$\frac{\Gamma \to P \Rightarrow Q \quad \Delta \to P}{\Gamma, \Delta \to Q} \quad (\Rightarrow\text{-}\mathit{elim}'),$$

where Γ, Δ must be interpreted as the union of Γ and Δ.

A moment of reflexion will reveal that the resulting proofs systems are equivalent (that is, every proof in one system can converted to a proof in the other system). The version of the $\Rightarrow$-elimination rule in Definition 1.2.1 may be considered preferable because it gives us the ability to make the sets of premises labeling leaves smaller. On the other hand, after experimenting with the construction of proofs, one gets the feeling that every proof can be simplified to a "unique minimal" proof, if we define "minimal" in a suitable sense, namely, that a minimal proof never contains an elimination rule immediately following an introduction rule (for more on this, see Section 1.9). Then, it turns out that to define the notion of uniqueness of proofs, the second version is preferable. However, it is important to realize that in general, a proposition may possess distinct minimal proofs!

In principle, it does not matter which of the two systems $\mathcal{N}_m^{\Rightarrow}$ or $\mathcal{NG}_m^{\Rightarrow}$ we use to construct deductions; it is basically a matter of taste. My experience is that I make fewer mistakes with the Gentzen-sequent style system $\mathcal{NG}_m^{\Rightarrow}$.

We now describe the inference rules dealing with the connectives $\wedge$, $\vee$ and $\perp$.

1.3 Adding $\wedge$, $\vee$, $\perp$; The Proof Systems $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp}$ and $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp}$

Recall that $\neg P$ is an abbreviation for $P \Rightarrow \perp$.

Definition 1.3.1 The axioms and inference rules for *(propositional) classical logic* are:

Axioms:

$$\frac{\Gamma, P}{P}$$

The $\Rightarrow$-*introduction rule*:

$$\dfrac{\dfrac{\Gamma, P^x}{Q}}{P \Rightarrow Q}\; x$$

The $\Rightarrow$-*elimination rule*:

$$\dfrac{\dfrac{\Gamma}{P \Rightarrow Q} \qquad \dfrac{\Delta}{P}}{Q}$$

The $\wedge$-*introduction rule*:

$$\dfrac{\dfrac{\Gamma}{P} \qquad \dfrac{\Delta}{Q}}{P \wedge Q}$$

The $\wedge$-*elimination rule*:

$$\dfrac{\dfrac{\Gamma}{P \wedge Q}}{P} \qquad\qquad \dfrac{\dfrac{\Gamma}{P \wedge Q}}{Q}$$

The $\vee$-*introduction rule*:

$$\dfrac{\dfrac{\Gamma}{P}}{P \vee Q} \qquad\qquad \dfrac{\dfrac{\Gamma}{Q}}{P \vee Q}$$

The $\vee$-*elimination rule*:

$$\dfrac{\dfrac{\Gamma}{P \vee Q} \qquad \dfrac{\Delta, P^x}{R} \qquad \dfrac{\Lambda, Q^y}{R}}{R}\; {\scriptstyle x,y}$$

The $\perp$-*elimination rule*:

$$\dfrac{\dfrac{\Gamma}{\perp}}{P}$$

The *proof-by-contradiction rule* (also known as *reductio ad absurdum rule*, for short *RAA*):

$$\dfrac{\dfrac{\Gamma, \neg P^x}{\perp}}{P}\; {\scriptstyle x}$$

Since $\neg P$ is an abbreviation for $P \Rightarrow \perp$, the $\neg$-introduction rule is a special case of the $\Rightarrow$-introduction rule (with $Q = \perp$). However, it is worth stating it explicitly:

The $\neg$-*introduction rule*:

$$\dfrac{\dfrac{\Gamma, P^x}{\bot}}{\neg P}\ x$$

Similarly, the $\neg$-elimination rule is a special case of $\Rightarrow$-elimination applied to $\neg P$ $(= P \Rightarrow \bot)$ and P:

The $\neg$-*elimination rule*:

$$\dfrac{\dfrac{\Gamma}{\neg P} \qquad \dfrac{\Delta}{P}}{\bot}$$

In the above axioms and rules, Γ, Δ or Λ may be empty, P, Q, R denote arbitrary propositions built up from the atoms in **PS** and all the premises labeled x are discharged. A *deduction tree* is a tree whose interior nodes correspond to applications of the above inference rules. A *proof tree* is a deduction tree such that *all its premises* are discharged. The above proof system is denoted $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\bot}$ (here, the subscript c stands for *classical*).

The system obtained by removing the proof-by-contradiction (RAA) rule is called *(propositional) intuitionistic logic* and is denoted $\mathcal{N}_i^{\Rightarrow,\wedge,\vee,\bot}$. The system obtained by deleting both the $\bot$-elimination rule and the proof-by-contradiction rule is called *(propositional) minimal logic* and is denoted $\mathcal{N}_m^{\Rightarrow,\wedge,\vee,\bot}$.

The version of $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\bot}$ in terms of Gentzen sequents is the following:

Definition 1.3.2 The axioms and inference rules of the system $\mathcal{NG}_i^{\Rightarrow,\wedge,\vee,\bot}$ (of *propositional classical logic, Gentzen-sequent style*) are listed below:

$$\Gamma, x \colon P \to P$$

$$\dfrac{\Gamma, x \colon P \to Q}{\Gamma \to P \Rightarrow Q} \quad (\Rightarrow\textit{-intro})$$

$$\dfrac{\Gamma \to P \Rightarrow Q \quad \Gamma \to P}{\Gamma \to Q} \quad (\Rightarrow\textit{-elim})$$

$$\dfrac{\Gamma \to P \quad \Gamma \to Q}{\Gamma \to P \wedge Q} \quad (\wedge\textit{-intro})$$

$$\dfrac{\Gamma \to P \wedge Q}{\Gamma \to P} \quad (\wedge\textit{-elim}) \qquad \dfrac{\Gamma \to P \wedge Q}{\Gamma \to Q} \quad (\wedge\textit{-elim})$$

$$\dfrac{\Gamma \to P}{\Gamma \to P \vee Q} \quad (\vee\textit{-intro}) \qquad \dfrac{\Gamma \to Q}{\Gamma \to P \vee Q} \quad (\vee\textit{-intro})$$

$$\frac{\Gamma \to P \vee Q \quad \Gamma, x\colon P \to R \quad \Gamma, y\colon Q \to R}{\Gamma \to R} \quad (\vee\text{-}elim)$$

$$\frac{\Gamma \to \perp}{\Gamma \to P} \quad (\perp\text{-}elim)$$

$$\frac{\Gamma, x\colon \neg P \to \perp}{\Gamma \to P} \quad (by\text{-}contra)$$

$$\frac{\Gamma, x\colon P \to \perp}{\Gamma \to \neg P} \quad (\neg\text{-introduction})$$

$$\frac{\Gamma \to \neg P \quad \Gamma \to P}{\Gamma \to \perp} \quad (\neg\text{-elimination})$$

Since the rule ($\perp$-*elim*) is trivial (does nothing) when $P = \perp$, from now on, we will assume that $P \neq \perp$. *Propositional minimal logic*, denoted $\mathcal{NG}_m^{\Rightarrow,\wedge,\vee,\perp}$, is obtained by dropping the ($\perp$-*elim*) and (*by-contra*) rules. *Propositional intuitionistic logic*, denoted $\mathcal{NG}_i^{\Rightarrow,\wedge,\vee,\perp}$, is obtained by dropping the (*by-contra*) rule.

When we say that a proposition, P, is *provable from* Γ, we mean that we can construct a proof tree whose conclusion is P and whose set of premises is Γ, in one of the systems $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp}$ or $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp}$. Therefore, when we use the word "provable" unqualified, we mean provable in *classical logic*. If P is provable from Γ in one of the intuitionistic systems $\mathcal{N}_i^{\Rightarrow,\wedge,\vee,\perp}$ or $\mathcal{NG}_i^{\Rightarrow,\wedge,\vee,\perp}$, then we say *intuitionistically provable* (and similarly, if P is provable from Γ in one of the systems $\mathcal{N}_m^{\Rightarrow,\wedge,\vee,\perp}$ or $\mathcal{NG}_m^{\Rightarrow,\wedge,\vee,\perp}$, then we say *provable in minimal logic*). When P is provable from Γ, most people write $\Gamma \vdash P$, or $\vdash \Gamma \to P$, sometimes with the name of the corresponding proof system tagged as a subscript on the sign $\vdash$ if necessary to avoid ambiguities. When Γ is empty, we just say P is provable (provable in intuitionistic logic, *etc.*) and write $\vdash P$.

We treat *logical equivalence* as a derived connective, that is, we view $P \equiv Q$ as an abbreviation for $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$. In view of the inference rules for $\wedge$, we see that to prove a logical equivalence $P \equiv Q$, we just have to prove both implications $P \Rightarrow Q$ and $Q \Rightarrow P$.

In view of the $\neg$-elimination rule, we may be tempted to interpret the provability of a negation, $\neg P$, is as "P is not provable". Indeed, if $\neg P$ and P were both provable, then $\perp$ would be provable. So, P should not be provable if $\neg P$ is. However, if P is not provable, then $\neg P$ is **not** provable in general! There are plenty of propositions such that neither P nor $\neg P$ is provable (for instance, P, with P an atomic proposition). Thus, the fact that P is not provable is not equivalent to the provability of $\neg P$ and we should not interpret $\neg P$ as "P is not provable".

Let us now make some (much-needed) comments about the above inference rules. There is no need to repeat our comments regarding the $\Rightarrow$-rules.

The $\wedge$-introduction rule says that in order to prove a conjunction $P \wedge Q$ from some premises Γ, all we have to do is to prove *both* that P is provable from Γ *and* that Q is provable from Γ. The $\wedge$-elimination rule says that once we have proved $P \wedge Q$ from Γ, then P (and Q) is also provable from Γ. This makes sense intuitively as $P \wedge Q$ is "stronger" than P and Q separately ($P \wedge Q$ is true iff both P and Q are true).

The $\vee$-introduction rule says that if P (or Q) has been proved from Γ, then $P \vee Q$ is also provable from Γ. Again, this makes sense intuitively as $P \vee Q$ is "weaker" than P and Q. The $\vee$-elimination rule formalizes the *proof-by-cases* method. It is a more subtle rule. The idea is that if we know that in the case where P is already assumed to be provable and similarly in the case where Q is already assumed to be provable that we can prove R (also using premises in Γ), then if $P \vee Q$ is also provable from Γ, as we have "covered both cases", it should be possible to prove R from Γ only (i.e., the premises P and Q are discarded).

The $\perp$-elimination rule formalizes the principle that once a false statement has been established, then anything should be provable.

The proof-by-contradiction rule formalizes the method of proof by contradiction! That is, in order to prove that P can be deduced from some premises Γ, one may assume the negation, $\neg P$, of P (intuitively, assume that P is false) and then derive a contradiction from Γ and $\neg P$ (i.e., derive falsity). Then, P actually follows from Γ *without using* $\neg P$ *as a premise*, i.e., $\neg P$ is discharged.

Most people, I believe, will be comfortable with the rules of minimal logic and will agree that they constitute a "reasonable" formalization of the rules of reasoning involving $\Rightarrow$, $\wedge$ and $\vee$. Indeed, these rules seem to express the intuitive meaning of the connectives $\Rightarrow$, $\wedge$ and $\vee$. However, some may question the two rules $\perp$-elimination and proof-by-contradiction. Indeed, their meaning is not as clear and, certainly, the proof-by-contradiction rule introduces a form of indirect reasoning that is somewhat worrisome.

The problem has to do with the meaning of disjunction and negation and more generally, with the notion of *constructivity* in mathematics. In fact, in the early 1900's, some mathematicians, especially L. Brouwer (1881-1966), questioned the validity of the proof-by-contradiction rule, among other principles. Two specific cases illustrate the problem, namely, the propositions

$$P \vee \neg P \quad \text{and} \quad \neg\neg P \Rightarrow P.$$

As we will see shortly, the above propositions are both provable in classical logic. Now, Brouwer and some mathematicians belonging to his school of thoughts (the so-called "intuitionsists" or "constructivists") advocate that in order to prove a disjunction, $P \vee Q$ (from some premises Γ) one has to either exhibit a proof of P or a proof or Q (from Γ). However, it can be shown that this fails for $P \vee \neg P$. The fact that $P \vee \neg P$ is provable (in classical logic) **does not** imply (in general) that either P is provable or that $\neg P$ is provable! That $P \vee \neg P$ is provable is sometimes called the *principle of the excluded middle*! In intuitionistic logic,

$P \vee \neg P$ is **not** provable (in general). Of course, if one gives up the proof-by-contradiction rule, then fewer propositions become provable. On the other hand, one may claim that the propositions that remain provable have more constructive proofs and thus, feels on safer grounds.

A similar controversy arises with $\neg\neg P \Rightarrow P$. If we give up the proof-by-contradiction rule, then this formula is no longer provable, i.e., $\neg\neg P$ is no longer equivalent to P. Perhaps this relates to the fact that if one says

" I don't have no money"

then this does not mean that this person has money! (Similarly with "I don't get no satisfaction", ...). However, note that one can still prove $P \Rightarrow \neg\neg P$ in minimal logic (try doing it!). Even stranger, $\neg\neg\neg P \Rightarrow \neg P$ is provable in intuitionistic (and minimal) logic, so $\neg\neg\neg P$ and $\neg P$ are equivalent intuitionistically!

Remark: Suppose we have a deduction

$$\frac{\Gamma, \neg P}{\perp}$$

as in the proof by contradiction rule. Then, by $\neg$-introduction, we get a deduction of $\neg\neg P$ from Γ:

$$\dfrac{\dfrac{\Gamma, \neg P^x}{\perp}}{\neg\neg P}\; x$$

So, if we knew that $\neg\neg P$ was equivalent to P (actually, if we knew that $\neg\neg P \Rightarrow P$ is provable) then the proof by contradiction rule would be justified as a valid rule (it follows from modus ponens). We can view the proof by contradiction rule as a sort of act of faith that consists in saying that if we can derive an inconsistency (i.e., chaos) by assuming the falsity of a statement P, then P has to hold in the first place. It not so clear that such an act of faith is justified and the intuitionists refuse to take it!

Constructivity in mathematics is a fascinating subject but it is a topic that is really outside the scope of this course. What we hope is that our brief and very incomplete discussion of constructivity issues made the reader aware that the rules of logic are not cast in stone and that, in particular, there isn't **only one** logic.

We feel safe in saying that most mathematicians work with classical logic and only few of them have reservations about using the proof-by-contradiction rule. Nevertherless, intuitionistic logic has its advantages, especially when it comes to proving the correctess of programs (a branch of computer science!). We will come back to this point several times in this course.

In the rest of this section, we make further useful remarks about (classical) logic and give some explicit examples of proofs illustrating the inference rules of classical logic. We begin by proving that $P \lor \neg P$ is provable in classical logic.

Proposition 1.3.3 *The proposition $P \lor \neg P$ is provable in classical logic.*

Proof. We prove that $P \lor (P \Rightarrow \perp)$ is provable by using the proof-by-contradiction rule as shown below:

$$\dfrac{\dfrac{((P \lor (P \Rightarrow \perp)) \Rightarrow \perp)^y \qquad \dfrac{\dfrac{\dfrac{((P \lor (P \Rightarrow \perp)) \Rightarrow \perp)^y \qquad \dfrac{P^x}{P \lor (P \Rightarrow \perp)}}{\perp}}{P \Rightarrow \perp}\; x}{P \lor (P \Rightarrow \perp)}}{\perp}}{P \lor (P \Rightarrow \perp)}\; y \text{ (by-contra)}$$

□

Next, we consider the equivalence of P and $\neg\neg P$.

Proposition 1.3.4 *The proposition $P \Rightarrow \neg\neg P$ is provable in minimal logic. The proposition $\neg\neg P \Rightarrow P$ is provable in classical logic. Therefore, in classical logic, P is equivalent to $\neg\neg P$.*

Proof. We leave that $P \Rightarrow \neg\neg P$ is provable in minimal logic as an exercise. Below is a proof of $\neg\neg P \Rightarrow P$ using the proof-by-contradiction rule:

$$\dfrac{\dfrac{\dfrac{((P \Rightarrow \perp) \Rightarrow \perp)^y \qquad (P \Rightarrow \perp)^x}{\perp}}{P}\; x \text{ (by-contra)}}{((P \Rightarrow \perp) \Rightarrow \perp) \Rightarrow P}\; y$$

□

The next proposition shows why $\perp$ can be viewed as the “ultimate” contradiction.

Proposition 1.3.5 *In intuitionistic logic, the propositions $\perp$ and $P \land \neg P$ are equivalent for all P. Thus, $\perp$ and $P \land \neg P$ are also equivalent in classical propositional logic*

Proof. We need to show that both $\perp \Rightarrow (P \land \neg P)$ and $(P \land \neg P) \Rightarrow \perp$ are provable in intuitionistic logic. The provability of $\perp \Rightarrow (P \land \neg P)$ is an immediate consequence or $\perp$-elimination, with $\Gamma = \emptyset$. For $(P \land \neg P) \Rightarrow \perp$, we have the following proof:

$$\dfrac{\dfrac{\dfrac{(P \land \neg P)^x}{\neg P} \qquad \dfrac{(P \land \neg P)^x}{P}}{\bot}}{(P \land \neg P) \Rightarrow \bot}\ x$$

□

So, in intuitionistic logic (and also in classical logic), $\bot$ is equivalent to $P \land \neg P$ for all P. This means that $\bot$ is the "ultimate" contradiction, it corresponds to total inconsistency. By the way, we could have the bad luck that the system $\mathcal{N}_c^{\Rightarrow,\land,\lor,\bot}$ (or $\mathcal{N}_i^{\Rightarrow,\land,\lor,\bot}$ or even $\mathcal{N}_m^{\Rightarrow,\land,\lor,\bot}$) is *inconsistent*, that is, that $\bot$ is provable! Fortunately, this is not the case, although this is hard to prove. (It is also the case that $P \lor \neg P$ and $\neg\neg P \Rightarrow P$ are **not** provable in intuitionistic logic, but this too is hard to prove!)

1.4 Clearing Up Differences Between ¬-introduction, ⊥-elimination and RAA

The differences between the rules, ¬-introduction, ⊥-elimination and the proof by contradiction rule (RAA) are often unclear to the uninitiated reader and this tends to cause confusion. In this section, we will try to clear up some common misconceptions about these rules.

Confusion 1. Why is RAA not a special case of ¬-introduction?

$$\dfrac{\dfrac{\Gamma, P^x}{\bot}}{\neg P}\ x\,(\neg\text{-intro}) \qquad\qquad \dfrac{\dfrac{\Gamma, \neg P^x}{\bot}}{P}\ x\,(\text{RAA})$$

The only apparent difference between ¬-introduction (on the left) and RAA (on the right) is that in RAA, the premise P is negated but the conclusion is not, whereas in ¬-introduction the premise P is not negated but the conclusion is.

The important difference is that the conclusion of RAA is **not** negated. If we had applied ¬-introduction instead of RAA on the right, we would have obtained

$$\dfrac{\dfrac{\Gamma, \neg P^x}{\bot}}{\neg\neg P}\ x\,(\neg\text{-intro})$$

where the conclusion would have been $\neg\neg P$ as opposed to P. However, as we already said earlier, $\neg\neg P \Rightarrow P$ is **not** provable intuitionistically. Consequenly, RAA **is not** a special case of ¬-introduction.

Confusion 2. Is there any difference between ⊥-elimination and RAA?

$$\cfrac{\cfrac{\Gamma}{\bot}}{P}\ (\bot\text{-elim}) \qquad\qquad \cfrac{\cfrac{\Gamma, \neg P^x}{\bot}}{P}\ {}_x\,(\text{RAA})$$

The difference is that $\bot$-elimination does not discharge any of its premises. In fact, RAA is a stronger rule which implies $\bot$-elimination as we now demonstate.

RAA implies $\bot$-elimination.

Suppose we have a deduction

$$\cfrac{\Gamma}{\bot}$$

Then, for any proposition P, we can add the premise $\neg P$ to every leaf of the above deduction tree and we get the deduction tree

$$\cfrac{\Gamma, \neg P}{\bot}$$

We can now apply RAA to get the following deduction tree of P from Γ (since $\neg P$ is discharged), and this is just the result of $\bot$-elimination:

$$\cfrac{\cfrac{\Gamma, \neg P^x}{\bot}}{P}\ {}_x\,(\text{RAA})$$

The above considerations also show that RAA is obtained from $\neg$-introduction by adding the new rule of $\neg\neg$-*elimination*:

$$\cfrac{\cfrac{\Gamma}{\neg\neg P}}{P}\quad (\neg\neg\text{-elimination})$$

Some authors prefer adding the $\neg\neg$-elimination rule to intuitionistic logic instead of RAA in order to obtain classical logic. As we just demonstrated, the two additions are equivalent: by adding either RAA or $\neg\neg$-elimination to intuitionistic logic, we get classical logic.

There is another way to obtain RAA from the rules of intuitionistic logic, this time, using the propositions of the form $P \vee \neg P$. We saw in Proposition 1.3.3 that all formulae of the form $P \vee \neg P$ are provable in classical logic (using RAA).

Confusion 3. Are propositions of the form $P \vee \neg P$ provable in intuitionistic logic?

The answer is **no**, which may be disturbing to some readers. In fact, it is quite difficult to prove that propositions of the form $P \vee \neg P$ are not provable in intuitionistic logic. One method consists in using the fact that intuitionistic proofs can be normalized (see Section 1.9 for more on normalization of proofs). Another method uses Kripke models (see Section 1.7 and van Dalen [44]).

Part of the difficulty in understanding at some intuitive level why propositions of the form $P \vee \neg P$ are not provable in intuitionistic logic is that the notion of truth based on the truth values **true** and **false** is deeply rooted in all of us. In this frame of mind, it seems ridiculous to question the provability of $P \vee \neg P$, since its truth value is **true** whether P is assigned the value **true** or **false**. Classical two-valued truth values semantics is too crude for intuitionistic logic.

Another difficulty is that it is tempting to equate the notion of truth and the notion of provability. Unfortunately, because classical truth values semantics is too crude for intuitionistic logic, there are propositions that are universally true (i.e., they evaluate to **true** for all possible truth assignments of the atomic letters in them) and yet they are **not** provable intuitionistically. The propositions $P \vee \neg P$ and $\neg\neg P \Rightarrow P$ are such examples.

One of the major motivations for advocating intuitionistic logic is that it yields proofs that are more constructive than classical proofs. For example, in classical logic, when we prove a disjunction $P \vee Q$, we generally can't conclude that either P or Q is provable, as examplified by $P \vee \neg P$. A more interesting example involving a non-constructive proof of a disjunction will be given in Section 1.5. But, in intuitionistic logic, from a proof of $P \vee Q$, it is possible to extract either a proof of P or a proof or Q (and similarly for existential statements, see Section 1.8). This property is not easy to prove. It is a consequence of the normal form for intuitionistic proofs (see Section 1.9).

In brief, besides being a fun intellectual game, intuitionistic logic is only an interesting alternative to classical logic if we care about the constructive nature of our proofs. But then, we are forced to abandon the classical two-valued truth values semantics and adopt other semantics such as Kripke semantics. If we do not care about the constructive nature of our proofs and if we want to stick to two-valued truth values semantics, then we should stick to classical logic. Most people do that, so don't feel bad if you are not comfortable with intuitionistic logic!

One way to gauge how intuitionisic logic differs from classical logic is to ask what kind of propositions need to be added to intuitionisic logic in order to get classical logic. It turns out that if all the propositions of the form $P \vee \neg P$ are considered to be axioms, then RAA follows from some of the rules of intuitionistic logic.

RAA holds in Intuitionistic logic + all axioms $P \vee \neg P$.

The proof involves a subtle use of the ⊥-elimination and ∨-elimination rules which may be a bit puzzling. Assume, as we do when when use the proof by contradiction rule (RAA) that we have a deduction

$$\frac{\Gamma, \neg P}{\perp}$$

Here is the deduction tree demonstrating that RAA is a derived rule:

$$\dfrac{P \lor \neg P \qquad \dfrac{P^x}{P} \qquad \dfrac{\dfrac{\Gamma, \neg P^y}{\bot}}{P}\ (\bot\text{-elim})}{P}\ {\scriptstyle x,y}\ (\lor\text{-elim})$$

At first glance, the rightmost subtree

$$\dfrac{\dfrac{\Gamma, \neg P^y}{\bot}}{P}\ (\bot\text{-elim})$$

appears to use RAA and our argument looks circular! But this is not so because the premise $\neg P$ labeled y is *not* discharged in the step that yields P as conclusion; the step that yields P is a $\bot$-elimination step. The premise $\neg P$ labeled y is actually discharged by the $\lor$-elimination rule (and so is the premise P labeled x). So, our argument establishing RAA is not circular after all!

In conclusion, intuitionistic logic is obtained from classical logic by *taking away the proof by contradiction rule (RAA)*. In this more restrictive proof system, we obtain more constructive proofs. In that sense, the situation is better than in classical logic. The major drawback is that we can't think in terms of classical truth values semantics anymore.

Conversely, classical logic is obtained from intuitionistic logic in at least three ways:

1. Add the proof by contradiction rule (RAA).

2. Add the $\neg\neg$-elimination rule.

3. Add all propositions of the form $P \lor \neg P$ as axioms.

1.5 Other Rules of Classical Logic and Examples of Proofs

In classical logic, we have the de Morgan laws:

Proposition 1.5.1 *The following equivalences (de Morgan laws) are provable in classical logic:*

$$\neg(P \land Q) \equiv \neg P \lor \neg Q$$
$$\neg(P \lor Q) \equiv \neg P \land \neg Q.$$

In fact, $\neg(P \lor Q) \equiv \neg P \land \neg Q$ and $(\neg P \lor \neg Q) \Rightarrow \neg(P \land Q)$ are provable in intuitionistic logic. The proposition $(P \land \neg Q) \Rightarrow \neg(P \Rightarrow Q)$ is provable in intuitionistic logic and $\neg(P \Rightarrow Q) \Rightarrow (P \land \neg Q)$ is provable in classical logic. Therefore, $\neg(P \Rightarrow Q)$ and $P \land \neg Q$ are equivalent in classical logic. Furthermore, $P \Rightarrow Q$ and $\neg P \lor Q$ are equivalent in classical logic and $(\neg P \lor Q) \Rightarrow (P \Rightarrow Q)$ is provable in intuitionistic logic.

Proof. Here is an intuitionistic proof of $(\neg P \lor Q) \Rightarrow (P \Rightarrow Q)$:

$$\dfrac{\dfrac{(\neg P \lor Q)^w \qquad \dfrac{\dfrac{\dfrac{\neg P^z \qquad P^x}{\bot}}{Q}}{P \Rightarrow Q}{\scriptstyle x} \qquad \dfrac{\dfrac{P^y \qquad Q^t}{Q}}{P \Rightarrow Q}{\scriptstyle y}}{P \Rightarrow Q}{\scriptstyle z,t}}{(\neg P \lor Q) \Rightarrow (P \Rightarrow Q)}{\scriptstyle w}$$

Here is a classical proof of $(P \Rightarrow Q) \Rightarrow (\neg P \lor Q)$:

$$\dfrac{\dfrac{\dfrac{(\neg(\neg P \lor Q))^y \qquad \dfrac{\dfrac{(P \Rightarrow Q)^z \qquad \dfrac{\dfrac{(\neg(\neg P \lor Q))^y \qquad \dfrac{\neg P^x}{\neg P \lor Q}}{\bot}}{P}{\scriptstyle x}\ \mathrm{RAA}}{Q}}{\neg P \lor Q}}{\bot}}{\neg P \lor Q}{\scriptstyle y}\ \mathrm{RAA}}{(P \Rightarrow Q) \Rightarrow (\neg P \lor Q)}{\scriptstyle z}$$

The other proofs are left as exercises. □

Propositions 1.3.4 and 1.5.1 show a property that is very specific to classical logic, namely, that the logical connectives $\Rightarrow, \land, \lor, \neg$ are not independent. For example, we have $P \land Q \equiv \neg(\neg P \lor \neg Q)$, which shows that $\land$ can be expressed in terms of $\lor$ and $\neg$. In intuitionistic logic, $\land$ and $\lor$ cannot be expressed in terms of each other via negation.

The fact that the logical connectives $\Rightarrow, \land, \lor, \neg$ are not independent in classical logic suggests the following question: Are there propositions, written in terms of $\Rightarrow$ only, that are provable classically but not provable intuitionistically?

The answer is yes! For instance, the proposition $((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$ (known as *Peirce's law*) is provable classically (do it) but it can be shown that it is not provable intuitionistically.

In addition to the proof by cases method and the proof by contradiction method, we also have the proof by contrapositive method valid in classical logic:

Proof by contrapositive rule:

$$\dfrac{\dfrac{\Gamma, \neg Q^x}{\neg P}}{P \Rightarrow Q}\; x$$

This rule says that in order to prove an implication $P \Rightarrow Q$ (from Γ), one may assume $\neg Q$ as proved, and then deduce that $\neg P$ is provable from Γ and $\neg Q$. This inference rule is valid in classical logic because we can construct the following proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, \neg Q^x}{\neg P} \qquad P^y}{\bot}}{Q}\; x \ \text{(by-contra)}}{P \Rightarrow Q}\; y$$

We will now give some explicit examples of proofs illustrating the proof principles that we just discussed.

Recall that the *set of integers* is the set

$$\mathbb{Z} = \{\cdots, -2, -1, 0, 1, 2, \cdots\}$$

and that the *set of natural numbers* is the set

$$\mathbb{N} = \{0, 1, 2, \cdots\}.$$

(Some authors exclude 0 from $\mathbb{N}$. We don't like this discrimination against zero.) An integer is *even* if it is divisible by 2, that is, if it can be written as $2k$, where $k \in \mathbb{Z}$. An integer is *odd* if it is not divisible by 2, that is, if it can be written as $2k + 1$, where $k \in \mathbb{Z}$. The following facts are essentially obvious:

(a) The sum of even integers is even.

(b) The sum of an even integer and of an odd integer is odd.

(c) The sum of two odd integers is even.

(d) The product of odd integers is odd.

(e) The product of an even integer with any integer is even.

Now, we prove the following fact using the proof by cases method.

Proposition 1.5.2 *Let a, b, c be odd integers. For any integers p and q, if p and q are not both even, then*

$$ap^2 + bpq + cq^2$$

is odd.

Proof. We consider the three cases:

1. p and q are odd. In this case as a, b and c are odd, by (d) all the products ap^2, bpq and cq^2 are odd. By (c), $ap^2 + bpq$ is even and by (b), $ap^2 + bpq + cq^2$ is odd.

2. p is even and q is odd. In this case, by (e), both ap^2 and bpq are even and by (d), cq^2 is odd. But then, by (a), $ap^2 + bpq$ is even and by (b), $ap^2 + bpq + cq^2$ is odd.

3. p is odd and q is even. This case is analogous to the previous case, except that p and q are interchanged. The reader should have no trouble filling in the details.

Since all three cases exhaust all possibilities for p and q not to be both even, the proof is complete by the $\vee$-elimination rule (applied twice). □

The set of rational numbers $\mathbb{Q}$ consists of all fractions p/q, where $p, q \in \mathbb{Z}$, with $q \neq 0$. We now use Proposition 1.5.2 and the proof by contradiction method to prove

Proposition 1.5.3 *Let a, b, c be odd integers. Then, the equation*

$$aX^2 + bX + c = 0$$

has no rational solution X.

Proof. We proceed by contradiction (by this, we mean that we use the proof by contradiction rule). So, assume that there is a rational solution $X = p/q$. We may assume that p and q have no common divisor, which implies that p and q are not both even. As $q \neq 0$, if $aX^2 + bX + c = 0$, then by multiplying by q^2, we get

$$ap^2 + bpq + cq^2 = 0.$$

However, as p and q are not both even and a, b, c are odd, we know from Proposition 1.5.2 that $ap^2 + bpq + cq^2$ is odd. This contradicts the fact that $p^2 + bpq + cq^2 = 0$ and thus, finishes the proof. □

As as example of the proof by contrapositive method, we prove that if an integer n^2 is even, then n must be even.

Observe that if an integer is not even then it is odd (and vice-versa). Thus, the contrapositive of our statement is: If n is odd, then n^2 is odd. But, to say that n is odd is to say that $n = 2k + 1$ and then, $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$, which shows that n^2 is odd.

A real number $a \in \mathbb{R}$ is said to be *irrational* if it cannot be expressed as a number in $\mathbb{Q}$ (a fraction). The reader should prove that $\sqrt{2}$ is irrational by adapting the arguments used in the two previous propositions.

Remark: Let us return briefly to the issue of constructivity in classical logic, in particular when it comes to disjunctions. Consider the question: are there two irrational real numbers a and b such that a^b is rational? Here is a way to prove that this indeed the case. Consider the number $\sqrt{2}^{\sqrt{2}}$. If this number is rational, then $a = \sqrt{2}$ and $b = \sqrt{2}$ is an answer to our question (since we already know that $\sqrt{2}$ is irrational). Now, observe that

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2}\times\sqrt{2}} = \sqrt{2}^2 = 2 \quad \text{is rational!}$$

Thus, if $\sqrt{2}^{\sqrt{2}}$ is irrational, then $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$ is an answer to our question. So, we proved that

($\sqrt{2}$ is irrational and $\sqrt{2}^{\sqrt{2}}$ is rational) or
($\sqrt{2}^{\sqrt{2}}$ and $\sqrt{2}$ are irrational and $(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}}$ is rational).

However, the above proof does not tell us whether $\sqrt{2}^{\sqrt{2}}$ is rational or not!

We see one of the shortcomings of classical reasoning: certain statements (in particular, disjunctive or existential) are provable but their proof does provide an explicit answer. It is in that sense that classical logic is not constructive.

Many more examples of non-constructive arguments in classical logic can be given.

1.6 Truth Values Semantics for Classical Logic Soundness and Completeness

So far, even though we have deliberately focused on proof theoy and ignored semantic issues, we feel that we can't postpone any longer a discussion of the truth values semantics for classical propositional logic.

We all learned early on that the logical connectives, $\Rightarrow$, $\wedge$, $\vee$ and $\neg$ can be interpreted as boolean functions, that is, functions whose arguments and whose values range over the set of *truth values*,

$$\mathbf{BOOL} = \{\mathbf{true}, \mathbf{false}\}.$$

These functions are given by the following *truth tables*:

P	Q	$P \Rightarrow Q$	$P \wedge Q$	$P \vee Q$	$\neg P$
true	**true**	**true**	**true**	**true**	**false**
true	**false**	**false**	**false**	**true**	**false**
false	**true**	**true**	**false**	**true**	**true**
false	**false**	**true**	**false**	**false**	**true**

Now, any proposition, P, built up over the set of atomic propositions, **PS**, (our propositional symbols) contains a finite set of propositional letters, say

$$\{P_1, \ldots, P_m\}.$$

If we assign some truth value (from **BOOL**) to each symbol, P_i, then we can "compute" the *truth value* of P under this assignment by using (recursively) the truth tables above. For example, the proposition $\mathbf{P}_1 \Rightarrow (\mathbf{P}_1 \Rightarrow \mathbf{P}_2)$, under the truth assignment,

$$\mathbf{P}_1 = \mathbf{true},\ \mathbf{P}_2 = \mathbf{false},$$

evaluates to **false**. However, under the truth assignment,

$$\mathbf{P}_1 = \mathbf{true},\ \mathbf{P}_2 = \mathbf{true},$$

our proposition evaluates to **true**.

If we now consider the proposition,

$$P = (\mathbf{P}_1 \Rightarrow (\mathbf{P}_2 \Rightarrow \mathbf{P}_1)),$$

then it is easy to see that P evaluates to **true** for all four possible truth assignments for $\mathbf{P}_1$ and $\mathbf{P}_2$.

Definition 1.6.1 We say that a proposition, P, is *satisfiable* iff it evalates to **true** for *some* truth assignment (taking values in **BOOL**) of the propositional symbols occurring in P and otherwise we say that it is *unsatisfiable*. A proposition, P, is *valid* (or a *tautology*) iff it evaluates to **true** for *all* truth assignments of the propositional symbols occurring in P.

The problem of deciding whether a proposition is satisfiable or not is called the *satisfiability problem* and is sometimes denoted by SAT. The problem of deciding whether a proposition is valid or not is called the *validity problem*. The satisfiability problem is a famous problem in computer science because of its complexity. Try it, solving it is not as easy as you think! In fact, the satisfiability problem turns out to be an *NP-complete* problem, a very important concept that you will learn about in CIS262. The validity problem is also important and it is related to SAT. Indeed, it is easy to see that a proposition, P, is valid iff $\neg P$ is unsatisfiable.

What's the relationship between validity and provability in the system $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp}$ (or $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp}$)?

Remarkably, in classical logic, validity and provability are equivalent!

In order to prove the above claim, we need to do two things:

(1) Prove that if a proposition, P, if provable in the system $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp}$ (or $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp}$), then it is valid. This is known as *soundness* or *consistency* (of the proof system).

(2) Prove that if a proposition, P, is valid, then it has a proof in the system $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp}$ (or $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp}$). This is known as the *completeness* (of the proof system).

In general, it is relatively easy to prove (1) but proving (2) can be quite complicated. In fact, some proof systems are *not* complete with respect to certain semantics. For instance, the proof system for intuitionistic logic, $\mathcal{N}_i^{\Rightarrow,\wedge,\vee,\perp}$ (or $\mathcal{NG}_i^{\Rightarrow,\wedge,\vee,\perp}$), is *not complete* with respect to truth values semantics! As an example, $((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$ (known as *Peirce's law*), is valid but it can be shown that it cannot be proved in intuitionistic logic.

In these notes, we will content ourselves with soundness.

Proposition 1.6.2 *(Soundness of $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp}$ and $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp}$) If a proposition, P, is provable in the system $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp}$ (or $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp}$), then it is valid (according to the truth values semantics).*

Sketch of Proof. It is enough to prove that if there is a deduction of a proposition, P, from a set of premises, Γ, then for every truth assignment for which all the propositions in Γ evaluate to **true**, then P evaluates to **true**. However, this is clear for the axioms and every inference rule preserves that property.

Now, if P is provable, a proof of P has an empty set of premises and so P evaluates to **true** for all truth assignments, which means that P is valid. $\square$

Theorem 1.6.3 *(Completeness of $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp}$ and $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp}$) If a proposition, P, is valid (according to the truth values semantics), then P is provable in the system $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp}$ (or $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp}$).*

Proofs of completeness for classical logic can be found in van Dalen [44] or Gallier [19] (but for a different proof system).

Soundness (Proposition 1.6.2) has a very useful consequence: In order to prove that a proposition, P, is *not provable*, it is enough to find a truth assignment for which P evaluates to **false**. We say that such a truth assignment is a *counter-example* for P (or that P can be *falsified*). For example, no propositional symbol, $\mathbf{P}_i$, is provable since it is falsified by the truth assignment $\mathbf{P}_i = \mathbf{false}$. Note that completeness amounts to the fact that every unprovable formula has a counter-example.

Remark: Truth values semantics is not the right kind of semantics for intuitionistic logic; it is too coarse. A more subtle kind of semantics is required. Among the various semantics for intuitionistic logic, one of the most natural is the notion of *Kripke model*. Then, again, soundness and completeness holds for intuitionistic proof systems (see Section 1.7 and van Dalen [44]).

1.7 Kripke Models for Intuitionistic Logic Soundness and Completeness

In this section, we briefly describe the semantics of intuitionistic propositional logic in terms of Kripke models. This section has been included to quench the thirst of those readers who can't wait to see what kind of decent semantics can be given for intuitionistic propositional logic and it can be safely omitted. We recommend reviewing the material of Section 4.1 before reading this section.

In classical truth values semantics based on $\mathbf{BOOL} = \{\mathbf{true}, \mathbf{false}\}$, we might say that truth is absolute. The idea of Kripke semantics is that there is a set of worlds, W, together with a partial ordering, $\leq$, on W, and that truth depends in which world we are. Furthermore, as we "go up" from a world u to a world v with $u \leq v$, truth "can only increase", that is, whatever is true in world u remains true in world v. Also, the truth of some propositions, such as $P \Rightarrow Q$ or $\neg P$, depends on "future worlds". With this type of semantics, which is no longer absolute, we can capture exactly the essence of intuitionistic logic. We now make these ideas precise.

Definition 1.7.1 A *Kripke model* for intuitionistic propositional logic is a pair, $\mathcal{K} = (W, \varphi)$, where W is a partially ordered (nonempty) set called a *set of worlds* and φ is a function, $\varphi\colon W \to \mathbf{BOOL}^{\mathbf{PS}}$, such that for every $u \in W$, the function $\varphi(u)\colon \mathbf{PS} \to \mathbf{BOOL}$ is an assignment of truth values to the propositional symbols in $\mathbf{PS}$ satisfying the following property: For all $u, v \in W$, for all $\mathbf{P}_i \in \mathbf{PS}$,

$$\text{if } u \leq v \text{ and } \varphi(u)(\mathbf{P}_i) = \mathbf{true}, \quad \text{then } \varphi(v)(\mathbf{P}_i) = \mathbf{true}.$$

As we said in our informal comments, truth can't decrease when we move from a world u to a world v with $u \leq v$ but truth can increase, that is it is possible that $\varphi(u)(\mathbf{P}_i) = \mathbf{false}$ and yet, $\varphi(v)(\mathbf{P}_i) = \mathbf{true}$. We use Kripke models to define the semantics of propositions as follows:

Definition 1.7.2 Given a Kripke model, $\mathcal{K} = (W, \varphi)$, for every $u \in W$ and for every proposition, P, we say that P *is satisfied by* $\mathcal{K}$ *at* u and we write $\varphi(u)(P) = \mathbf{true}$ iff

(a) $\varphi(u)(\mathbf{P}_i) = \mathbf{true}$, if $P = \mathbf{P}_i \in \mathbf{PS}$;

(b) $\varphi(u)(Q) = \mathbf{true}$ and $\varphi(u)(R) = \mathbf{true}$, if $P = Q \wedge R$;

(c) $\varphi(u)(Q) = \mathbf{true}$ or $\varphi(u)(R) = \mathbf{true}$, if $P = Q \vee R$;

(d) For all v such that $u \leq v$, if $\varphi(v)(Q) = \mathbf{true}$, then $\varphi(v)(R) = \mathbf{true}$, if $P = Q \Rightarrow R$;

(e) For all v such that $u \leq v$, $\varphi(v)(Q) = \mathbf{false}$, if $P = \neg Q$.

(f) $\varphi(u)(\perp) = \mathbf{false}$, that is, $\perp$ is not satisfied by $\mathcal{K}$ at u (for any $\mathcal{K}$ and any u).

We say that P is *valid in* $\mathcal{K}$ (or that $\mathcal{K}$ is a *model* of P) iff P is satisfied by $\mathcal{K} = (W, \varphi)$ at u for all $u \in W$ and we say that P is *intuitionistically valid* iff P is valid in every Kripke model, $\mathcal{K}$.

When P is satisfied by $\mathcal{K}$ at u we also say that P *is true at* u *in* $\mathcal{K}$. Note that the truth at $u \in W$ of a proposition of the form $Q \Rightarrow R$ or $\neg Q$ depends on the truth of Q and R at all "future worlds", $v \in W$, with $u \leq v$. Observe that classical truth values semantics corresponds to the special case where W consists of a single element (a single world).

If $W = \{0, 1\}$ ordered so that $0 \leq 1$ and if φ is given by

$$\begin{aligned} \varphi(0)(\mathbf{P}_i) &= \mathbf{false} \\ \varphi(1)(\mathbf{P}_i) &= \mathbf{true}, \end{aligned}$$

then $\mathcal{K} = (W, \varphi)$ is a Kripke structure. The reader should check that the proposition $P = (\mathbf{P}_i \vee \neg \mathbf{P}_i)$ has the value **false** at 0 because $\varphi(0)(\mathbf{P}_i) = \mathbf{false}$ but $\varphi(1)(\mathbf{P}_i) = \mathbf{true}$, so clause (e) fails for $\neg \mathbf{P}_i$ at $u = 0$. Therefore, $P = (\mathbf{P}_i \vee \neg \mathbf{P}_i)$ is not valid in $\mathcal{K}$ and thus, it is not intuitionistically valid. We escaped the classical truth values semantics by using a universe with two worlds. The reader should also check that

$$\begin{aligned} \varphi(u)(\neg\neg P) = \mathbf{true} \quad \text{iff} \quad & \text{for all } v \text{ such that } u \leq v \\ & \text{there some } w \text{ with } v \leq w \text{ so that } \varphi(w)(P) = \mathbf{true}. \end{aligned}$$

This shows that in Krikpe semantics, $\neg\neg P$ is weaker than P, in the sense that $\varphi(u)(\neg\neg P) = \mathbf{true}$ does not necessarily imply that $\varphi(u)(P) = \mathbf{true}$.

As we said in the previous section, Kripke semantics is a perfect fit to intuitionistic provability in the sense that soundness and completeness hold.

Proposition 1.7.3 *(Soundness of $\mathcal{N}_i^{\Rightarrow,\wedge,\vee,\perp}$ and $\mathcal{NG}_i^{\Rightarrow,\wedge,\vee,\perp}$) If a proposition, P, is provable in the system $\mathcal{N}_i^{\Rightarrow,\wedge,\vee,\perp}$ (or $\mathcal{NG}_i^{\Rightarrow,\wedge,\vee,\perp}$), then it is valid in every Kripke model, that is, it is intuitionistically valid.*

Proposition 1.7.3 is not hard to prove. We consider any deduction of a proposition, P, from a set of premises, Γ, and we prove that for every Kripke model, $\mathcal{K} = (W, \varphi)$, for every $u \in W$, if every premise in Γ is satisfied by $\mathcal{K}$ at u, then P is also satisfied by $\mathcal{K}$ at u. This is obvious for the axioms and it is easy to see that the inference rules preserve this property.

Completeness also holds, but it is harder to prove (see van Dalen [44]).

Theorem 1.7.4 *(Completeness of $\mathcal{N}_i^{\Rightarrow,\wedge,\vee,\perp}$ and $\mathcal{NG}_i^{\Rightarrow,\wedge,\vee,\perp}$) If a proposition, P, is intuitionistically valid, then P is provable in the system $\mathcal{N}_i^{\Rightarrow,\wedge,\vee,\perp}$ (or $\mathcal{NG}_i^{\Rightarrow,\wedge,\vee,\perp}$).*

Another proof of completeness for a different proof system for propositional intuitionistic logic (a Gentzen-sequent calculus equivalent to $\mathcal{NG}_i^{\Rightarrow,\wedge,\vee,\perp}$) is given in Takeuti [42]. We find this proof more instructive that van Dalen's proof. This proof also shows that if a proposition, P, is not intuitionistically provable, then there is a Kripke model, $\mathcal{K}$, where W is a *finite tree*, in which P is not valid. Such a Kripke model is called a *counter-example* for P.

We now add quantifiers to our language and give the corresponding inference rules.

1.8 Adding Quantifiers; The Proof Systems $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\forall,\exists,\perp}$, $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\forall,\exists,\perp}$

As we mentioned in Section 1.1, atomic propositions may contain variables. The intention is that such variables correspond to arbitrary objects. An example is

$$\text{human}(x) \Rightarrow \text{needs-to-drink}(x).$$

Now, in mathematics, we usually prove universal statements, that is statement that hold for all possible "objects", or existential statement, that is, statement asserting the existence of some object satisfying a given property. As we saw earlier, we assert that every human needs to drink by writing the proposition

$$\forall x(\text{human}(x) \Rightarrow \text{needs-to-drink}(x)).$$

Observe that once the quantifier $\forall$ (pronounced "for all" or "for every") is applied to the variable x, the variable x becomes a place-holder and replacing x by y or any other variable does not change anything. What matters is the locations to which the outer x points to in the inner proposition. We say that x is a *bound variable* (sometimes a "dummy variable").

If we want to assert that some human needs to drink we write

$$\exists x(\text{human}(x) \Rightarrow \text{needs-to-drink}(x));$$

Again, once the quantifier $\exists$ (pronounced "there exists") is applied to the variable x, the variable x becomes a place-holder. However, the intended meaning of the second proposition is very different and weaker than the first. It only asserts the existence of some object satisfying the statement

$$\text{human}(x) \Rightarrow \text{needs-to-drink}(x).$$

Statements may contain variables that are not bound by quantifiers. For example, in

$$\forall y\, \text{parent}(x, y)$$

the variable y is bound but the variable x is not. Here, the intended meaning of parent(x, y) is that x is a parent of y. Variables that are not bound are called *free*. The proposition

$$\forall y \exists x\, \text{parent}(x, y),$$

which contains only bound variables in meant to assert that every y has some parent x. Typically, in mathematics, we only prove statements without free variables. However, statements with free variables may occur during intermediate stages of a proof.

The intuitive meaning of the statement $\forall x P$ is that P holds for all possible objects x and the intuitive meaning of the statement $\exists x P$ is that P holds for some object x. Thus, we see that it would be useful to use symbols to denote various objects. For example, if we want to assert some facts about the "parent" predicate, we may want to introduce some *constant symbols* (for short, constants) such as "Jean", "Mia", *etc.* and write

$$\mathrm{parent}(\mathrm{Jean}, \mathrm{Mia})$$

to assert that Jean is a parent of Mia. Often, we also have to use *function symbols* (or *operators, constructors*), for instance, to write statement about numbers: $+$, $*$, *etc.* Using constant symbols, function symbols and variables, we can form *terms*, such as

$$(x^2 + 1)(3 * y + 2).$$

In addition to function symbols, we also use *predicate symbols*, which are names for atomic properties. We have already seen several examples of predicate symbols: "human", "parent". So, in general, when we try to prove properties of certain classes of objects (people, numbers, strings, graphs, *etc.*), we assume that we have a certain *alphabet* consisting of constant symbols, function symbols and predicate symbols. Using these symbols and an infinite supply of variables (assumed distinct from the variables which we use to label premises) we can form *terms and predicate terms*. We say that we have a *(logical) language*. Using this language, we can write compound statements.

Let us be a little more precise. In a *first-order language*, **L**, in addition to the logical connectives, $\Rightarrow, \wedge, \vee, \neg, \perp, \forall$ and $\exists$, we have a set, **L**, of *nonlogical symbols* consisting of

(i) A set **CS** of constant symbols, $c_1, c_2, \ldots,$.

(ii) A set **FS** of function symbols, $f_1, f_2, \ldots,$. Each function symbol, f, has a *rank*, $n_f \geq 1$, which is the number of arguments of f.

(iii) A set **PS** of predicate symbols, $P_1, P_2, \ldots,$. Each predicate symbol, P, has a *rank*, $n_P \geq 0$, which is the number of arguments of P. Predicate symbols of rank 0 are propositional letters, as in earlier sections.

(iv) The equality predicate, $=$, is added to our language when we want to deal with equations.

(v) First-order variables, $t_1, t_2, \ldots$, used to form quantified formulae.

The difference between function symbols and predicate symbols is that function symbols are interpreted as functions defined on a structure (for example, addition, $+$, on $\mathbb{N}$), whereas predicate symbols are interpreted as properties of objects, that is, they take the value **true** or **false**. An example is the language of *Peano arithmetic*, $\mathbf{L} = \{0, S, +, *, =\}$. Here, the intended structure is $\mathbb{N}$, 0 is of course zero, S is interpreted as the function $S(n) = n + 1$, the symbol $+$ is addition, $*$ is multiplication and $=$ is equality.

Using a first-order language, $\mathbf{L}$, we can form terms, predicate terms and formulae. The *terms over* $\mathbf{L}$ are the following expressions:

(i) Every variable, t, is a term;

(ii) Every constant symbol, $c \in \mathbf{CS}$, is a term;

(iii) If $f \in \mathbf{FS}$ is a function symbol taking n arguments and $\tau_1, \ldots, \tau_n$ are terms already constructed, then $f(\tau_1, \ldots, \tau_n)$ is a term.

The *predicate terms over* $\mathbf{L}$ are the following expressions:

(i) If $P \in \mathbf{PS}$ is a predicate symbol taking n arguments and $\tau_1, \ldots, \tau_n$ are terms already constructed, then $P(\tau_1, \ldots, \tau_n)$ is a predicate term. When $n = 0$, the predicate symbol, P, is a predicate term called a propositional letter.

(ii) When we allow the equality predicate, for any two terms τ_1 and τ_2, the expression $\tau_1 = \tau_2$ is a predicate term. It is usually called an *equation*.

The *(first-order) formulae over* $\mathbf{L}$ are the following expressions:

(i) Every predicate term, $P(\tau_1, \ldots, \tau_n)$, is an atomic formula. This includes all propositional letters. We also view $\perp$ (and sometimes $\top$) as an atomic formula.

(ii) When we allow the equality predicate, every equation, $\tau_1 = \tau_2$, is an atomic formula.

(iii) If P and Q are formulae already constructed, then $P \Rightarrow Q$, $P \wedge Q$, $P \vee Q$, $\neg P$ are compound formulae. We treat $P \equiv Q$ as an abbreviation for $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$, as before.

(iv) If P is a formula already constructed and t is any variable, then $\forall t P$ and $\exists t P$ are compound formulae.

All this can be made very precise but this is quite tedious. Our primary goal is to explain the basic rules of logic and not to teach a full-fledged logic course. We hope that our intuitive explanations will suffice and we now come to the heart of the matter, the inference rules for the quantifiers. Once again, for a complete treatment, readers are referred to Gallier [19] van Dalen [44] or Huth and Ryan [31].

Unlike the rules for $\Rightarrow, \vee, \wedge$ and $\perp$, which are rather straightforward, the rules for quantifiers are more subtle due the presence of variables (occurring in terms and predicates). We have to be careful to forbid inferences that would yield "wrong" results and for this we have to be very precise about the way we use free variables. More specifically, we have to exercise care when we make *substitutions* of terms for variables in propositions. For example, say we have the predicate "odd", intended to express that a number is odd. Now, we can substitute the term $(2y+1)^2$ for x in $\mathrm{odd}(x)$ and obtain

$$\mathrm{odd}((2y+1)^2).$$

More generally, if $P(t_1, t_2, \ldots, t_n)$ is a statement containing the free variables $t_1, \ldots, t_n$ and if $\tau_1, \ldots, \tau_n$ are terms, we can form the new statement

$$P[\tau_1/t_1, \ldots, \tau_n/t_n]$$

obtained by substituting the term τ_i for all free occurrences of the variable t_i, for $i = 1, \ldots, n$. By the way, we denote terms by the greek letter τ because we use the letter t for a variable and using t for both variables and terms would be confusing; sorry!

However, if $P(t_1, t_2, \ldots, t_n)$ contains quantifiers, some bad things can happen, namely, some of the variables occurring in some term τ_i may become quantified when τ_i is substituted for t_i. For example, consider

$$\forall x \exists y\, P(x, y, z)$$

which contains the free variable z and substitute the term $x + y$ for z: we get

$$\forall x \exists y\, P(x, y, x + y).$$

We see that the variables x and y occurring in the term $x + y$ become bound variables after substitution. We say that there is a "capture of variables".

This is not what we intended to happen! To fix this problem, we recall that bound variables are really place holders, so they can be renamed without changing anything. Therefore, we can rename the bound variables x and y in $\forall x \exists y\, P(x, y, z)$ to u and v, getting the statement $\forall u \exists v\, P(u, v, z)$ and now, the result of the substitution is

$$\forall u \exists v\, P(u, v, x + y).$$

Again, all this needs to be explained very carefuly but this can be done!

Finally, here are the inference rules for the quantifiers, first stated in a natural deduction style and then in sequent style. It is assumed that we use two disjoint sets of variables for labeling premises $(x, y, \cdots)$ and free variables $(t, u, v, \cdots)$. As we will see, the $\forall$-introduction rule and the $\exists$-elimination rule involve a crucial restriction on the occurrences of certain variables. Remember, *variables are terms*!

Definition 1.8.1 The *inference rules for the quantifiers* are

$\forall$-*introduction*:

$$\cfrac{\cfrac{\Gamma}{P[u/t]}}{\forall t P}$$

Here, u must be a variable that does not occur free in any of the propositions in Γ or in $\forall t P$. The notation $P[u/t]$ stands for the result of substituting u for all free occurrences of t in P. Recall that Γ denotes the set of premises of the above deduction tree so if this tree only has two nodes, then $P[u/t] \in \Gamma$ and t should not occur in P.

$\forall$-*elimination*:

$$\cfrac{\cfrac{\Gamma}{\forall t P}}{P[\tau/t]}$$

Here τ is an arbitrary term and it is assumed that bound variables in P have been renamed so that none of the variables in τ are captured after substitution.

$\exists$-*introduction*:

$$\cfrac{\cfrac{\Gamma}{P[\tau/t]}}{\exists t P}$$

As in $\forall$-elimination, τ is an arbitrary term and the same proviso on bound variables in P applies.

$\exists$-*elimination*:

$$\cfrac{\cfrac{\Gamma}{\exists t P} \qquad \cfrac{\Delta, P[u/t]^x}{C}}{C}\; x$$

Here, u must be a variable that does not occur free in any of the propositions in Δ, $\exists t P$, or C, and all premises $P[u/t]$ labeled x are discharged.

In the above rules, Γ or Δ may be empty, P, C denote arbitrary propositions constructed from a first-order language, $\mathbf{L}$, and t is *any* variable. The system of *first-order classical logic*, $\mathcal{N}_c^{\Rightarrow,\vee,\wedge,\perp,\forall,\exists}$ is obtained by adding the above rules to the system of propositional classical logic $\mathcal{N}_c^{\Rightarrow,\vee,\wedge,\perp}$. The system of *first-order intuitionistic logic*, $\mathcal{N}_i^{\Rightarrow,\vee,\wedge,\perp,\forall,\exists}$ is obtained by adding the above rules to the system of propositional intuitionistic logic $\mathcal{N}_i^{\Rightarrow,\vee,\wedge,\perp}$.

Using sequents, the quantifier rules in first-order logic are expressed as follows:

Definition 1.8.2 The *inference rules for the quantifiers in Gentzen-sequent style* are

$$\frac{\Gamma \to P[u/t]}{\Gamma \to \forall t P}\ (\forall\textit{-intro}) \qquad \frac{\Gamma \to \forall t P}{\Gamma \to P[\tau/t]}\ (\forall\textit{-elim})$$

where in ($\forall$-*intro*), u does not occur free in Γ or $\forall t P$;

$$\frac{\Gamma \to P[\tau/t]}{\Gamma \to \exists t P}\ (\exists\textit{-intro}) \qquad \frac{\Gamma \to \exists t P \quad z\colon P[u/t], \Gamma \to C}{\Gamma \to C}\ (\exists\textit{-elim})$$

where in ($\exists$-*elim*), u does not occur free in Γ, $\exists t P$, or C. Again, t is *any* variable.

The variable u is called the *eigenvariable* of the inference. The systems $\mathcal{NG}_c^{\Rightarrow,\vee,\wedge,\perp,\forall,\exists}$ and $\mathcal{NG}_i^{\Rightarrow,\vee,\wedge,\perp,\forall,\exists}$ are defined from the systems $\mathcal{NG}_c^{\Rightarrow,\vee,\wedge,\perp}$ and $\mathcal{NG}_i^{\Rightarrow,\vee,\wedge,\perp}$, respectively, by adding the above rules.

When we say that a proposition, P, is *provable from* Γ, we mean that we can construct a proof tree whose conclusion is P and whose set of premises is Γ, in one of the systems $\mathcal{N}_c^{\Rightarrow,\wedge,\vee,\perp,\forall,\exists}$ or $\mathcal{NG}_c^{\Rightarrow,\wedge,\vee,\perp,\forall,\exists}$. Therefore, as in propositional logic, when we use the word "provable" unqualified, we mean provable in *classical logic*. Otherwise, we say *intuitionistically provable* .

A first look at the above rules shows that universal formulae, $\forall t P$, behave somewhat like infinite conjunctions and that existential formulae, $\exists t P$, behave somewhat like infinite disjunctions.

The $\forall$-introduction rule looks a little strange but the idea behind it is actually very simple: Since u is totally unconstrained, if $P[u/t]$ is provable (from Γ), then intuitively $P[u/t]$ holds of any arbitrary object, and so, the statement $\forall t P$ should also be provable (from Γ). Note that the tree

$$\frac{P[u/t]}{\forall t P}$$

is generally an illegal deduction because it has the single premise, $P[u/t]$, and u occurs in $P[u/t]$ unless t does not occur in P.

The meaning of the $\forall$-elimination is that if $\forall t P$ is provable (from Γ), then P holds for all objects and so, in particular for the object denoted by the term τ, i.e., $P[\tau/t]$ should be provable (from Γ).

The $\exists$-introduction rule is dual to the $\forall$-elimination rule. If $P[\tau/t]$ is provable (from Γ), this means that the object denoted by τ satisfies P, so $\exists t P$ should be provable (this latter formula asserts the existence of some object satisfying P, and τ is such an object).

The $\exists$-elimination rule is reminiscent of the $\vee$-elimination rule and is a little more tricky. It goes as follows: Suppose that we proved $\exists t P$ (from Γ). Moreover, suppose that for every possible case, $P[u/t]$, we were able to prove C (from Γ). Then, as we have "exhausted" all possible cases and as we know from the provability of $\exists tP$ that some case must hold, we can conclude that C is provable (from Γ) without using $P[u/t]$ as a premise.

Like the $\lor$-elimination rule, the $\exists$-elimination rule is not very constructive. It allows making a conclusion (C) by considering alternatives without knowing which one actually occurs.

Remark: Anagolously to disjunction, in (first-order) intuitionistic logic, if an existential statement $\exists tP$ is provable, then from any proof of $\exists tP$, some term, τ, can be extracted so that $P[\tau/t]$ is provable. Such a term, τ, is called a *witness*. The witness property is not easy to prove. It follows from the fact that intuitionistic proofs have a normal form (see Section 1.9). However, no such property holds in classical logic (for instance, see the a^b rational with a, b irrational example revisited below).

Here is an example of a proof in the system $\mathcal{N}_c^{\Rightarrow,\lor,\land,\perp,\forall,\exists}$ (actually, in $\mathcal{N}_i^{\Rightarrow,\lor,\land,\perp,\forall,\exists}$) of the formula $\forall t(P \land Q) \Rightarrow \forall tP \land \forall tQ$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall t(P \land Q)^x}{P[u/t] \land Q[u/t]}}{P[u/t]}}{\forall tP} \qquad \dfrac{\dfrac{\dfrac{\forall t(P \land Q)^x}{P[u/t] \land Q[u/t]}}{Q[u/t]}}{\forall tQ}}{\forall tP \land \forall tQ}}{\forall t(P \land Q) \Rightarrow \forall tP \land \forall tQ}\; x$$

In the above proof, u is a new variable, i.e., a variable that does not occur free in P or Q. We also have used some basic properties of substitutions such as:

$$\begin{aligned}
(P \land Q)[\tau/t] &= P[\tau/t] \land Q[\tau/t] \\
(P \lor Q)[\tau/t] &= P[\tau/t] \lor Q[\tau/t] \\
(P \Rightarrow Q)[\tau/t] &= P[\tau/t] \Rightarrow Q[\tau/t] \\
(\neg P)[\tau/t] &= \neg P[\tau/t] \\
(\forall sP)[\tau/t] &= \forall sP[\tau/t] \\
(\exists sP)[\tau/t] &= \exists sP[\tau/t],
\end{aligned}$$

for any term, τ, such that no variable in τ is captured during the substitution (in particular, in the last two cases, the variable s does not occur in τ).

The reader should show that $\forall tP \land \forall tQ \Rightarrow \forall t(P \land Q)$ is also provable in $\mathcal{N}_i^{\Rightarrow,\lor,\land,\perp,\forall,\exists}$. However, in general, one can't just replace $\forall$ by $\exists$ (or $\land$ by $\lor$) and still obtain provable statements. For example, $\exists tP \land \exists tQ \Rightarrow \exists t(P \land Q)$ is not provable at all!

Here are some useful equivalences involving quantifiers. The first two are analogous to the de Morgan laws for $\land$ and $\lor$.

Proposition 1.8.3 *The following equivalences are provable in classical first-order logic:*

$$\begin{aligned} \neg\forall t P &\equiv \exists t \neg P \\ \neg\exists t P &\equiv \forall t \neg P \\ \forall t(P \wedge Q) &\equiv \forall t P \wedge \forall t Q \\ \exists t(P \vee Q) &\equiv \exists t P \vee \exists t Q. \end{aligned}$$

In fact, the last three and $\exists t \neg P \Rightarrow \neg\forall t P$ are provable intuitionistically. Moreover, the propositions $\exists t(P \wedge Q) \Rightarrow \exists t P \wedge \exists t Q$ and $\forall t P \vee \forall t Q \Rightarrow \forall t(P \vee Q)$ are provable in intuitionistic first-order logic (and thus, also in classical first-order logic).

Proof. Left as an exercise to the reader. □

Remark: We can illustrate, again, the fact that classical logic allows for non-constructive proofs by reexamining the example at the end of Section 1.3. There, we proved that if $\sqrt{2}^{\sqrt{2}}$ is rational, then $a = \sqrt{2}$ and $b = \sqrt{2}$ are both irrational numbers such that a^b is rational and if $\sqrt{2}^{\sqrt{2}}$ is irrational then $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$ are both irrational numbers such that a^b is rational. By $\exists$-introduction, we deduce that if $\sqrt{2}^{\sqrt{2}}$ is rational then there exist some irrational numbers a, b so that a^b is rational and if $\sqrt{2}^{\sqrt{2}}$ is irrational then there exist some irrational numbers a, b so that a^b is rational. In classical logic, as $P \vee \neg P$ is provable, by $\vee$-elimination, we just proved that there exist some irrational numbers a and b so that a^b is rational.

However, this argument does not give us explicitly numbers a and b with the required properties! It only tells us that such numbers must exist. Now, it turns out that $\sqrt{2}^{\sqrt{2}}$ is indeed irrational (this follows from the Gel'fond-Schneider Theorem, a hard theorem in number theory). Furthermore, there are also simpler explicit solutions such as $a = \sqrt{2}$ and $b = \log_2 9$, as the reader should check!

We conclude this section by giving an example of a "wrong proof". Here is an example in which the $\forall$-introduction rule is applied illegally, and thus, yields a statement which is actually false (not provable). In the incorrect "proof" below, P is an atomic predicate symbol taking two arguments (for example, "parent") and 0 is a constant denoting zero:

$$\cfrac{\cfrac{\cfrac{\cfrac{P(t,0)^x}{\forall t P(t,0)}\quad \text{illegal step!}}{P(t,0) \Rightarrow \forall t P(t,0)}\ x}{\forall t(P(t,0) \Rightarrow \forall t P(t,0))}}{P(0,0) \Rightarrow \forall t P(t,0)}$$

The problem is that the variable t occurs free in the premise $P[t/t, 0] = P(t, 0)$ and therefore, the application of the $\forall$-introduction rule in the first step is illegal. However, note that this premise is discharged in the second step and so, the application of the $\forall$-introduction rule in the third step is legal. The (false) conclusion of this faulty proof is that $P(0,0) \Rightarrow \forall t P(t,0)$ is provable. Indeed, there are plenty of properties such that the fact that the single instance, $P(0,0)$, holds does not imply that $P(t,0)$ holds for all t.

Remark: The above example shows why it is desirable to have premises that are universally quantified. A premise of the form $\forall t P$ can be instantiated to $P[u/t]$, using $\forall$-elimination, where u is a brand new variable. Later on, it may be possible to use $\forall$-introduction without running into trouble with free occurrences of u in the premises. But we still have to be very careful when we use $\forall$-introduction or $\exists$-elimination.

Before concluding this section, let us give a few more examples of proofs using the rules for the quantifiers. First, let us prove that

$$\forall t P \equiv \forall u P[u/t],$$

where u is any variable not free in $\forall t P$ and such that u is not captured during the substitution. This rule allows us to rename bound variables (under very mild conditions). We have the proofs

$$\dfrac{\dfrac{\dfrac{(\forall t P)^{\alpha}}{P[u/t]}}{\forall u P[u/t]}}{\forall t P \Rightarrow \forall u P[u/t]}\;\alpha$$

and

$$\dfrac{\dfrac{\dfrac{(\forall u P[u/t])^{\alpha}}{P[u/t]}}{\forall t P}}{\forall u P[u/t] \Rightarrow \forall t P}\;\alpha$$

Now, we give a proof (intuitionistic) of

$$\exists t(P \Rightarrow Q) \Rightarrow (\forall t P \Rightarrow Q),$$

where t does not occur (free or bound) in Q.

$$\dfrac{\dfrac{\dfrac{(\exists t(P \Rightarrow Q))^z \qquad \dfrac{(P[u/t] \Rightarrow Q)^x \qquad \dfrac{(\forall t P)^y}{P[u/t]}}{Q}}{Q}\;x}{\forall t P \Rightarrow Q}\;y}{\exists t(P \Rightarrow Q) \Rightarrow (\forall t P \Rightarrow Q)}\;z$$

In the above proof, u is a new variable that does not occur in Q, $\forall t P$, or $\exists t(P \Rightarrow Q)$. Since t does not occur in Q, we have

$$(P \Rightarrow Q)[u/t] = P[u/t] \Rightarrow Q.$$

The converse requires (RAA) and is a bit more complicated. To conclude, we give a proof (intuitionistic) of

$$(\forall t P \lor Q) \Rightarrow \forall t(P \lor Q),$$

where t does not occur (free or bound) in Q.

$$\dfrac{\dfrac{(\forall t P \lor Q)^z \qquad \dfrac{\dfrac{\dfrac{(\forall t P)^x}{P[u/t]}}{P[u/t] \lor Q}}{\forall t(P \lor Q)} \qquad \dfrac{\dfrac{Q^y}{P[u/t] \lor Q}}{\forall t(P \lor Q)}}{\forall t(P \lor Q)}\;x,y}{(\forall t P \lor Q) \Rightarrow \forall t(P \lor Q)}\;z$$

In the above proof, u is a new variable that does not occur in $\forall t P$ or Q. Since t does not occur in Q, we have

$$(P \lor Q)[u/t] = P[u/t] \lor Q.$$

The converse requires (RAA).

Obviously, every first-order formula that is provable intuitionistically is also provable classically and we know that there are formulae that are provable classically but *not* provable intuitionistically. Therefore, it appears that classical logic is more general than intuitionistic logic. However, this not not quite so because there is a way of interpreting classical logic into intuitionistic logic. To be more precise, every classical formula, A, can be translated into a formula, A^*, where A^* is classically equivalent to A and A is provable classically iff A^* is provable intuitionistically. Various translations are known, all based on a “trick” involving double-negation (This is because $\neg\neg\neg A$ and $\neg A$ are intuitionistically equivalent).

Translations were given Kolmogorov (1925), Gödel (1933) and Gentzen (1933). For example, Gödel used the following translation:

$$\begin{aligned}
A^* &= \neg\neg A, \quad \text{if } A \text{ is atomic,}\\
(\neg A)^* &= \neg A^*,\\
(A \wedge B)^* &= (A^* \wedge B^*),\\
(A \Rightarrow B)^* &= \neg(A^* \wedge \neg B^*),\\
(A \vee B)^* &= \neg(\neg A^* \wedge \neg B^*),\\
(\forall x A)^* &= \forall x A^*,\\
(\exists x A)^* &= \neg\forall x \neg A^*.
\end{aligned}$$

Actually, if we restrict our attention to propositions (that is, formulae without quantifiers), a theorem of Glivenko (1929) states that if a proposition, A, is provable classically, then $\neg\neg A$ is provable intuitionistically. In view of these results, the proponents of intuitionistic logic claim that classical logic is really a special case of intuitionistic logic! However, the above translations have some undesirable properties, as noticed by Girard. For more details on all this, see Gallier [17].

Several times in this Chapter, we have claimed that certain propositions are not provable in some logical system. What kind of reasoning do we use to validate such claims? In the next section, we briefly address this question as well as related ones.

1.9 Decision Procedures, Proof Normalization, Counter-Examples, Theories, etc.

In the previous sections, we saw how the rules of mathematical reasoning can be formalized in various natural deduction systems and we defined a precise notion of proof. We observed that finding a proof for a given proposition was not a simple matter, nor was it to acertain that a proposition is unprovable. Thus, it is natural to ask the following question:

The Decision Problem: Is there a general procedure which takes any arbitrary proposition, P, as input, always terminates in a finite number of steps, and tells us whether P is provable or not.

Clearly, it would be very nice if such a procedure existed, especially if it also produced a proof of P when P is provable.

Unfortunately, for rich enough languages, such as first-order logic, it is impossible to find such a procedure. This deep result known as the *undecidability of the decision problem* or *Church's Theorem* was proved by A. Church in 1936 (Actually, Church proved the undecidability of the validity problem but, by Gödel's completeness Theorem, validity and provability are equivalent).

Proving Church's Theorem is hard and a lot of work. One needs to develop a good deal of what is called the *theory of computation.* This involves defining models of computation such as *Turing machines* and proving other deeps results such as the *undecidability of the halting problem* and the *undecidability of the Post Correspondence Problem*, among other things. Some of this material is covered in CIS262, so be patient and your curiosity will be satisfied!

So, our hopes to find a "universal theorem prover" are crushed. However, if we restrict ourselves to propositional logic, classical or intuitionistic, it turns out that procedures solving the decision problem do exist and they even produce a proof of the input proposition when that proposition is provable.

Unfortunately, proving that such procedures exist and are correct in the propositional case is rather difficult, especially for intuitionistic logic. The difficulties have a lot to do with our choice of a natural deduction system. Indeed, even for the system $\mathcal{N}_m^{\Rightarrow}$ (or $\mathcal{NG}_m^{\Rightarrow}$), provable propositions may have infinitely many proofs. This makes the search process impossible; when do we know how to stop, especially if a proposition is not provable! The problem is that proofs may contain redundancies (Gentzen said "detours"). A typical example of redundancy is an elimination immediately follows an introduction, as in the following example in which $\mathcal{D}_1$ denotes a deduction with conclusion $\Gamma, x\colon A \to B$ and $\mathcal{D}_2$ denotes a deduction with conclusion $\Gamma \to A$.

$$\cfrac{\cfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma, x\colon A \to B\end{array}}{\Gamma \to A \Rightarrow B} \qquad\qquad \begin{array}{c}\mathcal{D}_2\\ \Gamma \to A\end{array}}{\Gamma \to B}$$

Intuitively, it should be possible to construct a deduction for $\Gamma \to B$ from the two deductions $\mathcal{D}_1$ and $\mathcal{D}_2$ without using at all the hypothesis $x\colon A$. This is indeed the case. If we look closely at the deduction $\mathcal{D}_1$, from the shape of the inference rules, assumptions are never created, and the leaves must be labeled with expressions of the form $\Gamma', \Delta, x\colon A, y\colon C \to C$ or $\Gamma, \Delta, x\colon A \to A$, where $y \neq x$ and either $\Gamma = \Gamma'$ or $\Gamma = \Gamma', y\colon C$. We can form a new deduction for $\Gamma \to B$ as follows: in $\mathcal{D}_1$, wherever a leaf of the form $\Gamma, \Delta, x\colon A \to A$ occurs, replace it by the deduction obtained from $\mathcal{D}_2$ by adding Δ to the premise of each sequent in $\mathcal{D}_2$. Actually, one should be careful to first make a fresh copy of $\mathcal{D}_2$ by renaming all the variables so that clashes with variables in $\mathcal{D}_1$ are avoided. Finally, delete the assumption $x\colon A$ from the premise of every sequent in the resulting proof. The resulting deduction is obtained by a kind of substitution and may be denoted as $\mathcal{D}_1[\mathcal{D}_2/x]$, with some minor abuse of notation. Note that the assumptions $x\colon A$ occurring in the leaves of the form $\Gamma', \Delta, x\colon A, y\colon C \to C$ were never used anyway. The step which consists in transforming the above redundant proof figure into the deduction $\mathcal{D}_1[\mathcal{D}_2/x]$ is called a *reduction step* or *normalization step.*

The idea of *proof normalization* goes back to Gentzen ([21], 1935). Gentzen noted that (formal) proofs can contain redundancies, or "detours", and that most complications in the analysis of proofs are due to these redundancies. Thus, Gentzen had the idea that the analysis of proofs would be simplified if it was possible to show that every proof can be converted to an equivalent irredundant proof, a proof in normal form. Gentzen proved a technical result to that effect, the "cut-elimination theorem", for a sequent-calculus formulation of first-order logic [21]. Cut-free proofs are direct, in the sense that they never use auxiliary lemmas via the cut rule.

Remark: It is important to note that Gentzen's result gives a particular algorithm to produce a proof in normal form. Thus, we know that every proof can be reduced to some normal form using a specific strategy, but there may be more than one normal form, and certain normalization strategies may not terminate.

About thirty years later, Prawitz ([36], 1965) reconsidered the issue of proof normalization, but in the framework of natural deduction rather than the framework of sequent calculi.[1] Prawitz explained very clearly what redundancies are in systems of natural deduction, and he proved that every proof can be reduced to a normal form. Furthermore, this normal form is unique. A few years later, Prawitz ([37], 1971) showed that in fact, every reduction sequence terminates, a property also called *strong normalization*.

A remarkable connection between proof normalization and the notion of computation must also be mentioned. Curry (1958) made the remarkably insightful observation that certain typed combinators can be viewed as representations of proofs (in a Hilbert system) of certain propositions (See in Curry and Feys [13] (1958), Chapter 9E, pages 312-315.) Building up on this observation, Howard ([30], 1969) described a general correspondence between propositions and types, proofs in natural deduction and certain typed λ-terms, and proof normalization and β-reduction. (The simply-typed-λ-calculus was invented by Church, 1940). This correspondence, usually referred to as the *Curry/Howard isomorphism* or *formulae–as–types principle*, is fundamental and very fruitful.

The Curry/Howard isomorphism establishes a deep correspondence between the notion of proof and the notion of computation. Furthermore, and this is the deepest aspect of the Curry/Howard isomorphism, proof normalization corresponds to term reduction in the λ-calculus associated with the proof system. To make the story short, the correspondence between proofs in intuitionistic logic and typed λ-terms on one-hand and between proof normalization and β-conversion on the other hand can be used to translate results about typed λ-terms into results about proofs in intuitionistic logic. By the way, some aspects of the Curry/Howard isomorphism are covered in CIS500.

In summary, using either some suitable intuitionistic sequent calculi and Gentzen's cut elimination theorem or some suitable typed λ-calculi and (strong) normalization results about them, it is possible to prove that there is a decision procedure for propositional intuitionistic logic. However, it can also be shown that the time-complexity of any such procedure is very high. Here, we are alluding to *complexity theory*, another active area of computer

[1] This is somewhat ironical, since Gentzen began his investigations using a natural deduction system, but decided to switch to sequent calculi (known as Gentzen systems!) for technical reasons.

science. You will learn about some basic and fundamental aspects of this theory in CIS262 when you learn about the two problems P and NP.

Readers who wish to learn more about these topics can read my two survey papers Gallier [18] (on the Correspondence Between Proofs and λ-Terms) and Gallier [17] (A Tutorial on Proof Systems and Typed λ-Calculi), both available on the web site

http://www.cis.upenn.edu/˜jean/gbooks/logic.html

and the excellent introduction to proof theory by Troelstra and Schwichtenberg [43].

Anybody who really wants to understand logic should of course take a look at Kleene [32] (the famous "I.M."), but this is not recommended to beginners!

Let us return to the question of deciding whether a proposition is not provable. To simplify the discussion, let us restrict our attention to propositional classical logic. So far, we have presented a very *proof-theoretic* view of logic, that is, a view based on the notion of provability as opposed to a more *semantic* view of based on the notions of truth and models. A possible excuse for our bias is that, as Peter Andrews (from CMU) puts it, "truth is elusive". Therefore, it is simpler to understand what truth is in terms of the more "mechanical" notion of provability. (Peter Andrews even gave the subtitle

To Truth Through Proof

to his logic book Andrews [1]!)

However, mathematicians are not mechanical theorem provers (even if they prove lots of stuff)! Indeed, mathematicians almost always think of the objects they deal with (functions, curves, surfaces, groups, rings, *etc.*) as rather concrete objects (even if they may not seem concrete to the uninitiated) and not as abstract entities solely characterized by arcane axioms.

It is indeed natural and fruitful to try to interpret formal statements semantically. For propositional classical logic, this can be done quite easily if we interpret atomic propositional letters using the truth values **true** and **false**, as explained in Section 1.6. Then, the crucial point that *every provable proposition (say in $\mathcal{NG}_c^{\Rightarrow,\vee,\wedge,\perp}$) has the value* **true** *no matter how we assign truth values to the letters in our proposition*. In this case, we say that P is *valid*.

The fact that provability implies validity is called *soundness* or *consistency* of the proof system. The soundness of the proof system $\mathcal{NG}_c^{\Rightarrow,\vee,\wedge,\perp}$ is easy to prove, as sketched in Section 1.6.

We now have a method to show that a proposition, P, is not provable: Find some truth assignment that makes P **false**.

Such an assignment falsifying P is called a *counter-example*. If P has a counter-example, then it can't be provable because if it were, then by soundness it would be **true** for all possible truth assignments.

But now, another question comes up: If a proposition is not provable, can we always find a counter-example for it. Equivalently, *is every valid proposition provable*? If every valid proposition is provable, we say that our proof system is *complete* (this is the *completeness* of our system).

The system $\mathcal{NG}_c^{\Rightarrow,\vee,\wedge,\perp}$ is indeed complete. In fact, *all* the classical systems that we have discussed are sound and complete. Completeness is usually a lot harder to prove than soundness. For first-order classical logic, this is known as *Gödel's completeness Theorem* (1929). Again, we refer our readers to Gallier [19] van Dalen [44] or or Huth and Ryan [31] for a thorough discussion of these matters. In the first-order case, one has to define *first-order structures* (or *first-order models*).

What about intuitionistic logic?

Well, one has to come up with a richer notion of semantics because it is no longer true that if a proposition is valid (in the sense of our two-valued semantics using **true**, **false**), then it is provable. Several semantics have been given for intuitionistic logic. In our opinion, the most natural is the notion of *Kripke model*, presented in Section 1.7. Then, again, soundness and completeness holds for intuitionistic proof systems, even in the first-order case (see Section 1.7 and van Dalen [44]).

In summary, semantic models can be use to provide *counter-examples* of unprovable propositions. This is a quick method to establish that a proposition is not provable.

The way we presented deduction trees and proof trees may have given our readers the impression that the set of premises, Γ, was just an auxiliary notion. Indeed, in all of our examples, Γ ends up being empty! However, nonempty Γ's are crucially needed if we want to develop theories about various kinds of structures and objects, such as the natural numbers, groups, rings, fields, trees, graphs, sets, *etc.* Indeed, we need to make definitions about the objects we want to study and we need to state some axioms asserting the main properties of these objects. We do this by putting these definitions and axioms in Γ. Actually, we have to allow Γ to be infinite but we still require that our deduction trees are finite; they can only use finitely many of the propositions in Γ. We are then interested in all propositions, P, such that $\Delta \to P$ is provable, where Δ is any finite subset of Γ; the set of all such P's is called a *theory*. Of course we have the usual problem of consistency: If we are not careful, our theory may be inconsistent, i.e., it may consist of all propositions.

Let us give two examples of theories.

Our first example is the *theory of equality*. Indeed, our readers may have noticed that we have avoided to deal with the equality relation. In practice, we can't do that.

Given a language, **L**, with a given supply of constant, function and predicate symbols, the theory of equality consists of the following formulae taken as axioms:

$$\begin{aligned}
&\forall(x = x)\\
&\forall x_1 \cdots \forall x_n \forall y_1 \cdots \forall y_n [(x_1 = y_1 \wedge \cdots \wedge x_n = y_n) \Rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)]\\
&\forall x_1 \cdots \forall x_n \forall y_1 \cdots \forall y_n [(x_1 = y_1 \wedge \cdots \wedge x_n = y_n) \wedge P(x_1, \ldots, x_n) \Rightarrow P(y_1, \ldots, y_n)],
\end{aligned}$$

for all function symbols (of n arguments) and all predicate symbols (of n arguments), including the equality predicate, $=$, itself.

It is not immediately clear from the above axioms that $=$ is reflexive and transitive but this can shown easily.

Our second example is the first-order theory of the natural numbers known as *Peano's arithmetic*.

Here, we have the constant 0 (zero), the unary function symbol S (for successor function; the intended meaning is $S(n) = n + 1$) and the binary function symbols $+$ (for addition) and $*$ (for multiplication). In addition to the axioms for the theory of equality we have the following axioms:

$$\begin{array}{l}
\forall x \neg (S(x) = 0) \\
\forall x \forall y (S(x) = S(y) \Rightarrow x = y) \\
\forall x \forall y (x + 0 = x) \\
\forall x \forall y (x + S(y) = S(x + y)) \\
\forall x \forall y (x * 0 = 0) \\
\forall x \forall y (x * S(y) = x * y + x) \\
[A(0) \land \forall x (A(x) \Rightarrow A(S(x)))] \Rightarrow \forall n A(n),
\end{array}$$

where A is any first-order formula with one free variable. This last axiom is the *induction axiom*. Observe how $+$ and $*$ are defined recursively in terms of 0 and S and that there are infinitely many induction axioms (countably many).

Many properties that hold for the natural numbers (i.e., are true when the symbols $0, S, +, *$ have their usual interpretation and all variables range over the natural numbers) can be proved in this theory (Peano's arithmetic), but not all! This is another very famous result of Gödel known as *Gödel's incompleteness Theorem* (1931). However, the topic of incompleteness is definitely oustside the scope of this course, so we will not say anymore about it. Another very interesting theory is *set theory*. There are a number of axiomatizations of set theory and we will discuss one of them (ZF) very briefly in the next section.

We close this section by repeating something we said ealier: There isn't just one logic but instead, *many* logics. In addition to classical and intuitionistic logic (propositional and first-order), there are: modal logics, higher-order logics and *linear logic*, a logic due to Jean-Yves Girard, attempting to unify classical and intuitionistic logic (among other goals). An excellent introduction to these logics can be found in Troelstra and Schwichtenberg [43]. We warn our readers that most presentations of linear logic are (very) difficult to follow. This is definitely true of Girard's seminal paper [23]. A more approachable version can be found in Girard, Lafont and Taylor [22], but most readers will still wonder what hit them when they attempt to read it.

In computer science, there is also *dynamic logic*, used to prove properties of programs and *temporal logic* and its variants (originally invented by A. Pnueli), to prove properties of real-time systems. So, logic is alive and well! Also, take a look at CIS482!

1.10 Basics Concepts of Set Theory

Having learned some fundamental notions of logic, it is now a good place before proceeding to more interesting things, such as functions and relations, to go through a very quick review of some basic concepts of set theory. This section will take the very "naive" point of view that a set is a collection of objects, the collection being regarded as a single object. Having first-order logic at our disposal, we could formalize set theory very rigorously in terms of axioms. This was done by Zermelo first (1908) and in a more satisfactory form by Zermelo and Fraenkel in 1921, in a theory known as the "Zermelo-Fraenkel" (ZF) axioms. Another axiomatization was given by John von Neumann in 1925 and later improved by Bernays in 1937. A modification of Bernay's axioms was used by Kurt Gödel in 1940. This approach is now known as "von Neumann-Bernays" (VNB) or "Gödel-Bernays" (GB) set theory. There are many books that give an axiomatic presentation of set theory. Among them, we recommend Enderton [15], which we find remarkably clear and elegant, Suppes [41] (a little more advanced) and Halmos [28], a classic (at a more elementary level).

However, it must be said that set theory was first created by Georg Cantor (1845-1918) between 1871 and 1879. However, Cantor's work was not unanimously well received by all mathematicians. Cantor regarded infinite objects as objects to be treated in much the same way as finite sets, a point of view that was shocking to a number of very prominent mathematicians who bitterly attacked him (among them, the powerful Kronecker). Also, it turns out that some paradoxes in set theory popped up in the early 1900, in particular, Russell's paradox. Russell's paradox (found by Russell in 1902) has to to with the

"set of all sets that are not members of themselves"

which we denote by

$$R = \{x \mid x \notin x\}.$$

(In general, the notation $\{x \mid P\}$ stand for the set of all objects satisfying the property P.)

Now, classically, either $R \in R$ or $R \notin R$. However, if $R \in R$, then the definition of R says that $R \notin R$; if $R \notin R$, then again, the definition of R says that $R \in R$!

So, we have a contradiction and the existence of such a set is a paradox. The problem is that we are allowing a property (here, $P(x) = x \notin x$), which is "too wild" and circular in nature. As we will see, the way out, as found by Zermelo, is to place a restriction on the property P and to also make sure that P picks out elements from some already given set (see the Subset Axioms below).

The apparition of these paradoxes prompted mathematicians, with Hilbert among its leaders, to put set theory on firmer grounds. This was achieved by Zermelo, Fraenkel, von Neumann, Bernays and Gödel, to only name the major players.

In what follows, we are assuming that we are working in classical logic. We will introduce various operations on sets using definitions involving the logical connectives $\wedge$, $\vee$, $\neg$, $\forall$ and $\exists$. In order to ensure the existence of some of these sets requires some of the axioms of set theory, but we will be rather casual about that.

Given a set, A, we write that some object, a, is an element of (belongs to) the set A as

$$a \in A$$

and that a is not an element of A (does not belong to A) as

$$a \notin A.$$

When are two sets A and B equal? This corresponds to the first axiom of set theory, called

Extensionality Axiom

Two sets A and B are equal iff they have exactly the same elements, that is

$$\forall x(x \in A \Rightarrow x \in B) \wedge \forall x(x \in B \Rightarrow x \in A).$$

The above says: Every element of A is an element of B and conversely.

There is a special set having no elements at all, the *empty set*, denoted $\emptyset$. This is the

Empty Set Axiom

There is a set having no members. This set is denoted $\emptyset$ and it is characterized by the property

$$\forall x(x \notin \emptyset).$$

Remark: Beginners often wonder whether there is more than one empty set. For example, is the empty set of professors distinct from the empty set of potatoes?

The answer is, by the extensionality axiom, there is only *one* empty set!

Given any two objects a and b, we can form the set $\{a, b\}$ containing exactly these two objects. Amazingly enough, this must also be an axiom:

Pairing Axiom

Given any two objects a and b (think sets), there is a set, $\{a, b\}$, having as members just a and b.

Observe that if a and b are identical, then we have the set $\{a, a\}$, which is denoted by $\{a\}$ and is called a *singleton set* (this set has a as its only element).

To form bigger sets, we use the union operation. This too requires an axiom.

Union Axiom (Version 1)

For any two sets A and B, there is a set, $A \cup B$, called the *union of A and B* defined by

$$x \in A \cup B \quad \text{iff} \quad (x \in A) \vee (x \in B).$$

This reads, x is a member of $A \cup B$ if either x belongs to A or x belongs to B (or both). We also write

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

Using the union operation, we can form bigger sets by taking unions with singletons. For example, we can form

$$\{a, b, c\} = \{a, b\} \cup \{c\}.$$

Remark: We can systematically construct bigger and bigger sets by the following method: Given any set, A, let

$$A^+ = A \cup \{A\}.$$

If we start from the empty set, we obtain sets that can be used to define the natural numbers and the $+$ operation corresponds to the successor function on the natural numbers, i.e., $n \mapsto n + 1$.

Another operation is the power set formation. It is indeed a "powerful" operation, in the sense that it allows us to form very big sets. For this, it is helpful to define the notion of inclusion between sets. Given any two sets, A and B, we say that *A is a subset of B* (or that *A is included in B*), denoted $A \subseteq B$, iff every element of A is also an element of B, i.e.

$$\forall x(x \in A \Rightarrow x \in B).$$

We say that *A is a proper subset of B* iff $A \subseteq B$ and $A \neq B$. This implies that that there is some $b \in B$ with $b \notin A$. We usually write $A \subset B$.

Observe that the equality of two sets can be expressed by

$$A = B \quad \text{iff} \quad A \subseteq B \quad \text{and} \quad B \subseteq A.$$

Power Set Axiom

Given any set, A, there is a set, $\mathcal{P}(A)$, (also denoted 2^A) called the *power set of A* whose members are exactly the subsets of A, i.e.,

$$X \in \mathcal{P}(A) \quad \text{iff} \quad X \subseteq A.$$

For example, if $A = \{a, b, c\}$, then

$$\mathcal{P}(A) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\},$$

a set containing 8 elements. Note that the empty set and A itself are always members of $\mathcal{P}(A)$.

Remark: If A has n elements, it is not hard to show that $\mathcal{P}(A)$ has 2^n elements. For this reason, many people, including me, prefer the notation 2^A for the power set of A.

At this stage, we would like to define intersection and complementation. For this, given any set, A, and given a property, P, (specified by a first-order formula) we need to be able to define the subset of A consisting of those elements satisfying P. This subset is denoted by

$$\{x \in A \mid P\}.$$

Unfortunately, there are problems with this construction. If the formula, P, is somehow a circular definition and refers to the subset that we are trying to define, then some paradoxes may arise!

The way out is to place a restriction on the formula used to define our subsets, and this leads to the subset axioms, first formulated by Zermelo. These axioms are also called *comprehension axioms* or *axioms of separation*.

Subset Axioms

For every first-order formula, P, we have the axiom:

$$\forall A \exists X \forall x(x \in X \quad \text{iff} \quad (x \in A) \wedge P),$$

where P does *not* contain X as a free variable. (However, P may contain x free.)

The subset axiom says that for every set, A, there is a set, X, consisting exactly of those elements of A so that P holds. For short, we usually write

$$X = \{x \in A \mid P\}.$$

As an example, consider the formula

$$P(B, x) = x \in B.$$

Then, the subset axiom says

$$\forall A \exists X \forall x(x \in A \wedge x \in B),$$

which means that X is the set of elements that belong both to A and B. This is called the *intersection of A and B*, denoted by $A \cap B$. Note that

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

We can also define the *relative complement of B in A*, denoted $A - B$, given by the formula $P(x, B) = x \notin B$, so that

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

In particular, if A is any given set and B is any subset of A, the set $A - B$ is also denoted $\overline{B}$ and is called the *complement of* B. Because $\wedge, \vee$ and $\neg$ satisfy the de Morgan laws (remember, we are dealing with classical logic), for any set X, the operations of union, intersection and complementation on subsets of X satisfy various identities, in particular the de Morgan laws

$$\begin{aligned}\overline{A \cap B} &= \overline{A} \cup \overline{B}\\ \overline{A \cup B} &= \overline{A} \cap \overline{B}\\ \overline{\overline{A}} &= A,\end{aligned}$$

and various associativity, commutativity and distributivity laws.

So far, the union axiom only applies to two sets but later on we will need to form infinite unions. Thus, it is necessary to generalize our union axiom as follows:

Union Axiom (Final Version)

Given any set X (think of X as a set of sets), there is a set, $\bigcup X$, defined so that

$$x \in \bigcup X \quad \text{iff} \quad \exists B(B \in X \wedge x \in B).$$

This says that $\bigcup X$ consists of all elements that belong to some member of X.

If we take $X = \{A, B\}$, where A and B are two sets, we see that

$$\bigcup \{A, B\} = A \cup B,$$

and so, our final version of the union axiom subsumes our previous union axiom which we now discard in favor of the more general version.

Observe that

$$\bigcup \{A\} = A, \quad \bigcup \{A_1, \ldots, A_n\} = A_1 \cup \cdots \cup A_n.$$

and in particular, $\bigcup \emptyset = \emptyset$.

Using the subset axiom, we can also define infinite intersections. For every nonempty set, X, there is a set, $\bigcap X$, defined by

$$x \in \bigcap X \quad \text{iff} \quad \forall B(B \in X \Rightarrow x \in B).$$

The existence of $\bigcap X$ is justified as follows: Since X is nonempty, it contains some set, A; let

$$P(X, x) = \forall B(B \in X \Rightarrow x \in B).$$

Then, the subset axiom asserts the existence of a set Y so that for every x,

$$x \in Y \quad \text{iff} \quad x \in A \quad \text{and} \quad P(X, x)$$

which is equivalent to

$$x \in Y \quad \text{iff} \quad P(X, x).$$

Therefore, the set Y is our desired set, $\bigcap X$.

Observe that

$$\bigcap\{A, B\} = A \cap B, \quad \bigcap\{A_1, \ldots, A_n\} = A_1 \cap \cdots \cap A_n.$$

Note that $\bigcap \emptyset$ is not defined. Intuitively, it would have to be the set of all sets, but such a set does not exist, as we now show. This is basically a version of Russell's paradox.

Theorem 1.10.1 *(Russell) There is no set of all sets, i.e., there is no set to which every other set belongs.*

Proof. Let A be any set. We construct a set, B, that does not belong to A. If the set of all sets existed, then we could produce a set that does not belong to it, a contradiction. Let

$$B = \{a \in A \mid a \notin a\}.$$

We claim that $B \notin A$. We proceed by contradiction, so assume $B \in A$. However, by the definition of B, we have

$$B \in B \quad \text{iff} \quad B \in A \quad \text{and} \quad B \notin B.$$

Since $B \in A$, the above is equivalent to

$$B \in B \quad \text{iff} \quad B \notin B,$$

which is a contradiction. Therefore, $B \notin A$ and we deduce that there is no set of all sets. □

Remarks:

(1) We should justify why the equivalence $B \in B$ iff $B \notin B$ is a contradiction. What we mean by "a contradiction" is that if the above equivalence holds, then we can derive $\perp$ (falsity) and thus, all propositions become provable. This is because we can show that for any proposition, P, if $P \equiv \neg P$ is provable, then $\neg(P \equiv \neg P)$ is also provable. We leave the proof of this fact as an easy exercise for the reader. By the way, this holds classically as well as intuitionistically.

(2) We said that in the subset axiom, the variable X is not allowed to occur free in P. A slight modification of Russell's paradox shows that allowing X to be free in P leads to paradoxical sets. For example, pick A to be any nonempty set and set $P(X, x) = x \notin X$. Then, look at the (alleged) set

$$X = \{x \in A \mid x \notin X\}.$$

As an exercise, the reader should show that X is empty iff X is nonempty!

This is as far as we can go with the elementary notions of set theory that we have introduced so far. In order to proceed further, we need to define relations and functions, which is the object of the next Chapter.

The reader may also wonder why we have not yet discussed infinite sets. This is because we don't know how to show that they exist! Again, perhaps surprisingly, this takes another axiom, the *axiom of infinity*. We also have to define when a set is infinite. However, we will not go into this right now. Instead, we will accept that the set of natural numbers, $\mathbb{N}$, exists and is infinite. Once we have the notion of a function, we will be able to show that other sets are infinite by comparing their "size" with that of $\mathbb{N}$ (This is the purpose of *cardinal numbers*, but this would lead us too far afield).

Remark: In an axiomatic presentation of set theory, the natural numbers can be defined from the empty set using the operation $A \mapsto A^+ = A \cup \{A\}$ introduced just after the union axiom. The idea due to von Neumann is that

$$\begin{array}{rcl} 0 & = & \emptyset \\ 1 & = & 0^+ = \{\emptyset\} = \{0\} \\ 2 & = & 1^+ = \{\emptyset, \{\emptyset\}\} = \{0, 1\} \\ 3 & = & 2^+ = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} = \{0, 1, 2\} \\ & \vdots & \\ n+1 & = & n^+ = \{0, 1, 2, \ldots, n\} \\ & \vdots & \end{array}$$

However, the above subsumes induction! Thus, we have to proceed in a different way to avoid circularities.

Definition 1.10.2 We say that a set, X, is *inductive* iff

(1) $\emptyset \in X$;

(2) For every $A \in X$, we have $A^+ \in X$.

Axiom of Infinity

There is some inductive set.

Having done this, we make the

Definition 1.10.3 A *natural number* is a set that belongs to every inductive set.

Using the subset axioms, we can show that there is a set whose members are exactly the natural numbers. The argument is very similar to the one used to prove that arbitrary intersections exist. By the Axiom of infinity, there is some inductive set, say A. Now consider the property, $P(x)$, which asserts that x belongs to every inductive set. By the subset axioms applied to P, there is a set, $\mathbb{N}$, such that

$$x \in \mathbb{N} \quad \text{iff} \quad x \in A \quad \text{and} \quad P(x)$$

and since A is inductive and P says that x belongs to every inductive set, the above is equivalent to

$$x \in \mathbb{N} \quad \text{iff} \quad P(x),$$

that is, $x \in \mathbb{N}$ iff x belongs to every inductive set. Therefore, the set of all natural numbers, $\mathbb{N}$, does exist. The set $\mathbb{N}$ is also denoted ω. We can now easily show

Theorem 1.10.4 *The set $\mathbb{N}$ is inductive and it is a subset of every inductive set.*

Proof. Recall that $\emptyset$ belongs to every inductive set; so, $\emptyset$ is a natural number (0). As $\mathbb{N}$ is the set of natural numbers, $\emptyset$ $(= 0)$ belongs to $\mathbb{N}$. Secondly, if $n \in \mathbb{N}$, this means that n belongs to every inductive set (n is a natural number), which implies that $n^+ = n+1$ belongs to every inductive set, which means that $n+1$ is a natural number, i.e., $n+1 \in \mathbb{N}$. Since $\mathbb{N}$ is the set of natural numbers and since every natural number belongs to every inductive set, we conclude that $\mathbb{N}$ is a subset of every inductive set. □

It would be tempting to view $\mathbb{N}$ as the intersection of the family of inductive sets, but unfortunately this family is not a set; it is too "big" to be a set.

As a consequence of the above fact, we obtain the

Induction Principle for $\mathbb{N}$: Any inductive subset of $\mathbb{N}$ is equal to $\mathbb{N}$ itself.

Now, in our setting, $0 = \emptyset$ and $n^+ = n+1$, so the above principle can be restated as follows:

Induction Principle for $\mathbb{N}$ (Version 2): For any subset, $S \subseteq \mathbb{N}$, if $0 \in S$ and $n+1 \in S$ whenever $n \in S$, then $S = \mathbb{N}$.

We will see how to rephrase this induction principle a little more conveniently in terms of the notion of function in the next chapter.

Remarks:

1. We still don't know what an infinite set is or, for that matter, that $\mathbb{N}$ is infinite! This will be shown in the next Chapter (see Corollary 2.9.7).

2. Zermelo-Fraenkel set theory (+ Choice) has three more axioms that we did not discuss: The *Axiom of Choice*, the *Replacement Axioms* and the *Regularity Axiom*. For our purposes, only the Axiom of Choice will be needed and we will introduce it in Chapter 2. Let us just say that the Replacement Axioms are needed to deal with ordinals and cardinals and that the Regularity Axiom is needed to show that every set is grounded. For more about these axioms, see Enderton [15], Chapter 7. The Regularity Axiom also implies that no set can be a member of itself, an eventuality that is not ruled out by our current set of axioms!

Chapter 2

Relations, Functions, Partial Functions

2.1 What is a Function?

We use functions all the time in Mathematics and in Computer Science. But, what exactly is a function?

Roughly speaking, a function, f, is a rule or mechanism, which takes input values in some *input domain*, say X, and produces output values in some *output domain*, say Y, in such a way that to each input $x \in X$ corresponds a *unique* output value $y \in Y$, denoted $f(x)$. We usually write $y = f(x)$, or better, $x \mapsto f(x)$.

Often, functions are defined by some sort of closed expression (a formula), but not always. For example, the formula

$$y = 2x$$

defines a function. Here, we can take both the input and output domain to be $\mathbb{R}$, the set of real numbers. Instead, we could have taken $\mathbb{N}$, the set of natural numbers; this gives us a different function. In the above example, $2x$ makes sense for all input x, whether the input domain is $\mathbb{N}$ or $\mathbb{R}$, so our formula yields a function defined for all of its input values.

Now, look at the function defined by the formula

$$y = \frac{x}{2}.$$

If the input and output domains are both $\mathbb{R}$, again this function is well-defined. However, what if we assume that the input and output domains are both $\mathbb{N}$? This time, we have a problem when x is odd. For example, $\frac{3}{2}$ is not an integer, so our function is not defined for all of its input values. It is a *partial function*. Observe that this function is defined for the set of even natural numbers (sometimes denoted $2\mathbb{N}$) and this set is called the *domain* (of definition) of f. If we enlarge the output domain to be $\mathbb{Q}$, the set of rational numbers, then our function is defined for all inputs.

Another example of a partial function is given by

$$y = \frac{x+1}{x^2 - 3x + 2},$$

assuming that both the input and output domains are $\mathbb{R}$. Observe that for $x = 1$ and $x = 2$, the denominator vanishes, so we get the undefined fractions $\frac{2}{0}$ and $\frac{3}{0}$. The function "blows up" for $x = 1$ and $x = 2$, its value is "infinity" ($= \infty$), which is not an element of $\mathbb{R}$. So, the domain of f is $\mathbb{R} - \{1, 2\}$.

In summary, functions need not be defined for all of their input values and we need to pay close attention to both the input and the ouput domain of our functions.

The following example illustrates another difficulty: Consider the function given by

$$y = \sqrt{x}.$$

If we assume that the input domain is $\mathbb{R}$ and that the output domain is $\mathbb{R}^+ = \{x \in \mathbb{R} \mid x \geq 0\}$, then this function is not defined for negative values of x. To fix this problem, we can extend the output domain to be $\mathbb{C}$, the complex numbers. Then we can make sense of $\sqrt{x}$ when $x < 0$. However, a new problem comes up: Every negative number, x, has two complex square roots, $-i\sqrt{-x}$ and $+i\sqrt{-x}$ (where i is "the" square root of -1). Which of the two should we pick?

In this case, we could systematically pick $+i\sqrt{-x}$ but what if we extend the input domain to be $\mathbb{C}$. Then, it is not clear which of the two complex roots should be picked, as there is no obvious total order on $\mathbb{C}$. We can treat f as a *multi-valued function*, that is, a function that may return several possible outputs for a given input value.

Experience shows that it is akward to deal with multi-valued functions and that it is best to treat them as relations (or to change the output domain to be a power set, which is equivalent to view the function as a relation).

Let us give one more example showing that it is not always easy to make sure that a formula is a proper definition of a function. Consider the function from $\mathbb{R}$ to $\mathbb{R}$ given by

$$f(x) = 1 + \sum_{n=1}^{\infty} \frac{x^n}{n!}.$$

Here, $n!$ is the function *factorial*, defined by

$$n! = n \cdot (n-1) \cdots 2 \cdot 1.$$

How do we make sense of this infinite expression? Well, that's where analysis comes in, with the notion of limit of a series, etc. It turns out that $f(x)$ is the exponential function $f(x) = e^x$. Actually, e^x is even defined when x is a complex number or even a square matrix (with real or complex entries)! Don't panic, we will not use such functions in this course.

Another issue comes up, that is, the notion of *computability*. In all of our examples, and for most functions we will ever need to compute, it is clear that it is possible to give a mechanical procedure, i.e., a computer program which computes our functions (even if it hard to write such a program or if such a program takes a very long time to compute the output from the input).

Unfortunately, there are functions which, although well-defined mathematically, are not computable! For an example, let us go back to first-order logic and the notion of provable proposition. Given a finite (or countably infinite) alphabet of function, predicate, constant symbols, and a countable supply of variables, it is quite clear that the set $\mathcal{F}$ of all propositions built up from these symbols and variables can be enumerated systematically. We can define the function, Prov, with input domain $\mathcal{F}$ and output domain $\{0, 1\}$, so that, for every proposition $P \in \mathcal{F}$,

$$\mathrm{Prov}(P) = \begin{cases} 1 & \text{if } P \text{ is provable (classically)} \\ 0 & \text{if } P \text{ is not provable (classically).} \end{cases}$$

Mathematically, for every proposition, $P \in \mathcal{F}$, either P is provable or it is not, so this function makes sense. However, by Church's Theorem (see Section 1.9), we know that there is **no** computer program that will terminate for all input propositions and give an answer in a finite number of steps! So, although the function Prov makes sense as an abstract function, it is not computable. Is this a paradox? No, if we are careful when defining a function not to incorporate in the definition any notion of computability and instead to take a more abstract and, in some some sense, naive view of a function as some kind of input/output process given by pairs ⟨input value, output value⟩ (without worrying about the way the output is "computed" from the input). A rigorous way to proceed is to use the notion of ordered pair and of graph of a function. Before we do so, let us point out some facts about functions that were revealed by our examples:

1. In order to define a function, in addition to defining its input/output behavior, it is also important to specify what is its *input domain* and its *output domain*.

2. Some functions may not be defined for all of their input values; a function can be a *partial function*.

3. The input/output behavior of a function can be defined by a set of ordered pairs. As we will see next, this is the *graph* of the function.

We are now going to formalize the notion of function (possibly partial) using the concept of ordered pair.

2.2 Ordered Pairs, Cartesian Products, Relations, Functions, Partial Functions

Given two sets, A and B, one of the basic constructions of set theory is the formation of an *ordered pair*, $\langle a, b\rangle$, where $a \in A$ and $b \in B$. Sometimes, we also write (a, b) for an ordered pair. The main property of ordered pairs is that if $\langle a_1, b_1\rangle$ and $\langle a_2, b_2\rangle$ are ordered pairs, where $a_1, a_2 \in A$ and $b_1, b_2 \in B$, then

$$\langle a_1, b_1\rangle = \langle a_2, b_2\rangle \quad \text{iff} \quad a_1 = a_2 \quad \text{and} \quad b_1 = b_2.$$

Observe that this property implies that,

$$\langle a, b\rangle \neq \langle b, a\rangle,$$

unless $a = b$. Thus, the ordered pair, $\langle a, b\rangle$, is not a notational variant for the set $\{a, b\}$; implicit to the notion of ordered pair is the fact that there is an order (even though we have not yet defined this notion yet!) among the elements of the pair. Indeed, in $\langle a, b\rangle$, the element a comes first and b comes second. Accordingly, given an ordered pair, $p = \langle a, b\rangle$, we will denote a by $pr_1(p)$ and b by $pr_2(p)$ (*first and second projection* or *first and second coordinate*).

Remark: Readers who like set theory will be happy to hear that an ordered pair, $\langle a, b\rangle$, can be defined as the set $\{\{a\}, \{a, b\}\}$. This definition is due to Kuratowski, 1921. An earlier (more complicated) definition given by N. Wiener in 1914 is $\{\{\{a\}, \emptyset\}, \{\{b\}\}\}$.

Now, from set theory, it can be shown that given two sets, A and B, the set of all ordered pairs $\langle a, b\rangle$, with $a \in A$ and $b \in B$, is a set denoted $A \times B$ and called the *Cartesian product of A and B* (in that order). By convention, we agree that $\emptyset \times B = A \times \emptyset = \emptyset$. To simplify the terminology, we often say *pair* for *ordered pair*, with the understanding that pairs are always ordered (otherwise, we should say set).

Of course, given three sets, A, B, C, we can form $(A \times B) \times C$ and we call its elements (ordered) *triples* (or *triplets*). To simplify the notation, we write $\langle a, b, c\rangle$ instead of $\langle\langle a, b\rangle, c\rangle$. More generally, given n sets $A_1, \ldots, A_n$ $(n \geq 2)$, we define the set of *n-tuples*, $A_1 \times A_2 \times \cdots \times A_n$, as $(\cdots((A_1 \times A_2) \times A_3) \times \cdots) \times A_n$. An element of $A_1 \times A_2 \times \cdots \times A_n$ is denoted by $\langle a_1, \ldots, a_n\rangle$ (an n-tuple). We agree that when $n = 1$, we just have A_1 and a 1-tuple is just an element of A_1.

We now have all we need to define relations.

Definition 2.2.1 Given two sets, A and B, a (binary) *relation, R, between A and B* is any subset $R \subseteq A \times B$ of ordered pairs from $A \times B$. When $\langle a, b\rangle \in R$, we also write aRb and we say that *a and b are related by R*. The set

$$dom(R) = \{a \in A \mid \exists b \in B,\ \langle a, b\rangle \in R\}$$

is called the *domain of* R and the set

$$range(R) = \{b \in B \mid \exists a \in A,\ \langle a, b\rangle \in R\}$$

is called the *range of* R. Note that $dom(R) \subseteq A$ and $range(R) \subseteq B$. When $A = B$, we often say that R *is a (binary) relation over* A.

Among all relations between A and B, we mention three relations that play a special role:

1. $R = \emptyset$, the *empty relation*. Note that $dom(\emptyset) = range(\emptyset) = \emptyset$. This is not a very exciting relation!

2. When $A = B$, we have the *identity relation*,

   $$\mathrm{id}_A = \{\langle a, a\rangle \mid a \in A\}.$$

   The identity relation relates every element to itself, and that's it! Note that $dom(\mathrm{id}_A) = range(\mathrm{id}_A) = A$.

3. The relation $A \times B$ itself. This relation relates every element of A to every element of B. Note that $dom(A \times B) = A$ and $range(A \times B) = B$.

Relations can be represented graphically by pictures often called graphs. (Beware, the term "graph" is very much overloaded. Later on, we will define what a graph is.) We depict the elements of both sets A and B as points (perhaps with different colors) and we indicate that $a \in A$ and $b \in B$ are related (i.e., $\langle a, b\rangle \in R$) by drawing an oriented edge (an arrow) starting from a (its source) and ending in b (its target). Here is an example:

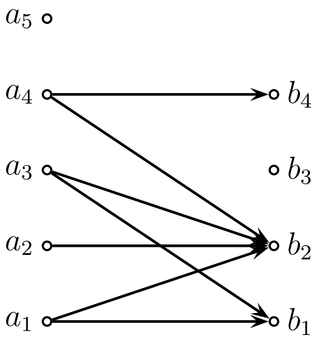

Figure 2.1: A binary relation, R

In Figure 2.1, $A = \{a_1, a_2, a_3, a_4, a_5\}$ and $B = \{b_1, b_2, b_3, b_4\}$. Observe that a_5 is not related to any element of B, b_3 is not related to any element of A and that some elements of A, namely, a_1, a_3, a_4, are related to several elements of B.

Now, given a relation, $R \subseteq A \times B$, some element $a \in A$ may be related to several distinct elements $b \in B$. If so, R does not correspond to our notion of a function, because we want our functions to be single-valued. So, we impose a natural condition on relations to get relations that correspond to functions.

Definition 2.2.2 We say that a relation, R, between two sets A and B is *functional* if for every $a \in A$, there is *at most one* $b \in B$ so that $\langle a, b\rangle \in R$. Equivalently, R is functional if for all $a \in B$ and all $b_1, b_2 \in B$, if $\langle a, b_1\rangle \in R$ and $\langle a, b_2\rangle \in R$, then $b_1 = b_2$.

The picture in Figure 2.2 shows an example of a functional relation.

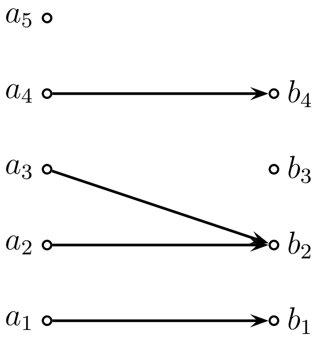

Figure 2.2: A functional relation G

Using Definition 2.2.2, we can give a rigorous definition of a function (partial or not).

Definition 2.2.3 A *partial function, f*, is a triple, $f = \langle A, G, B\rangle$, where A is a set called the *input domain of f*, B is a set called the *output domain of f* (sometimes *codomain of f*) and $G \subseteq A \times B$ is a functional relation called the *graph of f* (see Figure 2.3); we let $\mathit{graph}(f) = G$. We write $f\colon A \to B$ to indicate that A is the input domain of f and that B is the codomain of f and we let $\mathit{dom}(f) = \mathit{dom}(G)$ and $\mathit{range}(f) = \mathit{range}(G)$. For every $a \in \mathit{dom}(f)$, the unique element, $b \in B$, so that $\langle a, b\rangle \in \mathit{graph}(f)$ is denoted by $f(a)$ (so, $b = f(a)$). Often, we say that $b = f(a)$ is the *image of a by f*. The range of f is also called the *image of f* and is denoted $\mathrm{Im}\,(f)$. If $\mathit{dom}(f) = A$, we say that f is a *total function*, for short, a *function with domain A*.

Remarks:

1. If $f = \langle A, G, B\rangle$ is a partial function and $b = f(a)$ for some $a \in \mathit{dom}(f)$, we say that *f maps a to b*; we may write $f\colon a \mapsto b$. For any $b \in B$, the set

$$\{a \in A \mid f(a) = b\}$$

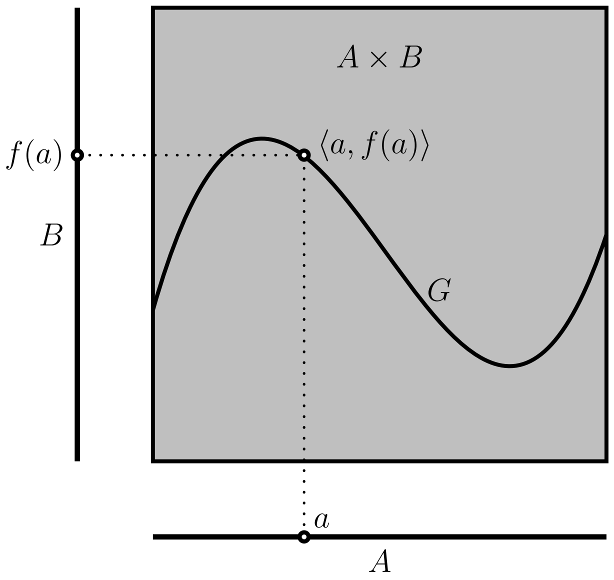

Figure 2.3: A (partial) function $\langle A, G, B\rangle$

is denoted $f^{-1}(b)$ and called the *inverse image* or *preimage of* b *by* f. (It is also called the *fibre of* f *above* b. We will explain this peculiar language later on.) Note that $f^{-1}(b) \neq \emptyset$ iff b is in the image (range) of f. Often, a function, partial or not, is called a *map*.

2. Note that Definition 2.2.3 allows $A = \emptyset$. In this case, we must have $G = \emptyset$ and, technically, $\langle \emptyset, \emptyset, B\rangle$ is total function! It is the *empty function from* $\emptyset$ *to* B.

3. When a partial function is a total function, we don't call it a "partial total function", but simply a "function". The usual pratice is that the term "function" refers to a total function. However, sometimes, we say "total function" to stress that a function is indeed defined on all of its input domain.

4. Note that if a partial function $f = \langle A, G, B\rangle$ is not a total function, then $dom(f) \neq A$ and for all $a \in A - dom(f)$, there is **no** $b \in B$ so that $\langle a, b\rangle \in graph(f)$. This corresponds to the intuitive fact that f does not produce any output for any value not in its domain of definition. We can imagine that f "blows up" for this input (as in the situation where the denominator of a fraction is 0) or that the program computing f loops indefinitely for that input.

5. If $f = \langle A, G, B\rangle$ is a total function and $A \neq \emptyset$, then $B \neq \emptyset$.

6. For any set, A, the identity relation, id_A, is actually a function $\mathrm{id}_A\colon A \to A$.

7. Given any two sets, A and B, the rules $\langle a, b\rangle \mapsto a = pr_1(\langle a, b\rangle)$ and $\langle a, b\rangle \mapsto b =$

$pr_2(\langle a, b\rangle)$ make pr_1 and pr_2 into functions $pr_1\colon A \times B \to A$ and $pr_2\colon A \times B \to B$ called the *first and second projections*.

8. A function, $f\colon A \to B$, is sometimes denoted $A \xrightarrow{f} B$. Some authors use a different kind of arrow to indicate that f is partial, for example, a dotted or dashed arrow. We will not go that far!

9. The set of all functions, $f\colon A \to B$, is denoted by B^A. If A and B are finite, A has m elements and B has n elements, it is easy to prove that B^A has n^m elements.

The reader might wonder why, in the definition of a (total) function, $f\colon A \to B$, we do not require $B = \mathrm{Im}\, f$, since we require that $\mathrm{dom}(f) = A$.

The reason has to do with experience and convenience. It turns out that in most cases, we know what the domain of a function is, but it may be very hard to determine exactly what its image is. Thus, it is more convenient to be flexible about the codomain. As long as we know that f maps into B, we are satisfied.

For example, consider functions, $f\colon \mathbb{R} \to \mathbb{R}^2$, from the real line into the plane. The image of such a function is a *curve* in the plane $\mathbb{R}^2$. Actually, to really get "decent" curves we need to impose some reasonable conditions on f, for example, to be differentiable. Even continuity may yield very strange curves (see Section 2.10). But even for a very well behaved function, f, it may be very hard to figure out what the image of f is. Consider the function, $t \mapsto (x(t), y(y))$, given by

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

The curve which is the image of this function, shown in Figure 2.4, is called the "lemniscate of Bernoulli".

Observe that this curve has a self-intersection at the origin, which is not so obvious at first glance.

2.3 Induction Principles on $\mathbb{N}$

Now that we have the notion of function, we can restate the induction principle (Version 2) stated at the send of Section 1.10 to make it more flexible. We define a *property of the natural numbers* as any function, $P\colon \mathbb{N} \to \{\mathbf{true}, \mathbf{false}\}$. The idea is that $P(n)$ holds iff $P(n) = \mathbf{true}$, else $P(n) = \mathbf{false}$. Then, we have the following principle:

Principle of Induction for $\mathbb{N}$ (Version 3).

Let P be any property of the natural numbers. In order to prove that $P(n)$ holds for all $n \in \mathbb{N}$, it is enough to prove that

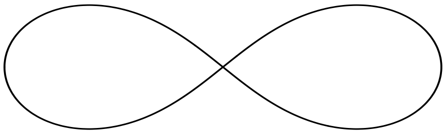

Figure 2.4: Lemniscate of Bernoulli

(1) $P(0)$ holds and

(2) For every $n \in \mathbb{N}$, the implication $P(n) \Rightarrow P(n+1)$ holds.

As a formula, (1) and (2) can be written

$$[P(0) \wedge (\forall n \in \mathbb{N})(P(n) \Rightarrow P(n+1))] \Rightarrow (\forall n \in \mathbb{N})P(n).$$

Step (1) is usually called the *basis* or *base step* of the induction and step (2) is called the *induction step*. In step (2), $P(n)$ is called the *induction hypothesis*. That the above induction principle is valid is given by the

Proposition 2.3.1 *The Principle of Induction stated above is valid.*

Proof. Let

$$S = \{n \in \mathbb{N} \mid P(n) = \mathbf{true}\}.$$

By the induction principle (Version 2) stated at the send of Section 1.10, it is enough to prove that S is inductive, because then $S = \mathbb{N}$ and we are done.

Since $P(0)$ hold, we have $0 \in S$. Now, if $n \in S$, i.e., if $P(n)$ holds, since $P(n) \Rightarrow P(n+1)$ holds for every n, we deduce that $P(n+1)$ holds, that is, $n+1 \in S$. Therefore, S is inductive as claimed and this finishes the proof. □

Induction is a very valuable tool for proving properties of the natural numbers and we will make extensive use of it. We will also see other more powerful induction principles. Let us give just one example illustrating how it is used.

Claim:

$$1 + 3 + 5 + \cdots + 2n + 1 = (n+1)^2,$$

where $n \in \mathbb{N}$.

For the basis of the induction, where $n = 0$, we get $1 = 1^2$, so the base step holds.

For the induction step, for any $n \in \mathbb{N}$, assume that

$$1 + 3 + 5 + \cdots + 2n + 1 = (n + 1)^2.$$

Consider $1 + 3 + 5 + \cdots + 2n + 1 + 2(n + 1) + 1 = 1 + 3 + 5 + \cdots + 2n + 1 + 2n + 3$. Then, using the induction hypotesis, we have

$$\begin{aligned} 1 + 3 + 5 + \cdots + 2n + 1 + 2n + 3 &= (n + 1)^2 + 2n + 3 \\ &= n^2 + 2n + 1 + 2n + 3 = n^2 + 4n + 4 \\ &= (n + 2)^2. \end{aligned}$$

Therefore, the induction step holds and this completes the proof by induction. $\square$

Sometimes, it is necessary to prove a property, $P(n)$, for all natural numbers $n \geq m$, where $m > 0$. Our induction principle does not seem to apply since the base case is not $n = 0$. However, we can define the property, $Q(n)$, given by

$$Q(n) = P(m + n), \quad n \in \mathbb{N},$$

and since $Q(n)$ holds for all $n \in \mathbb{N}$ iff $P(k)$ holds for all $k \geq m$, we can apply our induction principle to prove $Q(n)$ for all $n \in \mathbb{N}$ and thus, $P(k)$, for all $k \geq m$ (note, $k = m + n$). Of course, this amounts to considering that the base case is $n = m$ and this is what we always do without any further justification. Here is an example.

Let us prove that

$$(3n)^2 \leq 2^n, \quad \text{for all } \ n \geq 10.$$

The base case is $n = 10$. For $n = 10$, we get

$$(3 \times 10)^2 = 30^2 = 900 \leq 1024 = 2^{10},$$

which is indeed true. Let us now prove the induction step. Assuming that $(3n)^2 \leq 2^n$ holds for all $n \geq 10$, we want to prove that $(3(n + 1))^2 \leq 2^{n+1}$. Since

$$(3(n + 1))^2 = (3n + 3)^2 = (3n)^2 + 18n + 9$$

if we can prove that $18n+9 \leq (3n)^2$ when $n \geq 10$, using the induction hypothesis, $(3n)^2 \leq 2^n$, we will have

$$(3(n + 1))^2 = (3n)^2 + 18n + 9 \leq (3n)^2 + (3n)^2 \leq 2^n + 2^n = 2^{n+1},$$

establishing the induction step. However,

$$(3n)^2 - (18n + 9) = (3n - 3)^2 - 18$$

and $(3n - 3)^2 \geq 18$ as soon as $n \geq 3$, so $18n + 9 \leq (3n)^2$ when $n \geq 10$, as required.

There is another induction principle which is often more flexible that our original induction principle. This principle, called *complete induction* (or sometimes *strong induction*), is stated below.

Complete Induction Principle for $\mathbb{N}$.

In order to prove that a predicate, $P(n)$, holds for all $n \in \mathbb{N}$ it is enough to prove that

(1) $P(0)$ holds (the base case) and

(2) for every $m \in \mathbb{N}$, if $(\forall k \in \mathbb{N})(k \leq m \Rightarrow P(k))$ then $P(m+1)$.

The difference between ordinary induction and complete induction is that in complete induction, the induction hypothesis, $(\forall k \in \mathbb{N})(k \leq m \Rightarrow P(k))$, assumes that $P(k)$ holds for all $k \leq m$ and not just for m (as in ordinary induction), in order to deduce $P(m+1)$. This gives us more proving power as we have more knowledge in order to prove $P(m+1)$. Complete induction will be discussed more extensively in Section 4.3 and its validity will be proved as a consequence of the fact that every nonempty subset of $\mathbb{N}$ has a smallest element but we can also justify its validity as follows: Define $Q(m)$ by

$$Q(m) = (\forall k \in \mathbb{N})(k \leq m \Rightarrow P(k)).$$

Then, it is an easy exercise to show that if we apply our (ordinary) induction principle to $Q(m)$ (Induction Principle, Version 3), then we get the principle of complete induction. Here is an example of a proof using complete induction.

Define the sequence of natural numbers, u_n, (*Fibonacci sequence*) by

$$u_0 = 1,\ u_1 = 1,\ u_{n+2} = u_{n+1} + u_n,\ n \geq 0.$$

We claim that

$$u_n \geq \frac{3^{n-2}}{2^{n-3}}, \quad n \geq 3.$$

The base case corresponds to $n = 3$, where

$$u_3 = 3 \geq \frac{3^1}{2^0} = 3,$$

which is true. Note that we also need to consider the case $n = 4$ by itself before we do the induction step because even though $u_4 = u_3 + u_2$, the induction hypothesis only applies to u_3 ($n \geq 3$ in the inequality above). We have

$$u_4 = 5 \geq \frac{3^2}{2^1} = \frac{9}{2},$$

which is true since $10 > 9$. Now for the induction step where $n \geq 3$, we have

$$\begin{aligned} u_{n+2} &= u_{n+1} + u_n \\ &\geq \frac{3^{n-1}}{2^{n-2}} + \frac{3^{n-2}}{2^{n-3}} \\ &\geq \frac{3^{n-2}}{2^{n-3}}\left(1 + \frac{3}{2}\right) = \frac{3^{n-2}}{2^{n-3}}\frac{5}{2} \geq \frac{3^{n-2}}{2^{n-3}}\frac{9}{4} = \frac{3^n}{2^{n-1}}, \end{aligned}$$

since $\frac{5}{2} > \frac{9}{4}$, which concludes the proof of the induction step. Observe that we used the induction hypothesis for both u_{n+1} and u_n in order to deduce that it holds for u_{n+2}. This is where we needed the extra power of complete induction.

Remark: The Fibonacci sequence, u_n, is really a function from $\mathbb{N}$ to $\mathbb{N}$ defined recursively but we haven't proved yet that recursive definitions are legitimate methods for defining functions! In fact, certain restrictions are needed on the kind of recursion used to define functions. This topic will be explored further in Section 2.5. Using results from Section 2.5, it can be shown that the Fibonacci sequence is a well-defined function (but this does not follow immediately from Theorem 2.5.1).

A useful way to produce new relations or functions is to compose them.

2.4 Composition of Relations and Functions

We begin with the definition of the composition of relations.

Definition 2.4.1 Given two relations, $R \subseteq A \times B$ and $S \subseteq B \times C$, the *composition of R and S*, denoted $R \circ S$, is the relation between A and C defined by

$$R \circ S = \{\langle a, c\rangle \in A \times C \mid \exists b \in B,\ \langle a, b\rangle \in R \quad \text{and} \quad \langle b, c\rangle \in S\}.$$

One should check that for any relation $R \subseteq A \times B$, we have $\mathrm{id}_A \circ R = R$ and $R \circ \mathrm{id}_B = R$. If R and S are the graphs of functions, possibly partial, is $R \circ S$ the graph of some function? The answer is yes, as shown in the following

Proposition 2.4.2 *Let $R \subseteq A \times B$ and $S \subseteq B \times C$ be two relations.*

(a) If R and S are both functional relations, then $R \circ S$ is also a functional relation. Consequently, $R \circ S$ is the graph of some partial function.

(b) If $dom(R) = A$ and $dom(S) = B$, then $dom(R \circ S) = A$.

(c) If R is the graph of a (total) function from A to B and S is the graph of a (total) function from B to C, then $R \circ S$ is the graph of a (total) function from A to C.

Proof. (a) Assume that $\langle a, c_1\rangle \in R \circ S$ and $\langle a, c_2\rangle \in R \circ S$. By definition of $R \circ S$, there exist $b_1, b_2 \in B$ so that

$$\begin{aligned} \langle a, b_1\rangle \in R, &\quad \langle b_1, c_1\rangle \in S, \\ \langle a, b_2\rangle \in R, &\quad \langle b_2, c_2\rangle \in S. \end{aligned}$$

As R is functional, $\langle a, b_1\rangle \in R$ and $\langle a, b_2\rangle \in R$ implies $b_1 = b_2$. Let $b = b_1 = b_2$, so that $\langle b_1, c_1\rangle = \langle b, c_1\rangle$ and $\langle b_2, c_2\rangle = \langle b, c_2\rangle$. But, S is also functional, so $\langle b, c_1\rangle \in S$ and $\langle b, c_2\rangle \in S$ implies that $c_1 = c_2$, which proves that $R \circ S$ is functional.

(b) If $A = \emptyset$ then $R = \emptyset$ and so $R \circ S = \emptyset$, which implies that $dom(R \circ S) = \emptyset = A$. If $A \neq \emptyset$, pick any $a \in A$. The fact that $dom(R) = A \neq \emptyset$ means that there is some $b \in B$ so that $\langle a, b\rangle \in R$ and so, $B \neq \emptyset$. As $dom(S) = B \neq \emptyset$, there is some $c \in C$ so that $\langle b, c\rangle \in S$.

Then, by the definition of $R \circ S$, we see that $\langle a, c\rangle \in R \circ S$. Since the argument holds for any $a \in A$, we deduce that $dom(R \circ S) = A$.

(c) If R and S are the graphs of partial functions, then this means that they are functional and (a) implies that $R \circ S$ is also functional. This shows that $R \circ S$ is the graph of the partial function $\langle A, R \circ S, C\rangle$. If R and S are the graphs of total functions, then $dom(R) = A$ and $dom(S) = B$. By (b), we deduce that $dom(R \circ S) = A$. By the first part of (c), $R \circ S$ is the graph of the partial function $\langle A, R \circ S, C\rangle$, which is a total function, since $dom(R \circ S) = A$. $\square$

Proposition 2.4.2 shows that it is legitimate to define the composition of functions, possibly partial. Thus, we make the following

Definition 2.4.3 Given two functions, $f\colon A \to B$ and $g\colon B \to C$, possibly partial, the *composition of f and g*, denoted $g \circ f$, is the function (possibly partial)

$$g \circ f = \langle A, graph(f) \circ graph(g), C\rangle.$$

The reader must have noticed that the composition of two functions $f\colon A \to B$ and $g\colon B \to C$ is denoted $g \circ f$, whereas the graph of $g \circ f$ is denoted $graph(f) \circ graph(g)$. This "reversal" of the order in which function composition and relation composition are written is unfortunate and somewhat confusing.

Once again, we are victim of tradition. The main reason for writing function composition as $g \circ f$ is that traditionally, the result of applying a function, f, to an argument, x, is written $f(x)$. Then, $(g \circ f)(x) = g(f(x))$, because $z = (g \circ f)(x)$ iff there is some y so that $y = f(x)$ and $z = g(y)$, that is, $z = g(f(x))$. Some people, in particular algebraists, write function composition as $f \circ g$, but then, they write the result of applying a function f to an argument x as xf. With this convention, $x(f \circ g) = (xf)g$, which also makes sense.

We prefer to stick to the convention where we write $f(x)$ for the result of applying a function f to an argument x and, consequently, we use the notation $g \circ f$ for the composition of f with g, even though it is the opposite of the convention for writing the composition of relations.

Given any three relations, $R \subseteq A \times B$, $S \subseteq B \times C$ and $T \subseteq C \times D$, the reader should verify that

$$(R \circ S) \circ T = R \circ (S \circ T).$$

We say that composition is *associative*. Similarly, for any three functions (possibly partial), $f\colon A \to B$, $g\colon B \to C$ and $h\colon C \to D$, we have (associativity of function composition)

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

2.5 Recursion on $\mathbb{N}$

The following situation often occurs: We have some set, A, some fixed element, $a \in A$, some function, $g\colon A \to A$, and we wish to define a new function, $h\colon \mathbb{N} \to A$, so that

$$\begin{aligned} h(0) &= a, \\ h(n+1) &= g(h(n)) \qquad \text{for all} \quad n \in \mathbb{N}. \end{aligned}$$

This way of defining h is called a *recursive definition* (or a definition by *primitive recursion*). I would be surprised if any computer scientist had any trouble with this "definition" of h but how can we justify rigorously that such a function exists and is unique?

Indeed, the existence (and uniqueness) of h requires proof. The proof, although not really hard, is surprisingly involved and, in fact quite subtle. For those reasons, we will not give a proof of the following theorem but instead the main idea of the proof. The reader will find a complete proof in Enderton [15] (Chapter 4).

Theorem 2.5.1 *(Recursion Theorem on $\mathbb{N}$) Given any set, A, any fixed element, $a \in A$, and any function, $g\colon A \to A$, there is a unique function, $h\colon \mathbb{N} \to A$, so that*

$$\begin{aligned} h(0) &= a, \\ h(n+1) &= g(h(n)) \qquad \textit{for all} \quad n \in \mathbb{N}. \end{aligned}$$

Proof. The idea is to approximate h. To do this, define a function, f, to be *acceptable* iff

1. $dom(f) \subseteq \mathbb{N}$ and $range(f) \subseteq A$;
2. If $0 \in dom(f)$, then $f(0) = a$;
3. If $n+1 \in dom(f)$, then $n \in dom(f)$ and $f(n+1) = g(f(n))$.

Let $\mathcal{F}$ be the collection of all acceptable functions and set

$$h = \bigcup \mathcal{F}.$$

All we can say, so far, is that h is a relation. We claim that h is the desired function. For this, four things need to be proved:

1. The relation h is function.
2. The function h is acceptable.
3. The function h has domain $\mathbb{N}$.
4. The function h is unique.

As expected, we make heavy use of induction in proving (1), (2), (3) and (4). For complete details, see Enderton [15] (Chapter 4). □

Theorem 2.5.1 is very important. Indeed, experience shows that it is used almost as much as induction! As an example, we show how to define addition on $\mathbb{N}$. Indeed, at the moment, we know what the natural numbers are but we don't know what are the arithmetic operations such as $+$ or $*$! (at least, not in our axiomatic treatment; of course, nobody needs an axiomatic treatment to know how to add or multiply).

How do we define $m + n$, where $m, n \in \mathbb{N}$?

If we try to use Theorem 2.5.1 directly, we seem to have a problem, because addition is a function of two arguments, but h and g in the theorem only take one argument. We can overcome this problem in two ways:

(1) We prove a generalization of Theorem 2.5.1 involving functions of several arguments, but with recursion only in a *single* argument. This can be done quite easily but we have to be a little careful.

(2) For any fixed m, we define $add_m(n)$ as $add_m(n) = m + n$, that is, we define addition of a *fixed* m to any n. Then, we let $m + n = add_m(n)$.

Since solution (2) involves much less work, we follow it. Let S denote the successor function on $\mathbb{N}$, that is, the function given by

$$S(n) = n^+ = n + 1.$$

Then, using Theorem 2.5.1 with $a = m$ and $g = S$, we get a function, add_m, such that

$$\begin{aligned} add_m(0) &= m, \\ add_m(n+1) &= S(add_m(n)) = add_m(n) + 1 \qquad \text{for all} \quad n \in \mathbb{N}. \end{aligned}$$

Finally, for all $m, n \in \mathbb{N}$, we define $m + n$ by

$$m + n = add_m(n).$$

Now, we have our addition function on $\mathbb{N}$. But this is not the end of the story because we don't know yet that the above definition yields a function having the usual properties of addition, such as

$$\begin{aligned} m + 0 &= m \\ m + n &= n + m \\ (m + n) + p &= m + (n + p). \end{aligned}$$

To prove these properties, of course, we use induction!

We can also define multiplication. Mimicking what we did for addition, define $mult_m(n)$ by recursion as follows;

$$\begin{aligned} mult_m(0) &= 0, \\ mult_m(n+1) &= mult_m(n) + m \qquad \text{for all} \quad n \in \mathbb{N}. \end{aligned}$$

Then, we set

$$m \cdot n = mult_m(n).$$

Note how the recursive definition of $mult_m$ uses the adddition function, $+$, previously defined. Again, to prove the usual properties of multiplication as well as the distributivity of $\cdot$ over $+$, we use induction. Using recursion, we can define many more arithmetic functions. For example, the reader should try defining exponentiation, m^n.

We still haven't defined the usual ordering on the natural numbers but we will do so in the next chapter. Of course, we all know what it is and we will not refrain from using it. Still, it is interesting to give such a definition in our axiomatic framework.

2.6 Inverses of Functions and Relations

Given a function, $f\colon A \to B$ (possibly partial), with $A \neq \emptyset$, suppose there is some function, $g\colon B \to A$ (possibly partial), called a *left inverse of* f, such that

$$g \circ f = \mathrm{id}_A.$$

If such a g exists, we see that f must be total but more is true. Indeed, assume that $f(a) = f(b)$. Then, by applying g, we get

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

However, since $g \circ f = \mathrm{id}_A$, we have $(g \circ f)(a) = \mathrm{id}_A(a) = a$ and $(g \circ f)(b) = \mathrm{id}_A(a) = b$, so we deduce that

$$a = b.$$

Therefore, we showed that if a function, f, with nonempty domain, has a left inverse, then f is total and has the property that for all $a, b \in A$, $f(a) = f(b)$ implies that $a = b$, or equivalently $a \neq b$ implies that $f(a) \neq f(b)$. We say that f is *injective*. As we will see later, injectivity is a very desirable property of functions.

Remark: If $A = \emptyset$, then f is still considered to be injective. In this case, g is the empty partial function (and when $B = \emptyset$, both f and g are the empty function from $\emptyset$ to itself).

Now, suppose there is some function, $h\colon B \to A$ (possibly partial), with $B \neq \emptyset$, called a *right inverse of* f, but this time, we have

$$f \circ h = \mathrm{id}_B.$$

If such an h exists, we see that it must be total but more is true. Indeed, for any $b \in B$, as $f \circ h = \mathrm{id}_B$, we have

$$f(h(b)) = (f \circ h)(b) = \mathrm{id}_B(b) = b.$$

Therefore, we showed that if a function, f, with nonempty codomain has a right inverse, h, then h is total and f has the property that for all $b \in B$, there is some $a \in A$, namely, $a = h(b)$, so that $f(a) = b$. In other words, $\mathrm{Im}\,(f) = B$ or equivalently, every element in B is the image by f of some element of A. We say that f is *surjective*. Again, surjectivity is a very desirable property of functions.

Remark: If $B = \emptyset$, then f is still considered to be surjective but h is not total unless $A = \emptyset$, in which case f is the empty function from $\emptyset$ to itself.

If a function has a left inverse (respectively a right inverse), then it may have more than one left inverse (respectively right inverse).

If a function (possibly partial), $f\colon A \to B$, with $A, B \neq \emptyset$, happens to have both a left inverse, $g\colon B \to A$, and a right inverse, $h\colon B \to A$, then we know that f and h are total. We claim that $g = h$, so that g is total and moreover g is uniquely determined by f.

Lemma 2.6.1 *Let $f\colon A \to B$ be any function and suppose that f has a left inverse, $g\colon B \to A$, and a right inverse, $h\colon B \to A$. Then, $g = h$ and moreover, g is unique, which means that if $g'\colon B \to A$ is any function which is both a left and a right inverse of f, then $g' = g$.*

Proof. Assume that

$$g \circ f = \mathrm{id}_A \quad \text{and} \quad f \circ h = \mathrm{id}_B.$$

Then, we have

$$g = g \circ \mathrm{id}_B = g \circ (f \circ h) = (g \circ f) \circ h = \mathrm{id}_A \circ h = h.$$

Therefore, $g = h$. Now, if g' is any other left inverse of f and h' is any other right inverse of f, the above reasoning applied to g and h' shows that $g = h'$ and the same reasoning applied to g' and h' shows that $g' = h'$. Therefore, $g' = h' = g = h$, that is, g is uniquely determined by f. □

This leads to the following definition.

Definition 2.6.2 A function, $f\colon A \to B$, is said to be *invertible* iff there is a function, $g\colon B \to A$, which is both a left inverse and a right inverse, that is,

$$g \circ f = \mathrm{id}_A \quad \text{and} \quad f \circ g = \mathrm{id}_B.$$

In this case, we know that g is unique and it is denoted f^{-1}.

From the above discussion, if a function is invertible, then it is both injective and surjective. This shows that a function *generally does not have an inverse*. In order to have an inverse a function needs to be injective and surjective, but this fails to be true for many functions. It turns out that if a function is injective and surjective then it has an inverse. We will prove this in the next section.

The notion of inverse can also be defined for relations, but it is a somewhat weaker notion.

Definition 2.6.3 Given any relation, $R \subseteq A \times B$, the *converse* or *inverse* of R is the relation, $R^{-1} \subseteq B \times A$, defined by

$$R^{-1} = \{\langle b, a\rangle \in B \times A \mid \langle a, b\rangle \in R\}.$$

In other words, R^{-1} is obtained by swapping A and B and reversing the orientation of the arrows. Figure 2.5 below shows the inverse of the relation of Figure 2.1:

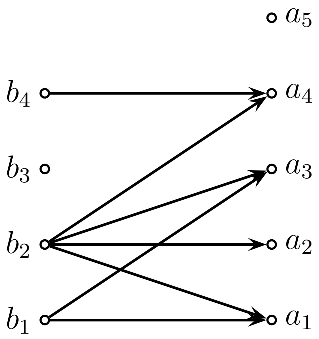

Figure 2.5: The inverse of the relation, R, from Figure 2.1

Now, if R is the graph of a (partial) function, f, beware that R^{-1} is generally *not* the graph of a function at all, because R^{-1} may not be functional. For example, the inverse of the graph G in Figure 2.2 is *not* functional, see below:

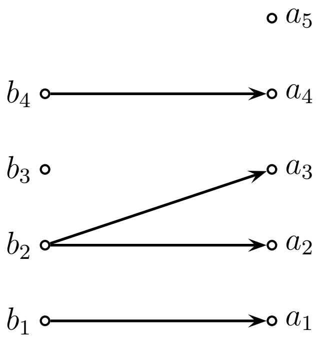

Figure 2.6: The inverse, G^{-1}, of the graph of Figure 2.2

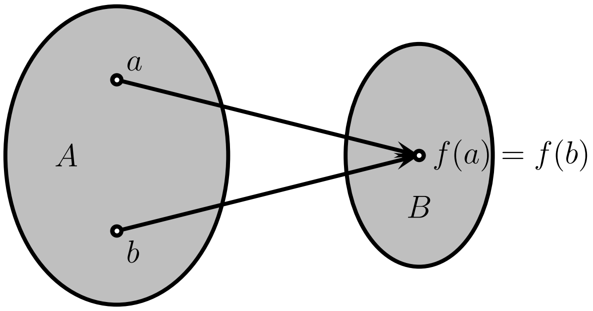

Figure 2.7: A non-injective function

The above example shows that one has to be careful not to view a function as a relation in order to take its inverse. In general, this process does not produce a function. This only works if the function is invertible.

Given any two relations, $R \subseteq A \times B$ and $S \subseteq B \times C$, the reader should prove that

$$(R \circ S)^{-1} = S^{-1} \circ R^{-1}.$$

(Note the switch in the order of composition on the right hand side.) Similarly, if $f \colon A \to B$ and $g \colon B \to C$ are any two invertible functions, then $g \circ f$ is invertible and

$$(g \circ f)^{-1} = f^{-1} \circ g^{-1}.$$

2.7 Injections, Surjections, Bijections, Permutations

We encountered injectivity and surjectivity in Section 2.6. For the record, let us give

Definition 2.7.1 Given any function, $f \colon A \to B$, we say that f *is injective* (or *one-to-one*) iff for all $a, b \in A$, if $f(a) = f(b)$, then $a = b$, or equivalently, if $a \neq b$, then $f(a) \neq f(b)$. We say that f *is surjective* (or *onto*) iff for every $b \in B$, there is some $a \in A$ so that $b = f(a)$, or equivalently if $\mathrm{Im}\,(f) = B$. The function f is *bijective* iff it is both injective and surjective. When $A = B$, a bijection $f \colon A \to A$ is called a *permutation of* A.

Remarks:

1. If $A = \emptyset$, then any function, $f \colon \emptyset \to B$ is (trivially) injective.

2. If $B = \emptyset$, then f is the empty function from $\emptyset$ to itself and it is (trivially) surjective.

3. A function, $f \colon A \to B$, is **not injective** iff **there exist** $a, b \in A$ with $a \neq b$ and **yet** $f(a) = f(b)$, see Figure 2.7.

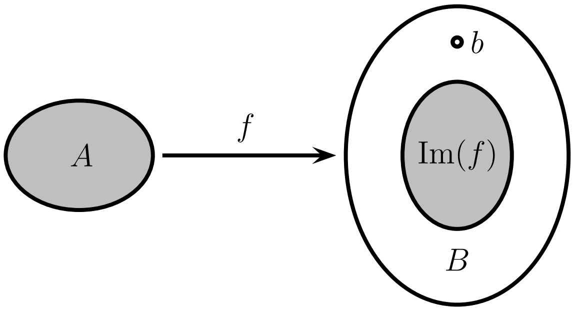

Figure 2.8: A non-surjective function

4. A function, $f\colon A \to B$, is **not surjective** iff **for some** $b \in B$, **there is no** $a \in A$ with $b = f(a)$, see Figure 2.8.

5. Since $\mathrm{Im}\, f = \{b \in B \mid (\exists a \in A)(b = f(a))\}$, a function $f\colon A \to B$ is always surjective onto its image.

6. The notation $f\colon A \hookrightarrow B$ is often used to indicate that a function, $f\colon A \to B$, is an injection.

7. If $A \neq \emptyset$, a function, $f\colon A \to B$, is injective iff for every $b \in B$, there *at most one* $a \in A$ such that $b = f(a)$.

8. If $A \neq \emptyset$, a function, $f\colon A \to B$, is surjective iff for every $b \in B$, there *at least one* $a \in A$ such that $b = f(a)$ iff $f^{-1}(b) \neq \emptyset$ for all $b \in B$.

9. If $A \neq \emptyset$, a function, $f\colon A \to B$, is bijective iff for every $b \in B$, there is *a unique* $a \in A$ such that $b = f(a)$.

10. When A is the finite set $A = \{1, \ldots, n\}$, also denoted $[n]$, it is not hard to show that there are $n!$ permutations of $[n]$.

The function, $f_1\colon \mathbb{Z} \to \mathbb{Z}$, given by $f_1(x) = x + 1$ is injective and surjective. However, the function, $f_2\colon \mathbb{Z} \to \mathbb{Z}$, given by $f_2(x) = x^2$ is neither injective nor surjective (why?). The function, $f_3\colon \mathbb{Z} \to \mathbb{Z}$, given by $f_3(x) = 2x$ is injective but not surjective. The function, $f_4\colon \mathbb{Z} \to \mathbb{Z}$, given by

$$f_4(x) = \begin{cases} k & \text{if } x = 2k \\ k & \text{if } x = 2k + 1 \end{cases}$$

is surjective but not injective.

Remark: The reader should prove that if A and B are finite sets, A has m elements and B has n elements ($m \leq n$) then the set of injections from A to B has

$$\frac{n!}{(n-m)!}$$

elements. The following Theorem relates the notions of injectivity and surjectivity to the existence of left and right inverses.

Theorem 2.7.2 *Let $f\colon A \to B$ be any function and assume $A \neq \emptyset$.*

(a) The function f is injective iff it has a left inverse, g (i.e., a function $g\colon B \to A$ so that $g \circ f = \mathrm{id}_A$).

(b) The function f is surjective iff it has a right inverse, h (i.e., a function $h\colon B \to A$ so that $f \circ h = \mathrm{id}_B$).

(c) The function f is invertible iff it is injective and surjective.

Proof. (a) We already proved in Section 2.6 that the existence of a left inverse implies injectivity. Now, assume f is injective. Then, for every $b \in \mathit{range}(f)$, there is a unique $a_b \in A$ so that $f(a_b) = b$. Since $A \neq \emptyset$, we may pick some a_0 in A. We define $g\colon B \to A$ by

$$g(b) = \begin{cases} a_b & \text{if } b \in \mathit{range}(f) \\ a_0 & \text{if } b \in B - \mathit{range}(f). \end{cases}$$

Then, $g(f(a)) = a$ for all $a \in A$, since $f(a) \in \mathit{range}(f)$ and a is the only element of A so that $f(a) = f(a)$! This shows that $g \circ f = \mathrm{id}_A$, as required.

(b) We already proved in Section 2.6 that the existence of a right inverse implies surjectivity. For the converse, assume that f is surjective. As $A \neq \emptyset$ and f is a function (i.e., f is total), $B \neq \emptyset$. So, for every $b \in B$, the preimage $f^{-1}(b) = \{a \in A \mid f(a) = b\}$ is nonempty. We make a function, $h\colon B \to A$, as follows: For each $b \in B$, pick some element $a_b \in f^{-1}(b)$ (which is nonempty) and let $h(b) = a_b$. By definition of $f^{-1}(b)$, we have $f(a_b) = b$ and so,

$$f(h(b)) = f(a_b) = b, \quad \text{for all } b \in B.$$

This shows that $f \circ h = \mathrm{id}_B$, as required.

(c) If f is invertible, we proved in Section 2.6 that f is injective and surjective. Conversely, if f is both injective and surjective, by (a), the function f has a left inverse g and by (b) it has a right inverse h. However, by Lemma 2.6.1, $g = h$, which shows that f is invertible. □

The alert reader may have noticed a "fast turn" in the proof of the converse in (b). Indeed, we constructed the function h by choosing, for each $b \in B$, some element in $f^{-1}(b)$. How do we justify this procedure from the axioms of set theory?

Well, we can't! For this, we need another (historically somewhat controversial) axiom, the *Axiom of Choice*. This axiom has many equivalent forms. We state the following form which is intuitively quite plausible:

Axiom of Choice (Graph Version).

For every relation, $R \subseteq A \times B$, there is a function, $f\colon A \to B$, with $\mathit{graph}(f) \subseteq R$ and $\mathit{dom}(f) = \mathit{dom}(R)$.

We see immediately that the Axiom of choice justifies the existence of the function h in part (b) of Theorem 2.7.2.

Remarks:

1. Let $f\colon A \to B$ and $g\colon B \to A$ be any two functions and assume that
$$g \circ f = \mathrm{id}_A.$$
Thus, f is a right inverse of g and g is a left inverse of f. So, by Theorem 2.7.2 (a) and (b), we deduce that f is injective and g is surjective. In particular, this shows that any left inverse of an injection is a surjection and that any right inverse of a surjection is an injection.

2. Any right inverse, h, of a surjection, $f\colon A \to B$, is called a *section* of f (which is an abbreviation for *cross-section*). This terminology can be better understood as follows: Since f is surjective, the preimage, $f^{-1}(b) = \{a \in A \mid f(b)\}$ of any element $b \in B$ is nonempty. Moreover, $f^{-1}(b_1) \cap f^{-1}(b_2) = \emptyset$ whenever $b_1 \neq b_2$. Therefore, the pairwise disjoint and nonempty subsets, $f^{-1}(b)$, where $b \in B$, partition A. We can think of A as a big "blob" consisting of the union of the sets $f^{-1}(b)$ (called fibres) and lying over B. The function f maps each fibre, $f^{-1}(b)$ onto the element, $b \in B$. Then, any right inverse, $h\colon B \to A$, of f picks out some element in each fibre, $f^{-1}(b)$, forming a sort of horizontal section of A shown as a curve in Figure 2.9.

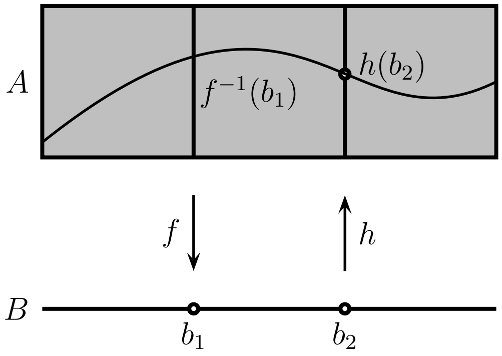

Figure 2.9: A section, h, of a surjective function, f.

3. Any left inverse, g, of an injection, $f\colon A \to B$, is called a *retraction* of f. The terminology reflects the fact that intuitively, as f is injective (thus, g is surjective), B is bigger than A and since $g \circ f = \mathrm{id}_A$, the function g "squeezes" B onto A in such a way that each point $b = f(a)$ in $\mathrm{Im}\, f$ is mapped back to its ancestor $a \in A$. So, B is "retracted" onto A by g.

Before discussing direct and inverse images, we define the notion of restriction and extension of functions.

Definition 2.7.3 Given two functions, $f\colon A \to C$ and $g\colon B \to C$, with $A \subseteq B$, we say that *f is the restriction of g to A* if $graph(f) \subseteq graph(g)$; we write $f = g \restriction A$. In this case, we also say that *g is an extension of f to B*.

2.8 Direct Image and Inverse Image

A function, $f\colon X \to Y$, induces a function from 2^X to 2^Y also denoted f and a function from 2^Y to 2^X, as shown in the following definition:

Definition 2.8.1 Given any function, $f\colon X \to Y$, we define the function $f\colon 2^X \to 2^Y$ so that, for every subset A of X,

$$f(A) = \{y \in Y \mid \exists x \in A,\ y = f(x)\}.$$

The subset, $f(A)$, of Y is called the *direct image of A under f*, for short, the *image of A under f*. We also define the function $f^{-1}\colon 2^Y \to 2^X$ so that, for every subset B of Y,

$$f^{-1}(B) = \{x \in X \mid \exists y \in B,\ y = f(x)\}.$$

The subset, $f^{-1}(B)$, of X is called the *inverse image of A under f* or the *preimage of A under f*.

Remarks:

1. The overloading of notation where f is used both for denoting the original function $f\colon X \to Y$ and the new function $f\colon 2^X \to 2^Y$ may be slightly confusing. If we observe that $f(\{x\}) = \{f(x)\}$, for all $x \in X$, we see that the new f is a natural extension of the old f to the subsets of X and so, using the same symbol f for both functions is quite natural after all. To avoid any confusion, some authors (including Enderton) use a different notation for $f(A)$, for example, $f[\![A]\!]$. We prefer not to introduce more notation and we hope that the context will make it clear which f we are dealing with.

2. The use of the notation f^{-1} for the function $f^{-1}\colon 2^Y \to 2^X$ may even be more confusing, because we know that f^{-1} is generally not a function from Y to X. However, it *is* a function from 2^Y to 2^X. Again, some authors use a different notation for $f^{-1}(B)$, for example, $f^{-1}[\![A]\!]$. Again, we will stick to $f^{-1}(B)$.

3. The set $f(A)$ is sometimes called the *push-forward of A along f* and $f^{-1}(B)$ is sometimes called the *pullback of B along f*.

4. Observe that $f^{-1}(y) = f^{-1}(\{y\})$, where $f^{-1}(y)$ is the preimage defined just after Definition 2.2.3.

5. Although this may seem counter-intuitive, the function f^{-1} has a better behavior than f with respect to union, intersection and complementation.

Some useful properties of $f\colon 2^X \to 2^Y$ and $f^{-1}\colon 2^Y \to 2^X$ are now stated without proof. The proofs are easy and left as exercises.

Proposition 2.8.2 *Given any function, $f\colon X \to Y$, the following properties hold:*

(1) For any $B \subseteq Y$, we have

$$f(f^{-1}(B)) \subseteq B.$$

(2) If $f\colon X \to Y$ is surjective, then

$$f(f^{-1}(B)) = B.$$

(3) For any $A \subseteq X$, we have

$$A \subseteq f^{-1}(f(A)).$$

(4) If $f\colon X \to Y$ is injective, then

$$A = f^{-1}(f(A)).$$

The next proposition deals with the behavior of $f\colon 2^X \to 2^Y$ and $f^{-1}\colon 2^Y \to 2^X$ with respect to union, intersection and complementation.

Proposition 2.8.3 *Given any function, $f\colon X \to Y$, the following properties hold:*

(1) For all $A, B \subseteq X$, we have

$$f(A \cup B) = f(A) \cup f(B).$$

(2)

$$f(A \cap B) \subseteq f(A) \cap f(B).$$

Equality holds if $f\colon X \to Y$ is injective.

(3)

$$f(A) - f(B) \subseteq f(A - B).$$

Equality holds if $f\colon X \to Y$ is injective.

(4) For all $C, D \subseteq Y$, we have

$$f^{-1}(C \cup D) = f^{-1}(C) \cup f^{-1}(D).$$

(5)

$$f^{-1}(C \cap D) = f^{-1}(C) \cap f^{-1}(D).$$

(6)

$$f^{-1}(C - D) = f^{-1}(C) - f^{-1}(D).$$

As we can see from Proposition 2.8.3, the function $f^{-1}\colon 2^Y \to 2^X$ has a better behavior than $f\colon 2^X \to 2^Y$ with respect to union, intersection and complementation.

2.9 Equinumerosity; The Pigeonhole Principle and the Schröder–Bernstein Theorem

The notion of size of a set is fairly intuitive for finite sets but what does it mean for infinite sets? How do we give a precise meaning to the questions:

(a) Do X and Y have the same size?

(b) Does X have more elements than Y?

For finite sets, we can rely on the natural numbers. We count the elements in the two sets and compare the resulting numbers. If one of the two sets is finite and the other is infinite, it seems fair to say that the infinite set has more elements than the finite one.

But what if both sets are infinite?

Remark: A critical reader should object that we have not yet defined what a finite set is (or what an infinite set is). Indeed, we have not! This can be done in terms of the natural numbers but, for the time being, we will rely on intuition. We should also point out that when it comes to infinite sets, experience shows that our intuition fails us miserably. So, we should be very careful.

Let us return to the case where we have two infinite sets. For example, consider $\mathbb{N}$ and the set of even natural numbers, $2\mathbb{N} = \{0, 2, 4, 6, \ldots\}$. Clearly, the second set is properly contained in the first. Does that make $\mathbb{N}$ bigger? On the other hand, the function $n \mapsto 2n$ is a bijection between the two sets, which seems to indicate that they have the same number of elements. Similarly, the set of squares of natural numbers, Squares $= \{0, 1, 4, 9, 16, 25, \ldots\}$ is properly contained in $\mathbb{N}$ and many natural numbers are missing from Squares. But, the map $n \mapsto n^2$ is a bijection between $\mathbb{N}$ and Squares, which seems to indicate that they have the same number of elements.

A more extreme example is provided by $\mathbb{N} \times \mathbb{N}$ and $\mathbb{N}$. Intuitively, $\mathbb{N} \times \mathbb{N}$ is two-dimensional and $\mathbb{N}$ is one-dimensional, so $\mathbb{N}$ seems much smaller than $\mathbb{N} \times \mathbb{N}$. However, it is possible to construct bijections between $\mathbb{N} \times \mathbb{N}$ and $\mathbb{N}$ (try to find one!). In fact, such a function, J, has the graph partially showed below:

$$\begin{array}{ccccccccc}
\vdots & & & & & & & & \\
3 & 6 & \ldots & & & & & & \\
 & & \searrow & & & & & & \\
2 & 3 & & 7 & \ldots & & & & \\
 & & \searrow & & \searrow & & & & \\
1 & 1 & & 4 & & 8 & \ldots & & \\
 & & \searrow & & \searrow & & \searrow & & \\
0 & 0 & & 2 & & 5 & & 9 & \\
 & 0 & & 1 & & 2 & & 3 & \ldots
\end{array}$$

The function J corresponds to a certain way of enumerating pairs of integers. Note that the value of $m+n$ is constant along each diagonal, and consequently, we have

$$\begin{aligned} J(m,n) &= 1+2+\cdots+(m+n)+m, \\ &= ((m+n)(m+n+1)+2m)/2, \\ &= ((m+n)^2+3m+n)/2. \end{aligned}$$

For example, $J(2,1) = ((2+1)^2+3\cdot 2+1)/2 = (9+6+1)/2 = 16/2 = 8$. The function

$$J(m,n) = \frac{1}{2}((m+n)^2+3m+n)$$

is a bijection but that's not so easy to prove!

Perhaps even more surprising, there are bijections between $\mathbb{N}$ and $\mathbb{Q}$. What about between $\mathbb{R}\times\mathbb{R}$ and $\mathbb{R}$? Again, the answer is yes, but that's harder to prove.

These examples suggest that the notion of bijection can be used to define rigorously when two sets have the same size. This leads to the concept of equinumerosity.

Definition 2.9.1 A set A is *equinumerous* to a set B, written $A \approx B$, iff there is a bijection $f\colon A \to B$. We say that A is *dominated* by B, written $A \preceq B$, iff there is an injection from A to B. Finally, we say that A is *strictly dominated* by B, written $A \prec B$, iff $A \preceq B$ and $A \not\approx B$.

Using the above concepts, we can give a precise definition of finiteness. Firstly, recall that for any $n \in \mathbb{N}$, we defined $[n]$ as the set $[n] = \{1,2,\ldots,n\}$, with $[0] = \emptyset$.

Definition 2.9.2 A set, A, is *finite* if it is equinumerous to a set of the form $[n]$, for some $n \in \mathbb{N}$. A set, A, is *infinite* iff it is not finite. We say that A is *countable* (or *denumerable*) iff A is dominated by $\mathbb{N}$.

Two pretty results due to Cantor (1873) are given in the next Theorem. These are among the earliest results of set theory. We assume that the reader is familiar with the fact that every number, $x \in \mathbb{R}$, can be expressed in decimal expansion (possibly infinite). For example,

$$\pi = 3.14159265358979\cdots$$

Theorem 2.9.3 *(Cantor's Theorem) (a) The set, $\mathbb{N}$, is not equinumerous to the set, $\mathbb{R}$, of real numbers.*

(b) For every set, A, there is no surjection from A onto 2^A. Consequently, no set, A, is equinumerous to its power set, 2^A.

Proof. (a) We use a famous proof method due to Cantor and known as a *diagonal argument*. We will prove that if we assume that there is a bijection, $f\colon \mathbb{N} \to \mathbb{R}$, then there is a real number z not belonging to the image of f, contradicting the surjectivity of f. Now, if f exists, we can form a bi-infinite array

$$\begin{aligned} f(0) &= k_0.d_{0\,1}d_{0\,2}d_{0\,3}d_{0\,4}\cdots,\\ f(1) &= k_1.d_{1\,1}d_{1\,2}d_{1\,3}d_{1\,4}\cdots,\\ f(2) &= k_2.d_{2\,1}d_{2\,2}d_{2\,3}d_{2\,4}\cdots,\\ &\vdots\\ f(n) &= k_n.d_{n\,1}d_{n\,2}\cdots d_{n\,n+1}\cdots,\\ &\vdots \end{aligned}$$

where k_n is the integer part of $f(n)$ and the $d_{n\,i}$ are the decimals of $f(n)$, with $i \geq 1$.

The number

$$z = 0.d_1d_2d_3\cdots d_{n+1}\cdots$$

is defined as follows: $d_{n+1} = 1$ if $d_{n\,n+1} \neq 1$, else $d_{n+1} = 2$ if $d_{n\,n+1} = 1$, for every $n \geq 0$, The definition of z shows that

$$d_{n+1} \neq d_{n\,n+1}, \quad \text{for all} \quad n \geq 0,$$

which implies that z is not in the above array, i.e., $z \notin \mathrm{Im}\, f$.

(b) The proof is a variant of Russell's paradox. Assume that there is a surjection, $g\colon A \to 2^A$; we construct a set, $B \subseteq A$, that is not in the image of g, a contradiction. Consider the set

$$B = \{a \in A \mid a \notin g(a)\}.$$

Obviously, $B \subseteq A$. However, for every $a \in A$,

$$a \in B \quad \text{iff} \quad a \notin g(a),$$

which shows that $B \neq g(a)$ for all $a \in A$, i.e., B is not in the image of g. □

As there is an obvious injection of $\mathbb{N}$ into $\mathbb{R}$, Theorem 2.9.3 shows that $\mathbb{N}$ is strictly dominated by $\mathbb{R}$. Also, as we have the injection $a \mapsto \{a\}$ from A into 2^A, we see that every set is strictly dominated by its power set. So, we can form sets as big as we want by repeatedly using the power set operation.

Remark: In fact, $\mathbb{R}$ is equinumerous to $2^{\mathbb{N}}$, but we will not prove this here.

The following proposition shows an interesting connection between the notion of power set and certain sets of functions. To state this proposition, we need the concept of characteristic function of a subset.

Given any set, X, for any subset, A, of X, define the *characteristic function of* A, denoted χ_A, as the function, $\chi_A \colon X \to \{0, 1\}$, given by

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A. \end{cases}$$

In other words, χ_A tests membership in A: For any $x \in X$, $\chi_A(x) = 1$ iff $x \in A$. Observe that we obtain a function, $\chi \colon 2^X \to \{0, 1\}^X$, from the power set of X to the set of characteristic functions from X to $\{0, 1\}$, given by

$$\chi(A) = \chi_A.$$

We also have the function, $\mathcal{S} \colon \{0, 1\}^X \to 2^X$, mapping any characteristic function to the set that it defines and given by

$$\mathcal{S}(f) = \{x \in X \mid f(x) = 1\},$$

for every characteristic function, $f \in \{0, 1\}^X$.

Proposition 2.9.4 *For any set, X, the function $\chi \colon 2^X \to \{0, 1\}^X$ from the power set of X to the set of characteristic functions on X is a bijection whose inverse is $\mathcal{S} \colon \{0, 1\}^X \to 2^X$.*

Proof. Simply check that $\chi \circ \mathcal{S} = \mathrm{id}$ and $\mathcal{S} \circ \chi = \mathrm{id}$, which is straightforward. □

In view of Proposition 2.9.4, there is a bijection between the power set 2^X and the set of functions in $\{0, 1\}^X$. If we write $2 = \{0, 1\}$, then we see that the two sets looks the same! This is the reason why the notation 2^X is often used for the power set (but others prefer $\mathcal{P}(X)$).

There are many other interesting results about equinumerosity. We only mention four more, all very important.

Theorem 2.9.5 *(Pigeonhole Principle) No set of the form $[n]$ is equinumerous to a proper subset of itself, where $n \in \mathbb{N}$,*

Proof. Although the Pigeonhole Principle seems obvious, the proof is not. In fact, the proof requires induction. We advice the reader to skip this proof and come back to it later after we have given more examples of proof by induction.

Suppose we can prove the following Claim:

Claim. Whenever a function, $f \colon [n] \to [n]$, is an injection, then it is a surjection onto $[n]$ (and thus, a bijection).

Observe that the above Claim implies the Pigeonhole Principle. This is proved by contradiction. So, assume there is a function, $f \colon [n] \to [n]$, such that f is injective and $\mathrm{Im}\, f = A \subseteq [n]$ with $A \neq [n]$, i.e., f is a bijection between $[n]$ and A, a proper subset of $[n]$.

Since $f\colon [n] \to [n]$ is injective, by the Claim, we deduce that $f\colon [n] \to [n]$ is surjective, i.e., $\operatorname{Im} f = [n]$, contradicting the fact that $\operatorname{Im} f = A \neq [n]$.

It remains to prove by induction on $n \in \mathbb{N}$ that if $f\colon [n] \to [n]$ is an injection, then it is a surjection (and thus, a bijection). For $n = 0$, f must be the empty function, which is a bijection.

Assume that the induction hypothesis holds for any $n \geq 0$ and consider any injection, $f\colon [n+1] \to [n+1]$. Observe that the restriction of f to $[n]$ is injective.

Case 1. The subset $[n]$ is closed under f, i.e., $f([n]) \subseteq [n]$. Then, we know that $f \upharpoonright [n]$ is injective and by the induction hypothesis, $f([n]) = [n]$. Since f is injective, we must have $f(n+1) = n+1$. Hence, f is surjective, as claimed.

Case 2. The subset $[n]$ is not closed under f, i.e., there is some $p \leq n$ such that $f(p) = n+1$. We can create a new injection, $\widehat{f}$, from $[n+1]$ to itself with the same image as f by interchanging two values of f so that $[n]$ closed under $\widehat{f}$. Define $\widehat{f}$ by

$$\begin{aligned}\widehat{f}(p) &= f(n+1)\\ \widehat{f}(n+1) &= f(p) = n+1\\ \widehat{f}(i) &= f(i), \qquad 1 \leq i \leq n,\ i \neq p.\end{aligned}$$

Then, $\widehat{f}$ is an injection from $[n+1]$ to itself and $[n]$ is closed under $\widehat{f}$. By Case 1, $\widehat{f}$ is surjective, and as $\operatorname{Im} f = \operatorname{Im} \widehat{f}$, we conclude that f is also surjective. □

Corollary 2.9.6 *(Pigeonhole Principle for finite sets) No finite set is equinumerous to a proper subset of itself.*

Proof. To say that a set, A, is finite is to say that there is a bijection, $g\colon A \to [n]$, for some $n \in \mathbb{N}$. Assume that there is a bijection, f, between A and some proper subset of A. Then, consider the function $g \circ f \circ g^{-1}$, from $[n]$ to itself, as shown in the diagram below:

$$\begin{tikzcd} A \arrow[d, "f"'] & {[n]} \arrow[l, "g^{-1}"'] \arrow[d, "g\circ f\circ g^{-1}"] \\ A \arrow[r, "g"'] & {[n]} \end{tikzcd}$$

The rest of proof consists in showing that $[n]$ would be equinumerous to a proper subset of itself, contradicting Theorem 2.9.5. We leave the details as an exercise. □

The pigeonhole principle is often used in the following way: If we have m distinct slots and $n > m$ distinct objects (the pigeons), then when we put all n objects into the m slots, two objects must end up in the same slot. This fact was apparently first stated explicitly by Dirichlet in 1834. As such, it is also known as *Dirichlet's box principle*.

Let A be a finite set. Then, by definition, there is a bijection, $f\colon A \to [n]$, for some $n \in \mathbb{N}$. We claim that such an n is unique. Otherwise, there would be another bijection, $g\colon A \to [p]$, for some $p \in \mathbb{N}$ with $n \neq p$. But now, we would have a bijection $g \circ f^{-1}$ between $[n]$ and $[p]$ with $n \neq p$. This would imply that there is either an injection from $[n]$ to a proper subset of itself or an injection from $[p]$ to a proper subset of itself,[1] contradicting the Pigeonhole Principle.

If A is a finite set, the unique natural number, $n \in \mathbb{N}$, such that $A \approx [n]$ is called the *cardinality of* n and we write $|A| = n$ (or sometimes, $\mathrm{card}(A) = n$).

Remark: The notion of cardinality also makes sense for infinite sets. What happens is that every set is equinumerous to a special kind of set (an initial ordinal) called a *cardinal number* but this topic is beyond the scope of this course. Let us simply mention that the cardinal number of $\mathbb{N}$ is denoted $\aleph_0$ (say "aleph" 0).

Corollary 2.9.7 *(a) Any set equinumerous to a proper subset of itself is infinite.*

(b) The set $\mathbb{N}$ is infinite.

Proof. Left as an exercise to the reader. □

Let us give another application of the pigeonhole principle involving sequences of integers. Given a finite sequence, S, of integers, $a_1, \ldots, a_n$, a *subsequence of* S is a sequence, $b_1, \ldots, b_m$, obtained by deleting elements from the original sequence and keeping the remaining elements in the same order as they originally appeared. More precisely, $b_1, \ldots, b_m$ is a subsequence of $a_1, \ldots, a_n$ if there is an injection, $g\colon \{1, \ldots, m\} \to \{1, \ldots, n\}$, such that $b_i = a_{g(i)}$ for all $i \in \{1, \ldots, m\}$ and $i \leq j$ implies $g(i) \leq g(j)$ for all $i, j \in \{1, \ldots, m\}$. For example, the sequence

$$1 \quad \mathbf{9} \quad 10 \quad \mathbf{8} \quad 3 \quad 7 \quad 5 \quad 2 \quad \mathbf{6} \quad \mathbf{4}$$

contains the subsequence

$$9 \quad 8 \quad 6 \quad 4.$$

An *increasing subsequence* is a subsequence whose elements are in strictly increasing order and a *decreasing subsequence* is a subsequence whose elements are in strictly decreasing order. For example, 9 8 6 4 is a decreasing subsequence of our original sequence. We now prove the following beautiful result due to Erdös and Szekeres:

Theorem 2.9.8 *(Erdös and Szekeres) Let n be any nonzero natural number. Every sequence of $n^2 + 1$ pairwise distinct natural numbers must contain either an increasing subsequence or a decreasing subsequence of length $n + 1$.*

[1] Recall that $n + 1 = \{0, 1, \ldots, n\} = [n] \cup \{0\}$. Here in our argument, we are using the fact that for any two natural numbers n, p, either $n \subseteq p$ or $p \subseteq n$. This fact is indeed true but requires a proof. The proof uses induction and some special properties of the natural numbers implied by the definition of a natural number as a set that belongs to every inductive set. For details, see Enderton [15], Chapter 4.

Proof. The proof proceeds by contradiction. So, assume there is a sequence, S, of n^2+1 pairwise distinct natural numbers so that all increasing or decreasing subsequences of S have length at most n. We assign to every element, s, of the sequence, S, a pair of natural numbers, (u_s, d_s), called a *label*, where u_s, is the length of a longest increasing subsequence of S that starts at s and where d_s is the length of a longest decreasing subsequence of S that starts at s.

Since there are no increasing or descreasing subsequences of length $n+1$ in S, observe that $1 \leq u_s, d_s \leq n$ for all $s \in S$. Therefore,

Claim 1: There are at most n^2 distinct labels (u_s, d_s), where $s \in S$.

We also assert

Claim 2: If s and t are any two distinct elements of S, then $(u_s, d_s) \neq (u_t, d_t)$.

We may assume that s precedes t in S since otherwise, we interchange s and t in the following argument. Since $s \neq t$, there are two cases:

(a) $s < t$. In this case, we know that there is an increasing subsequence of length u_t starting with t. If we insert s in front of this subsequence, we get an increasing subsequence of $u_t + 1$ elements starting at s. Then, as u_s is the maximal length of all increasing subsequences starting with s, we must have $u_t + 1 \leq u_s$, i.e.,

$$u_s > u_t,$$

which implies $(u_s, d_s) \neq (u_t, d_t)$.

(b) $s > t$. This case is similar to case (a), except that we consider a decreasing subsequence of length d_t starting with t. We conclude that

$$d_s > d_t$$

which implies $(u_s, d_s) \neq (u_t, d_t)$.

Therefore, in all cases, we proved that s and t have distinct labels.

Now, by Claim 1, there are only n^2 distinct labels and S has n^2+1 elements so, by the Pigeonhole Principle, two elements of S must have the same label. But, this contradicts Claim 2, which says that distinct elements of S have distinct labels. Therefore, S must have either an increasing subsequence or a decreasing subsequence of length $n+1$, as originally claimed. □

Remark: Note that this proof is not constructive in the sense that it does not produce the desired subsequence; it merely asserts that such a sequence exists.

Our next theorem is the historically famous Schröder-Bernstein Theorem, sometimes called the “Cantor-Bernstein Theorem.” Cantor proved the theorem in 1897 but his proof used a principle equivalent to the axiom of choice. Schröder announced the theorem in an

1896 abstract. His proof, published in 1898, had problems and he published a correction in 1911. The first fully satisfactory proof was given by Felix Bernstein and was published in 1898 in a book by Emile Borel. A shorter proof was given later by Tarski (1955) as a consequence of his fixed point theorem. We postpone giving this proof until the section on lattices (see Section 4.2).

Theorem 2.9.9 *(Schröder-Bernstein Theorem) Given any two sets, A and B, if there is an injection from A to B and an injection from B to A, then there is a bijection between A and B. Equivalently, if $A \preceq B$ and $B \preceq A$, then $A \approx B$.*

The Schröder-Bernstein Theorem is quite a remarkable result and it is a main tool to develop cardinal arithmetic, a subject beyond the scope of this course.

Our third theorem is perhaps the one that is the more surprising from an intuitive point of view. If nothing else, it shows that our intuition about infinity is rather poor.

Theorem 2.9.10 *If A is any infinite set, then $A \times A$ is equinumerous to A.*

Proof. The proof is more involved than any of the proofs given so far and it makes use of the axiom of choice in the form known as *Zorn's Lemma* (see Theorem 4.1.3). For these reasons, we omit the proof and instead refer the reader to Enderton [15] (Chapter 6). $\square$

In particular, Theorem 2.9.10 implies that $\mathbb{R}\times\mathbb{R}$ is in bijection with $\mathbb{R}$. But, geometrically, $\mathbb{R}\times\mathbb{R}$ is a plane and $\mathbb{R}$ is a line and, intuitively, it is surprising that a plane and a line would have "the same number of points." Nevertheless, that's what mathematics tells us!

Our fourth theorem also plays an important role in the theory of cardinal numbers.

Theorem 2.9.11 *(Cardinal comparability) Given any two sets, A and B, either there is an injection from A to B or there is an injection from B to A (that is, either $A \preceq B$ or $B \preceq A$).*

Proof. The proof requires the axiom of choice in a form known as the *Well-Ordering Theorem*, which is also equivalent to Zorn's lemma. For details, see Enderton [15] (Chapters 6 and 7). $\square$

Theorem 2.9.10 implies that there is a bijection between the closed line segment

$$[0,1] = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$$

and the closed unit square

$$[0,1]\times[0,1] = \{(x,y) \in \mathbb{R}^2 \mid 0 \leq x, y \leq 1\}$$

As an interlude, in the next section, we describe a famous space-filling function due to Hilbert. Such a function is obtained as the limit of a sequence of curves that can be defined recursively.

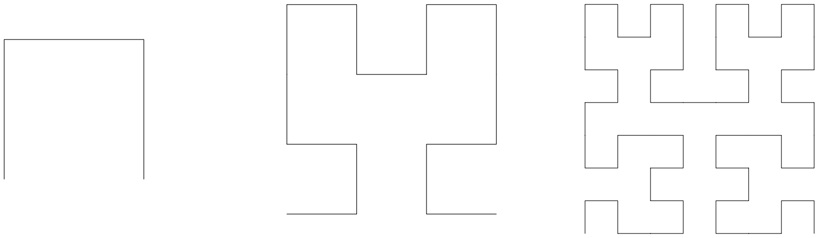

Figure 2.10: A sequence of Hilbert curves h_0, h_1, h_2

2.10 An Amazing Surjection: Hilbert's Space Filling Curve

In the years 1890-1891, Giuseppe Peano and David Hilbert discovered examples of *space filling functions* (also called *space filling curves*). These are surjective functions from the line segment, $[0, 1]$ onto the unit square and thus, their image is the whole unit square! Such functions defy intuition since they seem to contradict our intuition about the notion of dimension, a line segment is one-dimensional, yet the unit square is two-dimensional. They also seem to contradict our intuitive notion of area. Nevertheless, such functions do exist, even continuous ones, although to justify their existence rigouroulsy requires some tools from mathematical analysis. Similar curves were found by others, among which we mention Sierpinski, Moore and Gosper.

We will describe Hilbert's scheme for constructing such a square-filling curve. We define a sequence, (h_n), of polygonal lines, $h_n \colon [0, 1] \to [0, 1] \times [0, 1]$, starting from the simple pattern h_0 (a "square cap" $\sqcap$) shown on the left in Figure 2.10.

The curve h_{n+1} is obtained by scaling down h_n by a factor of $\frac{1}{2}$, and connecting the four copies of this scaled–down version of h_n obtained by rotating by $\pi/2$ (left lower part), rotating by $-\pi/2$ and translating right (right lower part), translating up (left upper part), and translating diagonally (right upper part), as illustrated in Figure 2.10.

It can be shown that the sequence (h_n) converges (uniformly) to a continuous curve $h \colon [0, 1] \to [0, 1] \times [0, 1]$ whose trace is the entire square $[0, 1] \times [0, 1]$. The Hilbert curve h is surjective, continuous, and nowhere differentiable. It also has infinite length! The curve h_5 is shown in Figure 2.11. You should try writing a computer program to plot these curves! By the way, it can be shown that no continuous square-filling function can be injective. It is also possible to define cube-filling curves and even higher-dimensional cube-filling curves! (see some of the web page links in the home page for CIS260)

Before we close this chapter and move on to special kinds of relations, namely, partial orders and equivalence relations, we illustrate how the notion of function can be used to define strings, multisets and indexed families rigorously.

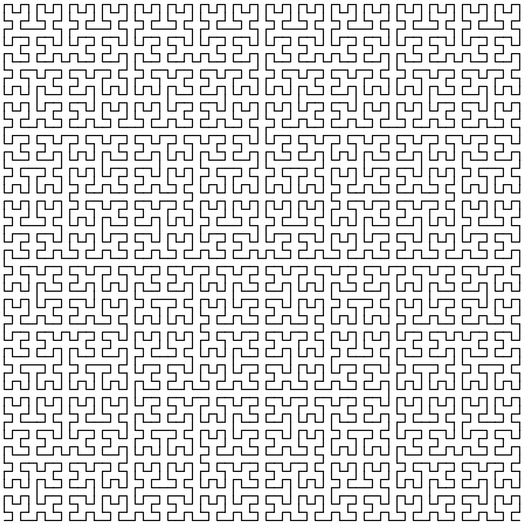

Figure 2.11: The Hilbert curve h_5

2.11 Strings, Multisets, Indexed Families

Strings play an important role in computer science and linguistics because they are the basic tokens that languages are made of. In fact, formal language theory takes the (somewhat crude) view that a language is a set of strings (you will study some formal language theory in CIS262). A string is a finite sequence of letters, for example "Jean", "Val", "Mia", "math", "gaga", "abab". Usually, we have some alphabet in mind and we form strings using letters from this alphabet. Strings are not sets, the order of the letters matters: "abab" and "baba" are different strings. What matters is the position of every letter. In the string "aba", the leftmost "a" is in position 1, "b" is in position 2 and the rightmost "b" is in position 3. All this suggests defining strings as certain kinds of functions whose domains are the sets $[n] = \{1, 2, \ldots, n\}$ (with $[0] = \emptyset$) encountered earlier. Here is the very beginning of the theory of formal languages.

Definition 2.11.1 An *alphabet*, Σ, is any **finite** set.

We often write $\Sigma = \{a_1, \ldots, a_k\}$. The a_i are called the *symbols* of the alphabet.

Remark: There will be a few occasions where we will allow infinite alphabets but normally an alphabet is assumed to be finite.

Examples:

$\Sigma = \{a\}$

$\Sigma = \{a, b, c\}$

$\Sigma = \{0, 1\}$

A string is a finite sequence of symbols. Technically, it is convenient to define strings as functions.

Definition 2.11.2 Given an alphabet, Σ, a *string over* Σ *(or simply a string) of length* n is any function

$$u\colon [n] \to \Sigma.$$

The integer n is the *length* of the string, u, and it is denoted by $|u|$. When $n = 0$, the special string, $u\colon [0] \to \Sigma$, of length 0 is called the *empty string, or null string*, and is denoted by ϵ.

Given a string, $u\colon [n] \to \Sigma$, of length $n \geq 1$, $u(i)$ is the i-th letter in the string u. For simplicity of notation, we denote the string u as

$$u = u_1 u_2 \ldots u_n,$$

with each $u_i \in \Sigma$.

For example, if $\Sigma = \{a, b\}$ and $u\colon [3] \to \Sigma$ is defined such that $u(1) = a$, $u(2) = b$, and $u(3) = a$, we write

$$u = aba.$$

Strings of length 1 are functions $u\colon [1] \to \Sigma$ simply picking some element $u(1) = a_i$ in Σ. Thus, we will identify every symbol $a_i \in \Sigma$ with the corresponding string of length 1.

The set of all strings over an alphabet Σ, including the empty string, is denoted as Σ^*. Observe that when $\Sigma = \emptyset$, then

$$\emptyset^* = \{\epsilon\}.$$

When $\Sigma \neq \emptyset$, the set Σ^* is countably infinite. Later on, we will see ways of ordering and enumerating strings.

Strings can be juxtaposed, or concatenated.

Definition 2.11.3 Given an alphabet, Σ, given two strings, $u\colon [m] \to \Sigma$ and $v\colon [n] \to \Sigma$, the *concatenation,* $u \cdot v$, *(also written* uv*) of* u *and* v is the string
$uv\colon [m + n] \to \Sigma$, defined such that

$$uv(i) = \begin{cases} u(i) & \text{if } 1 \leq i \leq m, \\ v(i - m) & \text{if } m + 1 \leq i \leq m + n. \end{cases}$$

In particular, $u\epsilon = \epsilon u = u$.

It is immediately verified that

$$u(vw) = (uv)w.$$

Thus, concatenation is a binary operation on Σ^* which is associative and has ϵ as an identity. Note that generally, $uv \neq vu$, for example for $u = a$ and $v = b$.

Definition 2.11.4 Given an alphabet Σ, given any two strings $u, v \in \Sigma^*$ we define the following notions as follows:

u is a prefix of v iff there is some $y \in \Sigma^*$ such that

$$v = uy.$$

u is a suffix of v iff there is some $x \in \Sigma^*$ such that

$$v = xu.$$

u is a substring of v iff there are some $x, y \in \Sigma^*$ such that

$$v = xuy.$$

We say that *u is a proper prefix (suffix, substring) of v* iff u is a prefix (suffix, substring) of v and $u \neq v$.

For example, *ga* is a prefix of *gallier*, the string *lier* is a suffix of *gallier* and *all* is a substring of *gallier*

Finally, languages are defined as follows.

Definition 2.11.5 Given an alphabet Σ, a *language over Σ (or simply a language)* is any subset, L, of Σ^*.

The next step would be to introduce various formalisms to define languages, such as automata or grammars but you'll have to take CIS262 to learn about these things!

We now consider multisets. We already encountered multisets in Section 1.2 when we defined the axioms of propositional logic. As for sets, in a multiset, the order of elements does not matter, but as in strings, multiple occurrences of elements matter. For example,

$$\{a, a, b, c, c, c\}$$

is a multiset with two occurrences of a, one occurrence of b and three occurrences of c. This suggests defining a multiset as a function with range $\mathbb{N}$, to specify the multiplicity of each element.

Definition 2.11.6 Given any set, S, a *multiset, M, over S* is any function, $M\colon S \to \mathbb{N}$. A *finite multiset, M, over S* is any function, $M\colon S \to \mathbb{N}$, such that $M(a) \neq 0$ only for finitely many $a \in S$. If $M(a) = k > 0$, we say that *a appears with mutiplicity k in M*.

For example, if $S = \{a, b, c\}$, we may use the notation $\{a, a, a, b, c, c\}$ for the multiset where a has multiplicity 3, b has multiplicity 1, and c has multiplicity 2.

The empty multiset is the function having the constant value 0. The *cardinality* $|M|$ of a (finite) multiset is the number

$$|M| = \sum_{a \in S} M(a).$$

Note that this is well-defined since $M(a) = 0$ for all but finitely many $a \in S$. For example

$$|\{a, a, a, b, c, c\}| = 6.$$

We can define the *union* of multisets as follows: If M_1 and M_2 are two multisets, then $M_1 \cup M_2$ is the multiset given by

$$(M_1 \cup M_2)(a) = M_1(a) + M_2(a), \quad \text{for all} \quad a \in S.$$

A multiset, M_1, is a *submultiset* of a multiset, M_2, if $M_1(a) \leq M_2(a)$, for all $a \in S$. The *difference of* M_1 *and* M_2 is the multiset, $M_1 - M_2$, given by

$$(M_1 - M_2)(a) = \begin{cases} M_1(a) - M_2(a) & \text{if } M_1(a) \geq M_2(a) \\ 0 & \text{if } M_1(a) < M_2(a). \end{cases}$$

Intersection of multisets can also be defined but we will leave this as an exercise.

Let us now discuss indexed families. The Cartesian product construct, $A_1 \times A_2 \times \cdots \times A_n$, allows us to form finite indexed sequences, $\langle a_1, \ldots, a_n \rangle$, but there are situations where we need to have infinite indexed sequences. Typically, we want to be able to consider families of elements indexed by some index set of our choice, say I. We can do this as follows:

Definition 2.11.7 Given any, X, and any other set, I, called the *index set*, the set of I-*indexed families (or sequences) of elements from* X is the set of all functions, $A \colon I \to X$; such functions are usually denoted $A = (A_i)_{i \in I}$. When X is a set of sets, each A_i is some set in X and we call $(A_i)_{i \in I}$ a *family of sets (indexed by* I*).*

Observe that if $I = [n] = \{1, \ldots, n\}$, then an I-indexed family is just a string over X. When $I = \mathbb{N}$, an $\mathbb{N}$-indexed family is called an *infinite sequence* or often just a *sequence*. In this case, we usually write (x_n) for such a sequence ($(x_n)_{n \in \mathbb{N}}$, if we want to be more precise). Also, note that although the notion of indexed family may seem less general than the notion of arbitrary collection of sets, this is an illusion. Indeed, given any collection of sets, X, we may choose the set index set I to be X itself, in wich case X appears as the range of the identity function, $\mathrm{id} \colon X \to X$.

The point of indexed families is that the operations of union and intersection can be generalized in an interesting way. We can also form infinite Cartesian products, which are very useful in algebra and geometry.

Given any indexed family of sets, $(A_i)_{i\in I}$, the *union of the family* $(A_i)_{i\in I}$, denoted $\bigcup_{i\in I} A_i$, is simply the union of the range of A, that is,

$$\bigcup_{i\in I} A_i = \bigcup \mathit{range}(A) = \{a \mid (\exists i \in I),\ a \in A_i\}.$$

Observe that when $I = \emptyset$, the union of the family is the empty set. When $I \neq \emptyset$, we say that we have a *nonempty family* (even though some of the A_i may be empty).

Similarly, if $I \neq \emptyset$, then the *intersection of the family,* $(A_i)_{i\in I}$, denoted $\bigcap_{i\in I} A_i$, is simply the intersection of the range of A, that is,

$$\bigcap_{i\in I} A_i = \bigcap \mathit{range}(A) = \{a \mid (\forall i \in I),\ a \in A_i\}.$$

Unlike the situation for union, when $I = \emptyset$, the intersection of the family does not exist. It would be the set of all sets, which does not exist.

It is easy to see that the laws for union, intersection and complementation generalize to families but we will leave this to the exercises.

An important construct generalizing the notion of finite Cartesian product is the product of families.

Definition 2.11.8 Given any family of sets, $(A_i)_{i\in I}$, the *product of the family* $(A_i)_{i\in I}$, denoted $\prod_{i\in I} A_i$, is the set

$$\prod_{i\in I} A_i = \{a\colon I \to \bigcup_{i\in I} A_i \mid (\forall i \in I),\ a(i) \in A_i\}.$$

Definition 2.11.8 says that the elements of the product $\prod_{i\in I} A_i$ are the functions, $a\colon I \to \bigcup_{i\in I} A_i$, such that $a(i) \in A_i$ for every $i \in I$. We denote the members of $\prod_{i\in I} A_i$ by $(a_i)_{i\in I}$ and we usually call them *I-tuples*. When $I = \{1, \ldots, n\} = [n]$, the members of $\prod_{i\in [n]} A_i$ are the functions whose graph consists of the sets of pairs

$$\{\langle 1, a_1\rangle, \langle 2, a_2\rangle, \ldots, \langle n, a_n\rangle\}, \quad a_i \in A_i,\ 1 \leq i \leq n,$$

and we see that the function

$$\{\langle 1, a_1\rangle, \langle 2, a_2\rangle, \ldots, \langle n, a_n\rangle\} \mapsto \langle a_1, \ldots, a_n\rangle$$

yields a bijection between $\prod_{i\in [n]} A_i$ and the Cartesian product $A_1 \times \cdots \times A_n$. Thus, if each A_i is nonempty, the product $\prod_{i\in [n]} A_i$ is nonempty. But what if I is infinite?

If I is infinite, we smell choice functions. That is, an element of $\prod_{i\in I} A_i$ is obtained by choosing for every $i \in I$ some $a_i \in A_i$. Indeed, the axiom of choice is needed to ensure that

$\prod_{i\in I} A_i \neq \emptyset$ if $A_i \neq \emptyset$ for all $i \in I$! For the record, we state this version (among many!) of the axiom of choice:

Axiom of Choice (Product Version)

For any family of sets, $(A_i)_{i\in I}$, if $I \neq \emptyset$ and $A_i \neq \emptyset$ for all $i \in I$, then $\prod_{i\in I} A_i \neq \emptyset$.

Given the product of a family of sets, $\prod_{i\in I} A_i$, for each $i \in I$, we have the function $pr_i \colon \prod_{i\in I} A_i \to A_i$, called the *ith projection function*, defined by

$$pr_i((a_i)_{i\in I}) = a_i.$$

Chapter 3

Some Counting Problems; Multinomial Coefficients, The Inclusion-Exclusion Principle, Sylvester's Formula, The Sieve Formula

3.1 Counting Permutations and Functions

In this short section, we consider some simple counting problems. Let us begin with permutations. Recall that a *permutation* of a set, A, is any bijection between A and itself. If A is a finite set with n elements, we mentioned earlier (without proof) that A has $n!$ permutations, where the *factorial function*, $n \mapsto n!$ $(n \in \mathbb{N})$, is given recursively by:

$$\begin{aligned} 0! &= 1 \\ (n+1)! &= (n+1)n!. \end{aligned}$$

The reader should check that the existence of the function, $n \mapsto n!$, can be justified using the Recursion Theorem (Theorem 2.5.1).

Proposition 3.1.1 *The number of permutations of a set of n elements is $n!$.*

Proof. We prove that if A and B are any two finite sets of the same cardinality, n, then the number of bijections between A and B is $n!$. Now, in the special case where $B = A$, we get our theorem.

The proof is by induction on n. For $n = 0$, the empty set has one bijection (the empty function). So, there are $0! = 1$ permutations, as desired.

Assume inductively that if A and B are any two finite sets of the same cardinality, n, then the number of bijections between A and B is $n!$. If A and B are sets with $n + 1$

elements, then pick any element, $a \in A$, and write $A = A' \cup \{a\}$, where $A' = A - \{a\}$ has n elements. Now, any bijection, $f \colon A \to B$, must assign some element of B to a and then $f \upharpoonright A'$ is a bijection between A' and $B' = B - \{f(a)\}$. By the induction hypothesis, there are $n!$ bijections between A' and B'. Since there are $n+1$ ways of picking $f(a)$ in B, the total number of bijections between A and B is $(n+1)n! = (n+1)!$, establishing the induction hypothesis. □

Let us also count the number of functions between two finite sets.

Proposition 3.1.2 *If A and B are finite sets with $|A| = m$ and $|B| = n$, then the set of function, B^A, from A to B has n^m elements.*

Proof. We proceed by induction on m. For $m = 0$, we have $A = \emptyset$, and the only function is the empty function. In this case, $n^0 = 1$ and the base case holds.

Assume the induction hypothesis holds for m and assume $|A| = m+1$. Pick any element, $a \in A$, and let $A' = A - \{a\}$, a set with m elements. Any function, $f \colon A \to B$, assigns an element, $f(a) \in B$, to a and $f \upharpoonright A'$ is a function from A' to B. By the induction hypothesis, there are n^m functions from A' to B. Since there are n ways of assigning $f(a) \in B$ to a, there are $n \cdot n^m = n^{m+1}$ functions from A to B, establishing the induction hypothesis. □

As a corollary, we determine the cardinality of a finite power set.

Corollary 3.1.3 *For any finite set, A, if $|A| = n$, then $|2^A| = 2^n$.*

Proof. By proposition 2.9.4, there is a bijection between 2^A and the set of functions $\{0,1\}^A$. Since $|\{0,1\}| = 2$, we get $|2^A| = |\{0,1\}^A| = 2^n$, by Proposition 3.1.2. □

Computing the value of the factorial function for a few inputs, say $n = 1, 2 \ldots, 10$, shows that it grows very fast. For example,

$$10! = 3,628,800.$$

Is it possible to quantify how fast factorial grows compared to other functions, say n^n or e^n? Remarkably, the answer is yes. A beautiful formula due to James Stirling (1692-1770) tells us that

$$n! \cong \sqrt{2\pi n}\left(\frac{n}{e}\right)^n,$$

which means that

$$\lim_{n\to\infty} \frac{n!}{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n} = 1.$$

Here, of course,

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!} + \cdots$$

the base of the natural logarithm. It is even possible to estimate the error. It turns out that

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\lambda_n},$$

where

$$\frac{1}{12n+1} < \lambda_n < \frac{1}{12n},$$

a formula due to Jacques Binet (1786-1856).

Let us introduce some notation used for comparing the rate of growth of functions. We begin with the "Big oh" notation.

Given any two functions, $f\colon \mathbb{N} \to \mathbb{R}$ and $g\colon \mathbb{N} \to \mathbb{R}$, we say that *$f$ is $O(g)$ (or $f(n)$ is $O(g(n))$)* iff there is some $N > 0$ and a constant $c > 0$ such that

$$|f(n)| \leq c|g(n)|, \quad \text{for all} \quad n \geq N.$$

In other words, for n large enough, $|f(n)|$ is bounded by $c|g(n)|$. We sometimes write $n >> 0$ to indicate that n is "large."

For example λ_n is $O(\frac{1}{12n})$. By abuse of notation, we often write $f(n) = O(g(n))$ even though this does not make sense.

The "Big omega" notation means the following: *f is $\Omega(g)$ (or $f(n)$ is $\Omega(g(n))$)* iff there is some $N > 0$ and a constant $c > 0$ such that

$$|f(n)| \geq c|g(n)|, \quad \text{for all} \quad n \geq N.$$

The reader should check that $f(n)$ is $O(g(n))$ iff $g(n)$ is $\Omega(f(n))$.

We can combine O and Ω to get the "Big theta" notation: *f is $\Theta(g)$ (or $f(n)$ is $\Theta(g(n))$)* iff there is some $N > 0$ and some constants $c_1 > 0$ and $c_2 > 0$ such that

$$c_1|g(n)| \leq |f(n)| \leq c_2|g(n)|, \quad \text{for all} \quad n \geq N.$$

Finally, the "Little oh" notation expresses the fact that a function, f, has much slower growth than a function g. We say that *f is $o(g)$ (or $f(n)$ is $o(g(n))$)* iff

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.$$

For example, $\sqrt{n}$ is $o(n)$.

3.2 Counting Subsets of Size k; Binomial and Multinomial Coefficients

Let us now count the number of subsets of cardinality k of a set of cardinality n, with $0 \leq k \leq n$. Denote this number by $\binom{n}{k}$ (say "n choose k"). Actually, in the proposition below, it will be more convenient to assume that $k \in \mathbb{Z}$.

Proposition 3.2.1 *For all $n \in \mathbb{N}$ and all $k \in \mathbb{Z}$, if $\binom{n}{k}$ denotes the number of subsets of cardinality k of a set of cardinality n, then*

$$\begin{aligned}\binom{0}{0} &= 1\\ \binom{n}{k} &= 0 \quad \textit{if} \quad k \notin \{0, 1, \ldots, n\}\\ \binom{n}{k} &= \binom{n-1}{k} + \binom{n-1}{k-1} \quad (n \geq 1,\ 0 \leq k \leq n).\end{aligned}$$

Proof. Obviously, when k is "out of range", that is, when $k \notin \{0, 1, \ldots, n\}$, we have

$$\binom{n}{k} = 0.$$

Next, assume that $0 \leq k \leq n$. Clearly, we may assume that our set is $[n] = \{1, \ldots, n\}$ ($[0] = \emptyset$). If $n = 0$, we have

$$\binom{0}{0} = 1,$$

since the empty set is the only subset of size 0.

If $n \geq 1$, we need to consider the cases $k = 0$ and $k = n$ separately. If $k = 0$, then the only subset of $[n]$ with 0 elements is the empty set, so

$$\binom{n}{0} = 1 = \binom{n-1}{0} + \binom{n-1}{-1} = 1 + 0,$$

since $\binom{n-1}{0} = 1$ and $\binom{n-1}{-1} = 0$. If $k = n$, then the only subset of $[n]$ with n elements is $[n]$ itself, so

$$\binom{n}{n} = 1 = \binom{n-1}{n} + \binom{n-1}{n-1} = 0 + 1,$$

since $\binom{n-1}{n} = 0$ and $\binom{n-1}{n-1} = 1$.

If $1 \leq k \leq n-1$, then there are two kinds of subsets of $\{1, \ldots, n\}$ having k elements: those containing 1, and those not containing 1. Now, there are as many subsets of k elements from $\{1, \ldots, n\}$ containing 1 as there are subsets of $k-1$ elements from $\{2, \ldots, n\}$, namely $\binom{n-1}{k-1}$, and there are as many subsets of k elements from $\{1, \ldots, n\}$ not containing 1 as there are subsets of k elements from $\{2, \ldots, n\}$, namely $\binom{n-1}{k}$. Thus, the number of subsets of $\{1, \ldots, n\}$ consisting of k elements is $\binom{n-1}{k} + \binom{n-1}{k-1}$, which is equal to $\binom{n}{k}$. □

The numbers $\binom{n}{k}$ are also called *binomial coefficients*, because they arise in the expansion of the binomial expression $(a+b)^n$, as we will see shortly. The binomial coefficients can be computed inductively using the formula

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

(sometimes known as *Pascal's recurrence formula*) by forming what is usually called *Pascal's triangle*, which is based on the recurrence for $\binom{n}{k}$:

n	$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$	$\cdots$
0	1								
1	1	1							
2	1	2	1						
3	1	3	3	1					
4	1	4	6	4	1				
5	1	5	10	10	5	1			
6	1	6	15	20	15	6	1		
7	1	7	21	35	35	21	7	1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

We can also give the following explicit formula for $\binom{n}{k}$ in terms of the factorial function:

Proposition 3.2.2 *For all $n, k \in \mathbb{N}$, with $0 \leq k \leq n$, we have*

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Proof. Left as an exercise to the reader (use induction on n and Pascal's recurrence formula).
□

Then, it is very easy to see that

$$\binom{n}{k} = \binom{n}{n-k}.$$

Remark: The binomial coefficients were already known in the twelfth century by the Indian Scholar Bhaskra. Pascal's triangle was taught back in 1265 by the Persian philosopher, Nasir-Ad-Din.

We now prove the "binomial formula" (also called "binomial theorem").

Proposition 3.2.3 *(Binomial Formula) For any two reals $a, b \in \mathbb{R}$ (or more generally, any two commuting variables a, b, i.e., satisfying $ab = ba$), we have the formula:*

$$(a+b)^n = a^n + \binom{n}{1}a^{n-1}b + \binom{n}{2}a^{n-2}b^2 + \cdots + \binom{n}{k}a^{n-k}b^k + \cdots + \binom{n}{n-1}ab^{n-1} + b^n.$$

The above can be written concisely as

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k.$$

Proof. We proceed by induction on n. For $n = 0$, we have $(a+b)^0 = 1$ and the sum on the righthand side is also 1, since $\binom{0}{0} = 1$.

Assume inductively that the formula holds for n. Since

$$(a+b)^{n+1} = (a+b)^n(a+b),$$

using the induction hypothesis, we get

$$\begin{aligned}
(a+b)^{n+1} &= (a+b)^n(a+b)\\
&= \left(\sum_{k=0}^{n}\binom{n}{k}a^{n-k}b^k\right)(a+b)\\
&= \sum_{k=0}^{n}\binom{n}{k}a^{n+1-k}b^k + \sum_{k=0}^{n}\binom{n}{k}a^{n-k}b^{k+1}\\
&= a^{n+1} + \sum_{k=1}^{n}\binom{n}{k}a^{n+1-k}b^k + \sum_{k=0}^{n-1}\binom{n}{k}a^{n-k}b^{k+1} + b^{n+1}\\
&= a^{n+1} + \sum_{k=1}^{n}\binom{n}{k}a^{n+1-k}b^k + \sum_{k=1}^{n}\binom{n}{k-1}a^{n+1-k}b^k + b^{n+1}\\
&= a^{n+1} + \sum_{k=1}^{n}\left(\binom{n}{k} + \binom{n}{k-1}\right)a^{n+1-k}b^k + b^{n+1}\\
&= \sum_{k=0}^{n+1}\binom{n+1}{k}a^{n+1-k}b^k,
\end{aligned}$$

where we used Proposition 3.2.1 to go from the next to the last line to the last line. This establishes the induction step and thus, proves the binomial formula. □

We also stated earlier that the number of injections between a set with m elements and a set with n elements, where $m \leq n$, is given by $\frac{n!}{(n-m)!}$ and we now prove it.

Proposition 3.2.4 *The number of injections between a set, A, with m elements and a set, B, with n elements, where $m \leq n$, is given by $\frac{n!}{(n-m)!} = n(n-1)\cdots(n-m+1)$.*

Proof. We proceed by induction on $m \leq n$. If $m = 0$, then $A = \emptyset$ and there is only one injection, namely the empty function from $\emptyset$ to B. Since $\frac{n!}{(n-0)!} = \frac{n!}{n!} = 1$, the base case holds.

Assume the induction hypothesis holds for m and consider a set, A, with $m+1$ elements, where $m+1 \leq n$. Pick any element $a \in A$ and let $A' = A - \{a\}$, a set with m elements. Any injection, $f\colon A \to B$, assigns some element, $f(a) \in B$, to a and then $f \restriction A'$ is an injection from A' to $B' = B - \{f(a)\}$, a set with $n-1$ elements. By the induction hypothesis, there are

$$\frac{(n-1)!}{(n-1-m)!}$$

injections from A' to B'. Since there are n ways of picking $f(a)$ in B, the number of injections from A to B is

$$n\frac{(n-1)!}{(n-1-m)!} = \frac{n!}{(n-(m+1))!},$$

establishing the induction hypothesis. □

Counting the number of surjections between a set with n elements and a set with p elements, where $n \geq p$, is harder. We state the following formula without giving a proof right now. Finding a proof of this formula is an interesting exercise. We will give a quick proof using the Inclusion-Exclusion Principle in Section 3.3.

Proposition 3.2.5 *The number of surjections, $S_{n\,p}$, between a set, A, with n elements and a set, B, with p elements, where $n \geq p$, is given by*

$$S_{n\,p} = p^n - \binom{p}{1}(p-1)^n + \binom{p}{2}(p-2)^n + \cdots + (-1)^{p-1}\binom{p}{p-1}.$$

Remarks:

1. It can be shown that $S_{n\,p}$ satisfies the following peculiar version of Pascal's recurrence formula:

$$S_{n\,p} = p(S_{n-1\,p} + S_{n-1\,p-1}), \qquad p \geq 2,$$

   and, of course, $S_{n\,1} = 1$ and $S_{n\,p} = 0$ if $p > n$. Using this recurrence formula and the fact that $S_{n\,n} = n!$, simple expressions can be obtained for $S_{n+1\,n}$ and $S_{n+2\,n}$.

2. The numbers, $S_{n\,p}$, are intimately related to the so-called *Strirling numbers of the second kind*, denoted $\left\{ {n \atop p} \right\}$, $S(n,p)$, or $S_n^{(p)}$, which count the number of partitions of a set of n elements into p nonempty pairwise disjoint blocks (see Section 4.5). In fact,

$$S_{n\,p} = p!\left\{ {n \atop p} \right\}.$$

   The Strirling numbers, $\left\{ {n \atop p} \right\}$, satisfy a recurrence equation which is another variant of Pascal's recurrence formula:

$$\begin{aligned} \left\{ {n \atop 1} \right\} &= 1 \\ \left\{ {n \atop n} \right\} &= 1 \\ \left\{ {n \atop p} \right\} &= \left\{ {n-1 \atop p-1} \right\} + p\left\{ {n-1 \atop p} \right\} \qquad (1 \leq p < n). \end{aligned}$$

The total numbers of partitions of a set with $n \geq 1$ elements is given by the *Bell number*,

$$b_n = \sum_{p=1}^{n} \left\{ {n \atop p} \right\}.$$

There is a recurrence formula for the Bell numbers but it is complicated and not very useful because the formula for b_{n+1} involves all the previous Bell numbers.

A good reference for all these special numbers is Graham, Knuth and Patashnik [25], Chapter 6.

The binomial coefficients can be generalized as follows. For all $n, m, k_1, \ldots, k_m \in \mathbb{N}$, with $k_1 + \cdots + k_m = n$ and $m \geq 2$, we have the *multinomial coefficient*,

$$\binom{n}{k_1 \cdots k_m},$$

which counts the number of ways of splitting a set of n elements into an ordered sequence of m disjoint subsets, the ith subset having $k_i \geq 0$ elements. Such sequences of disjoint subsets whose union is $\{1, \ldots, n\}$ itself are sometimes called *ordered partitions*. Beware that some of the subsets in an ordered partition may be empty, so we feel that the terminology "partition" is confusing since as will see in Section 4.5, the subsets that form a partition are never empty. Note that when $m = 2$, the number of ways splitting a set of n elements into two disjoint subsets where the first subset has k_1 elements and the second subset has $k_2 = n - k_1$ elements is precisely the number of subsets of size k_1 of a set of n elements, that is

$$\binom{n}{k_1\, k_2} = \binom{n}{k_1}.$$

Observe that the order of the m subsets matters. For example, for $n = 5$, $m = 4$, $k_1 = 2$ and $k_2 = k_3 = k_4 = 1$, the sequences of subsets $(\{1,2\}, \{3\}, \{4\}, \{5\})$, $(\{1,2\}, \{3\}, \{5\}, \{4\})$, $(\{1,2\}, \{5\}, \{3\}, \{4\})$, $(\{1,2\}, \{4\}, \{3\}, \{5\})$, $(\{1,2\}, \{4\}, \{5\}, \{3\})$, $(\{1,2\}, \{5\}, \{4\}, \{3\})$ are all different and they correspond to the same partition, $\{\{1,2\}, \{3\}, \{4\}, \{5\}\}$.

Proposition 3.2.6 *For all $n, m, k_1, \ldots, k_m \in \mathbb{N}$, with $k_1 + \cdots + k_m = n$ and $m \geq 2$, we have*

$$\binom{n}{k_1 \cdots k_m} = \frac{n!}{k_1! \cdots k_m!}.$$

Proof. There are $\binom{n}{k_1}$ ways of forming a subset of k_1 elements from the set of n elements; there are $\binom{n-k_1}{k_2}$ ways of forming a subset of k_2 elements from the remaining $n - k_1$ elements; there are $\binom{n-k_1-k_2}{k_3}$ ways of forming a subset of k_3 elements from the remaining $n - k_1 - k_2$ elements and so on; finally, there are $\binom{n-k_1-\cdots-k_{m-2}}{k_{m-1}}$ ways of forming a subset of k_{m-1} elements from the remaining $n-k_1-\cdots-k_{m-2}$ elements and there remains a set of $n-k_1-\cdots-k_{m-1}=k_m$ elements. This shows that

$$\binom{n}{k_1\cdots k_m}=\binom{n}{k_1}\binom{n-k_1}{k_2}\cdots\binom{n-k_1-\cdots-k_{m-2}}{k_{m-1}}.$$

But then, using the fact that $k_m=n-k_1-\cdots-k_{m-1}$, we get

$$\begin{aligned}\binom{n}{k_1\cdots k_m} &= \frac{n!}{k_1!(n-k_1)!}\frac{(n-k_1)!}{k_2!(n-k_1-k_2)!}\cdots\frac{(n-k_1-\cdots-k_{m-2})!}{k_{m-1}!(n-k_1-\cdots-k_{m-1})!}\\ &= \frac{n!}{k_1!\cdots k_m!},\end{aligned}$$

as claimed.

As in the binomial case, it is convenient to set

$$\binom{n}{k_1\cdots k_m}=0$$

if $k_i<0$ or $k_i>n$, for any i, with $1\leq i\leq m$. Then, Proposition 3.2.1 is generalized as follows:

Proposition 3.2.7 *For all $n,m,k_1,\ldots,k_m\in\mathbb{N}$, with $k_1+\cdots+k_m=n$, $n\geq 1$ and $m\geq 2$, we have*

$$\binom{n}{k_1\cdots k_m}=\sum_{i=1}^{m}\binom{n-1}{k_1\cdots(k_i-1)\cdots k_m}.$$

Proof. Note that we have $k_i-1=-1$ when $k_i=0$. First, observe that

$$k_i\binom{n}{k_1\cdots k_m}=n\binom{n-1}{k_1\cdots(k_i-1)\cdots k_m}$$

even if $k_i=0$. This is because if $k_i\geq 1$, then

$$\binom{n}{k_1\cdots k_m}=\frac{n!}{k_1!\cdots k_m!}=\frac{n}{k_i}\frac{(n-1)!}{k_1!\cdots(k_i-1)!\cdots k_m!}=\frac{n}{k_i}\binom{n-1}{k_1\cdots(k_i-1)\cdots k_m},$$

and so,

$$k_i\binom{n}{k_1\cdots k_m}=n\binom{n-1}{k_1\cdots(k_i-1)\cdots k_m}.$$

With our convention that $\binom{n-1}{k_1\cdots-1\cdots k_m}=0$, the above identity also holds when $k_i=0$. Then, we have

$$\begin{aligned}\sum_{i=1}^{m}\binom{n-1}{k_1\cdots(k_i-1)\cdots k_m} &= \left(\frac{k_1}{n}+\cdots+\frac{k_m}{n}\right)\binom{n}{k_1\cdots k_m}\\ &= \binom{n}{k_1\cdots k_m},\end{aligned}$$

since $k_1 + \cdots + k_m = n$. $\square$

Remark: Proposition 3.2.7 shows that Pascal's triangle generalizes to "higher dimensions", that is, to $m \geq 3$. Indeed, it is possible to give a geometric interpretation of Proposition 3.2.7 in which the multinomial coefficients corresponding to those $k_1, \ldots, k_m$ with $k_1 + \cdots + k_m = n$ lie on the hyperplane of equation $x_1 + \cdots + x_m = n$ in $\mathbb{R}^m$, and all the multinomial coefficients for which $n \leq N$, for any fixed N, lie in a generalized tetrahedron called a *simplex*. When $m = 3$, the multinomial coefficients for which $n \leq N$ lie in a tetrahedron whose faces are the planes of equations, $x = 0$; $y = 0$; $z = 0$; and $x + y + z = N$.

We have also the following generalization of Proposition 3.2.3:

Proposition 3.2.8 *(Multinomial Formula) For all $n, m \in \mathbb{N}$ with $m \geq 2$, for all pairwise commuting variables $a_1, \ldots, a_m$, we have*

$$(a_1 + \cdots + a_m)^n = \sum_{\substack{k_1, \ldots, k_m \geq 0 \\ k_1 + \cdots + k_m = n}} \binom{n}{k_1 \cdots k_m} a_1^{k_1} \cdots a_m^{k_m}.$$

Proof. We proceed by induction on n and use Proposition 3.2.7. The case $n = 0$ is trivially true.

Assume the induction hypothesis holds for $n \geq 0$, then we have

$$\begin{aligned}
(a_1 + \cdots + a_m)^{n+1} &= (a_1 + \cdots + a_m)^n (a_1 + \cdots + a_m) \\
&= \left(\sum_{\substack{k_1, \ldots, k_m \geq 0 \\ k_1 + \cdots + k_m = n}} \binom{n}{k_1 \cdots k_m} a_1^{k_1} \cdots a_m^{k_m} \right) (a_1 + \cdots + a_m) \\
&= \sum_{i=1}^{m} \sum_{\substack{k_1, \ldots, k_m \geq 0 \\ k_1 + \cdots + k_m = n}} \binom{n}{k_1 \cdots k_i \cdots k_m} a_1^{k_1} \cdots a_i^{k_i + 1} \cdots a_m^{k_m} \\
&= \sum_{i=1}^{m} \sum_{\substack{k_1, \ldots, k_m \geq 0,\, k_i \geq 1 \\ k_1 + \cdots + k_m = n+1}} \binom{n}{k_1 \cdots (k_i - 1) \cdots k_m} a_1^{k_1} \cdots a_i^{k_i} \cdots a_m^{k_m}.
\end{aligned}$$

We seem to hit a snag, namely, that $k_i \geq 1$, but recall that

$$\binom{n}{k_1 \cdots -1 \cdots k_m} = 0,$$

so we have

$$\begin{aligned}
(a_1 + \cdots + a_m)^{n+1} &= \sum_{i=1}^{m} \sum_{\substack{k_1,\ldots,k_m \geq 0,\, k_i \geq 1 \\ k_1+\cdots+k_m = n+1}} \binom{n}{k_1 \cdots (k_i - 1) \cdots k_m} a_1^{k_1} \cdots a_i^{k_i} \cdots a_m^{k_m} \\
&= \sum_{i=1}^{m} \sum_{\substack{k_1,\ldots,k_m \geq 0, \\ k_1+\cdots+k_m = n+1}} \binom{n}{k_1 \cdots (k_i - 1) \cdots k_m} a_1^{k_1} \cdots a_i^{k_i} \cdots a_m^{k_m} \\
&= \sum_{\substack{k_1,\ldots,k_m \geq 0, \\ k_1+\cdots+k_m = n+1}} \left(\sum_{i=1}^{m} \binom{n}{k_1 \cdots (k_i - 1) \cdots k_m} \right) a_1^{k_1} \cdots a_i^{k_i} \cdots a_m^{k_m} \\
&= \sum_{\substack{k_1,\ldots,k_m \geq 0, \\ k_1+\cdots+k_m = n+1}} \binom{n+1}{k_1 \cdots k_i \cdots k_m} a_1^{k_1} \cdots a_i^{k_i} \cdots a_m^{k_m},
\end{aligned}$$

where we used Proposition 3.2.7 to justify the last equation. Therefore, the induction step is proved and so is our proposition. □

How many terms occur on the right-hand side of the multinomial formula? After a moment of reflexion, we see that this is the number of finite multisets of size n whose elements are drawn from a set of m elements, which is also equal to the number of m-tuples, $k_1, \ldots, k_m$, with $k_i \in \mathbb{N}$ and

$$k_1 + \cdots + k_m = n.$$

The following proposition is left an exercise:

Proposition 3.2.9 *The number of finite multisets of size $n \geq 0$ whose elements come from a set of size $m \geq 1$ is*

$$\binom{m+n-1}{n}.$$

3.3 The Inclusion-Exclusion Principle, Sylvester's Formula, The Sieve Formula

We close this chapter with the proof of a poweful formula for determining the cardinality of the union of a finite number of (finite) sets in terms of the cardinalities of the various intersections of these sets. This identity variously attributed Nicholas Bernoulli, de Moivre, Sylvester and Poincaré has many applications to counting problems and to probability theory. We begin with the "baby case" of two finite sets.

Proposition 3.3.1 *Given any two finite sets, A, and B, we have*

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

Proof. This formula is intuitively obvious because if some element, $a \in A \cup B$, belongs to both A and B then it is counted twice in $|A|+|B|$ and so we need to subtract its contribution to $A \cap B$. Now,

$$A \cup B = (A - (A \cap B)) \cup (A \cap B) \cup (B - (A \cap B)),$$

where the three sets on the right-hand side are pairwise disjoint. If we let $a = |A|$, $b = |B|$ and $c = |A \cap B|$, then it is clear that

$$\begin{aligned} |A - (A \cap B)| &= a - c \\ |B - (A \cap B)| &= b - c, \end{aligned}$$

so we get

$$\begin{aligned} |A \cup B| &= |A - (A \cap B)| + |A \cap B| + |B - (A \cap B)| \\ &= a - c + c + b - c = a + b - c \\ &= |A| + |B| - |A \cap B|, \end{aligned}$$

as desired. One can also give a proof by induction on $n = |A \cup B|$. □

We would like to generalize the formula of Proposition 3.3.1 to any finite collection of finite sets, $A_1, \ldots, A_n$. A moment of reflexion shows that when $n = 3$, we have

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

One of the obstacles in generalizing the above formula to n sets is purely notational: We need a way of denoting arbitrary intersections of sets belonging to a family of sets indexed by $\{1, \ldots, n\}$. We can do this by using indices ranging over subsets of $\{1, \ldots, n\}$, as opposed to indices ranging over integers. So, for example, for any nonempty subset, $I \subseteq \{1, \ldots, n\}$, the expression $\bigcap_{i\in I} A_i$ denotes the intersection of all the subsets whose index, i, belongs to I.

Theorem 3.3.2 *(Inclusion-Exclusion Principle) For any finite sequence, $A_1, \ldots, A_n$, of $n \geq 2$ subsets of a finite set, X, we have*

$$\left|\bigcup_{k=1}^{n} A_k\right| = \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ I \neq \emptyset}} (-1)^{(|I|-1)} \left|\bigcap_{i\in I} A_i\right|.$$

Proof. We proceed by induction on $n \geq 2$. The base case, $n = 2$, is exactly Proposition 3.3.1. Let us now consider the induction step. We can write

$$\bigcup_{k=1}^{n+1} A_k = \left(\bigcup_{k=1}^{n} A_k\right) \cup \{A_{n+1}\}$$

and so, by Proposition 3.3.1, we have

$$\begin{aligned}\left|\bigcup_{k=1}^{n+1} A_k\right| &= \left|\left(\bigcup_{k=1}^{n} A_k\right) \cup \{A_{n+1}\}\right| \\ &= \left|\bigcup_{k=1}^{n} A_k\right| + |A_{n+1}| - \left|\left(\bigcup_{k=1}^{n} A_k\right) \cap \{A_{n+1}\}\right|.\end{aligned}$$

We can apply the induction hypothesis to the first term and we get

$$\left|\bigcup_{k=1}^{n} A_k\right| = \sum_{\substack{J\subseteq\{1,\ldots,n\}\\ J\neq\emptyset}} (-1)^{(|J|-1)} \left|\bigcap_{j\in J} A_j\right|.$$

Using distributivity of intersection over union, we have

$$\left(\bigcup_{k=1}^{n} A_k\right) \cap \{A_{n+1}\} = \bigcup_{k=1}^{n} (A_k \cap A_{n+1}).$$

Again, we can apply the induction hypothesis and obtain

$$\begin{aligned}-\left|\bigcup_{k=1}^{n}(A_k \cap A_{n+1})\right| &= -\sum_{\substack{J\subseteq\{1,\ldots,n\}\\ J\neq\emptyset}} (-1)^{(|J|-1)} \left|\bigcap_{j\in J}(A_j \cap A_{n+1})\right| \\ &= \sum_{\substack{J\subseteq\{1,\ldots,n\}\\ J\neq\emptyset}} (-1)^{|J|} \left|\bigcap_{j\in J\cup\{n+1\}} A_j\right| \\ &= \sum_{\substack{J\subseteq\{1,\ldots,n\}\\ J\neq\emptyset}} (-1)^{(|J\cup\{n+1\}|-1)} \left|\bigcap_{j\in J\cup\{n+1\}} A_j\right|.\end{aligned}$$

Putting all this together, we get

$$\begin{aligned}\left|\bigcup_{k=1}^{n+1} A_k\right| &= \sum_{\substack{J\subseteq\{1,\ldots,n\}\\ J\neq\emptyset}} (-1)^{(|J|-1)} \left|\bigcap_{j\in J} A_j\right| + |A_{n+1}| + \sum_{\substack{J\subseteq\{1,\ldots,n\}\\ J\neq\emptyset}} (-1)^{(|J\cup\{n+1\}|-1)} \left|\bigcap_{j\in J\cup\{n+1\}} A_j\right| \\ &= \sum_{\substack{J\subseteq\{1,\ldots,n+1\}\\ J\neq\emptyset,\, n+1\notin J}} (-1)^{(|J|-1)} \left|\bigcap_{j\in J} A_j\right| + \sum_{\substack{J\subseteq\{1,\ldots,n+1\}\\ n+1\in J}} (-1)^{(|J|-1)} \left|\bigcap_{j\in J} A_j\right| \\ &= \sum_{\substack{I\subseteq\{1,\ldots,n+1\}\\ I\neq\emptyset}} (-1)^{(|I|-1)} \left|\bigcap_{i\in I} A_i\right|,\end{aligned}$$

establishing the induction hypothesis and finishing the proof. □

As an application of the Inclusion-Exclusion Principle, let us prove the formula for counting the number of surjections from $\{1, \ldots, n\}$ to $\{1, \ldots, p\}$, with $p \leq n$, given in Proposition 3.2.5.

Recall that the total number of functions from $\{1, \ldots, n\}$ to $\{1, \ldots, p\}$ is p^n. The trick is to count the number of functions that are *not* surjective. Any such function has the property that its image misses one element from $\{1, \ldots, p\}$. So, if we let

$$A_i = \{f \colon \{1, \ldots, n\} \to \{1, \ldots, p\} \mid i \notin \mathrm{Im}\,(f)\},$$

we need to count $|A_1 \cup \cdots \cup A_p|$. But, we can easily do this using the Inclusion-Exclusion Principle. Indeed, for any nonempty subset, I, of $\{1, \ldots, p\}$, with $|I| = k$, the functions in $\bigcap_{i \in I} A_i$ are exactly the functions whose range misses I. But, these are exactly the functions from $\{1, \ldots, n\}$ to $\{1, \ldots, p\} - I$ and there are $(p-k)^n$ such functions. Thus,

$$\left| \bigcap_{i \in I} A_i \right| = (p-k)^n.$$

As there are $\binom{p}{k}$ subsets, $I \subseteq \{1, \ldots, p\}$, with $|I| = k$, the contribution of all k-fold intersections to the Inclusion-Exclusion Principle is

$$\binom{p}{k} (p-k)^n.$$

Note that $A_1 \cap \cdots \cap A_p = \emptyset$, since functions have a nonempty image. Therefore, the Inclusion-Exclusion Principle yields

$$|A_1 \cup \cdots \cup A_p| = \sum_{k=1}^{p-1} (-1)^{k-1} \binom{p}{k} (p-k)^n,$$

and so, the number of surjections, $S_{n\,p}$, is

$$\begin{aligned} S_{n\,p} &= p^n - |A_1 \cup \cdots \cup A_p| = p^n - \sum_{k=1}^{p-1} (-1)^{k-1} \binom{p}{k} (p-k)^n \\ &= \sum_{k=0}^{p-1} (-1)^k \binom{p}{k} (p-k)^n \\ &= p^n - \binom{p}{1} (p-1)^n + \binom{p}{2} (p-2)^n + \cdots + (-1)^{p-1} \binom{p}{p-1}, \end{aligned}$$

which is indeed the formula of Proposition 3.2.5.

Another amusing application of the Inclusion-Exclusion Principle is the formula giving the number, p_n, of permutations of $\{1, \ldots, n\}$ that leave no element fixed (i.e., $f(i) \neq i$, for all $i \in \{1, \ldots, n\}$). Such permutations are often called *derangements*. We get

$$\begin{aligned} p_n &= n!\left(1 - \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{(-1)^k}{k!} + \cdots + \frac{(-1)^n}{n!}\right) \\ &= n! - \binom{n}{1}(n-1)! + \binom{n}{2}(n-2)! + \cdots + (-1)^n. \end{aligned}$$

Remark: We know (using the series expansion for e^x in which we set $x = -1$) that

$$\frac{1}{e} = 1 - \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{(-1)^k}{k!} + \cdots .$$

Consequently, the factor of $n!$ in the above formula for p_n is the sum of the first $n+1$ terms of $\frac{1}{e}$ and so,

$$\lim_{n \to \infty} \frac{p_n}{n!} = \frac{1}{e}.$$

It turns out that the series for $\frac{1}{e}$ converges very rapidly, so $p_n \approx \frac{1}{e}n!$. The ratio $p_n/n!$ has an interesting interpretation in terms of probabilities. Assume n persons go to a restaurant (or to the theatre, etc.) and that they all check their coats. Unfortunately, the cleck loses all the coat tags. Then, $p_n/n!$ is the probability that nobody will get her or his own coat back! As we just explained, this probability is roughly $\frac{1}{e} \approx \frac{1}{3}$, a surprisingly large number.

The Inclusion-Exclusion Principle can be easily generalized in a useful way as follows: Given a finite set, X, let m be any given function, $m\colon X \to \mathbb{R}_+$, and for any nonempty subset, $A \subseteq X$, set

$$m(A) = \sum_{a \in A} m(a),$$

with the convention that $m(\emptyset) = 0$ (Recall that $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$). For any $x \in X$, the number $m(x)$ is called the *weight* (or *measure*) of x and the quantity $m(A)$ is often called the *measure of the set A*. For example, if $m(x) = 1$ for all $x \in A$, then $m(A) = |A|$, the cardinality of A, which is the special case that we have been considering. For any two subsets, $A, B \subseteq X$, it is obvious that

$$\begin{aligned} m(A \cup B) &= m(A) + m(B) \\ m(X - A) &= m(X) - m(A) \\ m(\overline{A \cup B}) &= m(\overline{A} \cap \overline{B}) \\ m(\overline{A \cap B}) &= m(\overline{A} \cup \overline{B}), \end{aligned}$$

where $\overline{A} = X - A$. Then, we have the following version of Theorem 3.3.2:

Theorem 3.3.3 *(Inclusion-Exclusion Principle, Version 2) Given any measure function, $m\colon X \to \mathbb{R}_+$, for any finite sequence, $A_1, \ldots, A_n$, of $n \geq 2$ subsets of a finite set, X, we have*

$$m\left(\bigcup_{k=1}^{n} A_k\right) = \sum_{\substack{I\subseteq\{1,\ldots,n\}\\ I\neq\emptyset}} (-1)^{(|I|-1)}\, m\left(\bigcap_{i\in I} A_i\right).$$

Proof. The proof is obtained from the proof of Theorem 3.3.2 by changing everywhere any expression of the form $|B|$ to $m(B)$. □

A useful corollary of Theorem 3.3.3 often known as Sylvester's formula is:

Theorem 3.3.4 *(Sylvester's Formula) Given any measure, $m\colon X \to \mathbb{R}_+$, for any finite sequence, $A_1, \ldots, A_n$, of $n \geq 2$ subsets of a finite set, X, the measure of the set of elements of X that do not belong to any of the sets A_i is given by*

$$m\left(\bigcap_{k=1}^{n} \overline{A}_k\right) = m(X) + \sum_{\substack{I\subseteq\{1,\ldots,n\}\\ I\neq\emptyset}} (-1)^{|I|}\, m\left(\bigcap_{i\in I} A_i\right).$$

Proof. Observe that

$$\bigcap_{k=1}^{n} \overline{A}_k = X - \bigcup_{k=1}^{n} A_k.$$

Consequently, using Theorem 3.3.3, we get

$$\begin{aligned} m\left(\bigcap_{k=1}^{n} \overline{A}_k\right) &= m\left(X - \bigcup_{k=1}^{n} A_k\right) \\ &= m(X) - m\left(\bigcup_{k=1}^{n} A_k\right) \\ &= m(X) - \sum_{\substack{I\subseteq\{1,\ldots,n\}\\ I\neq\emptyset}} (-1)^{(|I|-1)}\, m\left(\bigcap_{i\in I} A_i\right) \\ &= m(X) + \sum_{\substack{I\subseteq\{1,\ldots,n\}\\ I\neq\emptyset}} (-1)^{|I|}\, m\left(\bigcap_{i\in I} A_i\right), \end{aligned}$$

establishing Sylvester's formula. □

Note that if we use the convention that when the index set, I, is empty then

$$\bigcap_{i\in\emptyset} A_i = X,$$

then the term $m(X)$ can be included in the above sum by removing the condition that $I \neq \emptyset$ and this version of Sylvester's formula is written:

$$m\left(\bigcap_{k=1}^{n} \overline{A}_k\right) = \sum_{I \subseteq \{1,\ldots,n\}} (-1)^{|I|}\, m\left(\bigcap_{i \in I} A_i\right).$$

Sometimes, it is also convenient to regroup terms involving subsets, I, having the same cardinality and another way to state Sylvester's formula is as follows:

$$m\left(\bigcap_{k=1}^{n} \overline{A}_k\right) = \sum_{k=0}^{n} (-1)^k \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ |I|=k}} m\left(\bigcap_{i \in I} A_i\right). \qquad \text{(Sylvester's Formula)}$$

Finally, Sylvester's formula can be generalized to a formula usually known as the "Sieve Formula":

Theorem 3.3.5 *(Sieve Formula) Given any measure, $m\colon X \to \mathbb{R}_+$, for any finite sequence, $A_1, \ldots, A_n$, of $n \geq 2$ subsets of a finite set, X, the measure of the set of elements of X that belong to exactly p of the sets A_i $(0 \leq p \leq n)$ is given by*

$$T_n^p = \sum_{k=p}^{n} (-1)^{k-p} \binom{k}{p} \sum_{\substack{I \subseteq \{1,\ldots,n\} \\ |I|=k}} m\left(\bigcap_{i \in I} A_i\right).$$

Proof. Observe that the set of elements of X that belong to exactly p of the sets A_i (with $0 \leq p \leq n$) is given by the expression

$$\bigcup_{\substack{I \subseteq \{1,\ldots,n\} \\ |I|=p}} \left(\bigcap_{i \in I} A_i \cap \bigcap_{j \notin I} \overline{A}_j\right).$$

For any subset, $I \subseteq \{1, \ldots, n\}$, if we apply Sylvester's formula to $X = \bigcap_{i \in I} A_i$ and to the subsets $A_j \cap \bigcap_{i \in I} A_i$ for which $j \notin I$ (*i.e.*, $j \in \{1, \ldots, n\} - I$), we get

$$m\left(\bigcap_{i \in I} A_i \cap \bigcap_{j \notin I} \overline{A}_j\right) = \sum_{\substack{J \subseteq \{1,\ldots,n\} \\ I \subseteq J}} (-1)^{|J|-|I|}\, m\left(\bigcap_{j \in J} A_j\right).$$

Hence,

$$\begin{aligned}
T_n^p &= \sum_{\substack{I\subseteq\{1,\ldots,n\}\\ |I|=p}} m\left(\bigcap_{i\in I} A_i \cap \bigcap_{j\notin I} \overline{A}_j\right)\\
&= \sum_{\substack{I\subseteq\{1,\ldots,n\}\\ |I|=p}} \sum_{\substack{J\subseteq\{1,\ldots,n\}\\ I\subseteq J}} (-1)^{|J|-|I|}\, m\left(\bigcap_{j\in J} A_j\right)\\
&= \sum_{\substack{J\subseteq\{1,\ldots,n\}\\ |J|\geq p}} \sum_{\substack{I\subseteq J\\ |I|=p}} (-1)^{|J|-|I|}\, m\left(\bigcap_{j\in J} A_j\right)\\
&= \sum_{k=p}^{n} (-1)^{k-p}\binom{k}{p} \sum_{\substack{J\subseteq\{1,\ldots,n\}\\ |J|=k}} m\left(\bigcap_{j\in J} A_j\right),
\end{aligned}$$

establishing the Sieve formula. □

Observe that Sylvester's Formula is the special case of the Sieve Formula for which $p = 0$. The Inclusion-Exclusion Principle (and its relatives) plays an important role in combinatorics and probability theory as the reader will verify by consulting any text on combinatorics. A classical reference on combinatorics is Berge [2]; a more recent is Cameron [8]; a more recent and more advanced is Stanley [40]. Another great (but deceptively tough) reference covering discrete mathematics and including a lot of combinatorics is Graham, Knuth and Patashnik [25]. Conway and Guy [10] is another beautiful book that presents many fascinating and intriguing geometric and combinatorial properties of numbers in a very untertaining manner.

We are now ready to study special kinds of relations: Partial orders and equivalence relations.

Chapter 4

Partial Orders, Lattices, Well Founded Orderings, Equivalence Relations, Distributive Lattices, Boolean Algebras, Heyting Algebras

4.1 Partial Orders

There are two main kinds of relations that play a very important role in mathematics and computer science:

1. Partial orders

2. Equivalence relations.

In this section and the next few ones, we define partial orders and investigate some of their properties. As we will see, the ability to use induction is intimately related to a very special property of partial orders known as well-foundedness.

Intuitively, the notion of order among elements of a set, X, captures the fact some elements are bigger than others, perhaps more important, or perhaps that they carry more information. For example, we are all familiar with the natural ordering, $\leq$, of the integers

$$\cdots, -3 \leq -2 \leq -1 \leq 0 \leq 1 \leq 2 \leq 3 \leq \cdots,$$

the ordering of the rationals (where $\frac{p_1}{q_1} \leq \frac{p_2}{q_2}$ iff $\frac{p_2 q_1 - p_1 q_2}{q_1 q_2} \geq 0$, i.e., $p_2 q_1 - p_1 q_2 \geq 0$ if $q_1 q_2 > 0$ else $p_2 q_1 - p_1 q_2 \leq 0$ if $q_1 q_2 < 0$), and the ordering of the real numbers. In all of the above orderings, note that for any two number a and b, either $a \leq b$ or $b \leq a$. We say that such orderings are *total* orderings.

A natural example of an ordering which is not total is provided by the subset ordering. Given a set, X, we can order the subsets of X by the subset relation: $A \subseteq B$, where A, B

are any subsets of X. For example, if $X = \{a, b, c\}$, we have $\{a\} \subseteq \{a, b\}$. However, note that neither $\{a\}$ is a subset of $\{b, c\}$ nor $\{b, c\}$ is a subset of $\{a\}$. We say that $\{a\}$ and $\{b, c\}$ are *incomparable*. Now, not all relations are partial orders, so which properties characterize partial orders? Our next definition gives us the answer.

Definition 4.1.1 A binary relation, $\leq$, on a set, X, is a *partial order* (or *partial ordering*) iff it is *reflexive*, *transitive* and *antisymmetric*, that is:

(1) (*Reflexivity*): $a \leq a$, for all $a \in X$;

(2) (*Transitivity*): If $a \leq b$ and $b \leq c$, then $a \leq c$, for all $a, b, c \in X$.

(3) (*antisymmetry*): If $a \leq b$ and $b \leq a$, then $a = b$, for all $a, b \in X$.

A partial order is a *total order (ordering)* (or *linear order (ordering)*) iff for all $a, b \in X$, either $a \leq b$ or $b \leq a$. When neither $a \leq b$ nor $b \leq a$, we say that *a and b are incomparable*. A subset, $C \subseteq X$, is a *chain* iff $\leq$ induces a total order on C (so, for all $a, b \in C$, either $a \leq b$ or $b \leq a$). The *strict order (ordering), $<$, associated with* $\leq$ is the relation defined by: $a < b$ iff $a \leq b$ and $a \neq b$. If $\leq$ is a partial order on X, we say that the pair $\langle X, \leq \rangle$ is a *partially ordered set* or for short, a *poset*.

Remark: Observe that if $<$ is the strict order associated with a partial order, $\leq$, then $<$ is transitive and *anti-reflexive*, which means that

(4) $a \not< a$, for all $a \in X$.

Conversely, let $<$ be a relation on X and assume that $<$ is transitive and anti-reflexive. Then, we can define the relation $\leq$ so that $a \leq b$ iff $a = b$ or $a < b$. It is easy to check that $\leq$ is a partial order and that the strict order associated with $\leq$ is our original relation, $<$.

Given a poset, $\langle X, \leq \rangle$, by abuse of notation, we often refer to $\langle X, \leq \rangle$ as the *poset* X, the partial order $\leq$ being implicit. If confusion may arise, for example when we are dealing with several posets, we denote the partial order on X by $\leq_X$.

Here are a few examples of partial orders.

1. **The subset ordering**. We leave it to the reader to check that the subset relation, $\subseteq$, on a set, X, is indeed a partial order. For example, if $A \subseteq B$ and $B \subseteq A$, where $A, B \subseteq X$, then $A = B$, since these assumptions are exactly those needed by the extensionality axiom.

2. **The natural order on** $\mathbb{N}$. Although we all know what is the ordering of the natural numbers, we should realize that if we stick to our axiomatic presentation where we defined the natural numbers as sets that belong to every inductive set (see Definition 1.10.3), then we haven't yet defined this ordering. However, this is easy to do since the natural numbers are sets. For any $m, n \in \mathbb{N}$, define $m \leq n$ as $m = n$ or $m \in n$! Then, it is not hard check that this relation is a total order (Actually, some of the details are a bit tedious and require induction, see Enderton [15], Chapter 4).

3. **Orderings on strings**. Let $\Sigma = \{a_1, \ldots, a_n\}$ be an alphabet. The prefix, suffix and substring relations defined in Section 2.11 are easily seen to be partial orders. However, these orderings are not total. It is sometimes desirable to have a total order on strings and, fortunately, the lexicographic order (also called dictionnary order) achieves this goal. In order to define the *lexicographic order* we assume that the symbols in Σ are totally ordered, $a_1 < a_2 < \cdots < a_n$. Then, given any two strings, $u, v \in \Sigma^*$, we set

$$u \preceq v \quad \begin{cases} \text{if } v = uy, \text{ for some } y \in \Sigma^*, \text{ or} \\ \text{if } u = xa_iy,\ v = xa_jz, \\ \text{and } a_i < a_j, \text{ for some } x, y, z \in \Sigma^*. \end{cases}$$

   In other words, either u is a prefix of v or else u and v share a common prefix, x, and then there is a differring symbol, a_i in u and a_j in v, with $a_i < a_j$. It is fairly tedious to prove that the lexicographic order is a partial order. Moreover, the lexicographic order is a total order.

4. **The divisibility order on** $\mathbb{N}$. Let us begin by defining divisibility in $\mathbb{Z}$. Given any two integers, $a, b \in \mathbb{Z}$, with $b \neq 0$, we say that b *divides* a (a *is a multiple of* b) iff $a = bq$ for some $q \in \mathbb{Z}$. Such a q is called the *quotient of* a *and* b. Most number theory books use the notation $b \mid a$ to express that b divides a. We leave the verification that the divisibility relation is reflexive and transitive as an easy exercise. What about antisymmetry? So, assume that $b \mid a$ and $a \mid b$ (thus, $a, b \neq 0$). This means that there exist $q_1, q_2 \in \mathbb{Z}$ so that

$$a = bq_1 \quad \text{and} \quad b = aq_2.$$

   From the above, we deduce that $b = bq_1q_2$, that is

$$b(1 - q_1q_2) = 0.$$

   As $b \neq 0$, we conclude that

$$q_1q_2 = 1.$$

   Now, let us restrict ourselves to $\mathbb{N}_+ = \mathbb{N} - \{0\}$, so that $a, b \geq 1$. It follows that $q_1, q_2 \in \mathbb{N}$ and in this case, $q_1q_2 = 1$ is only possible iff $q_1 = q_2 = 1$. Therefore, $a = b$ and the divisibility relation is indeed a partial order on $\mathbb{N}_+$. Why is divisibility not a partial order on $\mathbb{Z} - \{0\}$?

Given a poset, $\langle X \leq \rangle$, if X is finite, then there is a convenient way to describe the partial order $\leq$ on X using a graph. In preparation for that, we need a few preliminary notions.

Consider an arbitrary poset, $\langle X \leq \rangle$ (not necessarily finite). Given any element, $a \in X$, the following situations are of interest:

1. For **no** $b \in X$ do we have $b < a$. We say that a is a *minimal element* (of X).

2. There is some $b \in X$ so that $b < a$ and there is **no** $c \in X$ so that $b < c < a$. We say that b is an *immediate predecessor of* a.

3. For **no** $b \in X$ do we have $a < b$. We say that a is a *maximal element* (of X).

4. There is some $b \in X$ so that $a < b$ and there is **no** $c \in X$ so that $a < c < b$. We say that b is an *immediate successor of* a.

Note that an element may have more than one immediate predecessor (or more than one immediate successor).

If X is a finite set, then it is easy to see that every element that is not minimal has an immediate predecessor and any element that is not maximal has an immediate successor (why?). But if X is infinite, for example, $X = \mathbb{Q}$, this may not be the case. Indeed, given any two distinct rational numbers, $a, b \in \mathbb{Q}$, we have

$$a < \frac{a+b}{2} < b.$$

Let us now use our notion of immediate predecessor to draw a diagram representing a finite poset, $\langle X, \leq \rangle$. The trick is to draw a picture consisting of nodes and oriented edges, where the nodes are all the elements of X and where we draw an oriented edge from a to b iff a is an immediate predecessor of b. Such a diagram is called a *Hasse diagram* for $\langle X, \leq \rangle$. Observe that if $a < c < b$, then the diagram does **not** have edges corresponding to the relations $a < c$ and $c < b$. However, such information can be recovered from the diagram by following paths consisting of one or several consecutive edges (we are a bit informal here, but we will define directed graphs and paths more rigorously later). Similarly, the self-loops corresponding to the the reflexive relations $a \leq a$ are omitted. A Hasse diagram is an economical representation of a finite poset and it contains the same amount of information as the partial order, $\leq$.

Here is the diagram associated with the partial order on the power set of the two element set, $\{a, b\}$:

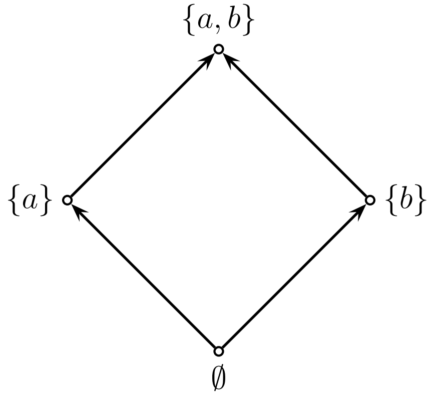

Figure 4.1: The partial order of the power set $2^{\{a,b\}}$

Here is the diagram associated with the partial order on the power set of the three element set, $\{a, b, c\}$:

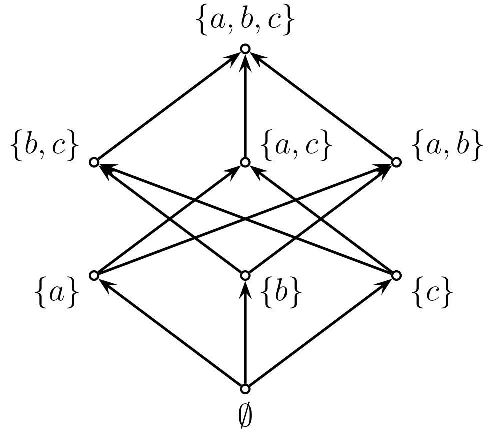

Figure 4.2: The partial order of the power set $2^{\{a,b,c\}}$

Note that $\emptyset$ is a minimal element of the above poset (in fact, the smallest element) and $\{a, b, c\}$ is a maximal element (in fact, the greatest element). In the above example, there is a unique minimal (resp. maximal) element. A less trivial example with multiple minimal and maximal elements is obtained by deleting $\emptyset$ and $\{a, b, c\}$:

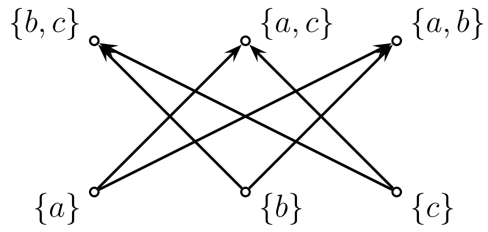

Figure 4.3: Minimal and maximal elements in a poset

Given a poset, $\langle X, \leq \rangle$, observe that if there is some element $m \in X$ so that $m \leq x$ for all $x \in X$, then m is unique. Indeed, if m' is another element so that $m' \leq x$ for all $x \in X$, then if we set $x = m'$ in the first case, we get $m \leq m'$ and if we set $x = m$ in the second case, we get $m' \leq m$, from which we deduce that $m = m'$, as claimed. Such an element, m, is called the *smallest* or the *least element* of X. Similarly, an element, $b \in X$, so that $x \leq b$ for all $x \in X$ is unique and is called the *greatest element* of X.

We summarize some of our previous definitions and introduce a few more useful concepts in

Definition 4.1.2 Let $\langle X, \leq \rangle$ be a poset and let $A \subseteq X$ be any subset of X. An element, $b \in X$, is a *lower bound of* A iff $b \leq a$ for all $a \in A$. An element, $m \in X$, is an *upper bound of* A iff $a \leq m$ for all $a \in A$. An element, $b \in X$, is the *least element of* A iff $b \in A$ and $b \leq a$ for all $a \in A$. An element, $m \in X$, is the *greatest element of* A iff $m \in A$ and $a \leq m$ for all $a \in A$. An element, $b \in A$, is *minimal in* A iff $a < b$ for no $a \in A$, or equivalently, if for all $a \in A$, $a \leq b$ implies that $a = b$. An element, $m \in A$, is *maximal in* A iff $m < a$ for no $a \in A$, or equivalently, if for all $a \in A$, $m \leq a$ implies that $a = m$. An element, $b \in X$, is the *greatest lower bound of* A iff the set of lower bounds of A is nonempty and if b is the greatest element of this set. An element, $m \in X$, is the *least upper bound of* A iff the set of upper bounds of A is nonempty and if m is the least element of this set.

Remarks:

1. If b is a lower bound of A (or m is an upper bound of A), then b (or m) may not belong to A.

2. The least element of A is a lower bound of A that also belongs to A and the greatest element of A is an upper bound of A that also belongs to A. When $A = X$, the least element is often denoted $\perp$, sometimes 0, and the greatest element is often denoted $\top$, sometimes 1.

3. Minimal or maximal elements of A belong to A but they are not necessarily unique.

4. The greatest lower bound (or the least upper bound) of A may not belong to A. We use the notation $\bigwedge A$ for the greatest lower bound of A and the notation $\bigvee A$ for the least upper bound of A. In computer science, some people also use $\bigsqcup A$ instead of $\bigvee A$ and the symbol $\bigsqcup$ upside down instead of $\bigwedge$. When $A = \{a, b\}$, we write $a \wedge b$ for $\bigwedge\{a, b\}$ and $a \vee b$ for $\bigvee\{a, b\}$. The element $a \wedge b$ is called the *meet of* a *and* b and $a \vee b$ is the *join of* a *and* b. (Some computer scientists use $a \sqcap b$ for $a \wedge b$ and $a \sqcup b$ for $a \vee b$.)

5. Observe that if it exists, $\bigwedge \emptyset = \top$, the greatest element of X and if its exists, $\bigvee \emptyset = \perp$, the least element of X. Also, if it exists, $\bigwedge X = \perp$ and if it exists, $\bigvee X = \top$.

The reader should look at the posets in Figures 4.2 and 4.3 for examples of the above notions.

For the sake of completeness, we state the following fundamental result known as Zorn's Lemma even though it is unlikely that we will use it in this course. Zorn's lemma turns out to be equivalent to the axiom of choice. For details and a proof, the reader is referred to Suppes [41] or Enderton [15].

Theorem 4.1.3 *(Zorn's Lemma) Given a poset, $\langle X, \leq \rangle$, if every nonempty chain in X has an upper-bound, then X has some maximal element.*

When we deal with posets, it is useful to use functions that are order-preserving as defined next.

Definition 4.1.4 Given two posets $\langle X, \leq_X\rangle$ and $\langle Y, \leq_Y\rangle$, a function, $f\colon X \to Y$, is *monotonic* (or *order-preserving*) iff for all $a, b \in X$,

$$\text{if} \quad a \leq_X b \quad \text{then} \quad f(a) \leq_Y f(b).$$

4.2 Lattices and Tarski's Fixed Point Theorem

We now take a closer look at posets having the property that every two elements have a meet and a join (a greatest lower bound and a least upper bound). Such posets occur a lot more than we think. A typical example is the power set under inclusion, where meet is intersection and join is union.

Definition 4.2.1 A *lattice* is a poset in which any two elements have a meet and a join. A *complete lattice* is a poset in which any subset has a greatest lower bound and a least upper bound.

According to part (5) of the remark just before Zorn's Lemma, observe that a complete lattice must have a least element, $\perp$, and a greatest element, $\top$.

Remark: The notion of complete lattice is due to G. Birkhoff (1933). The notion of a lattice is due to Dedekind (1897) but his definition used properties (L1)-(L4) listed in Proposition 4.2.2. The use of meet and join in posets was first studied by C. S. Peirce (1880).

Figure 4.4 shows the lattice structure of the power set of $\{a, b, c\}$. It is actually a complete lattice.

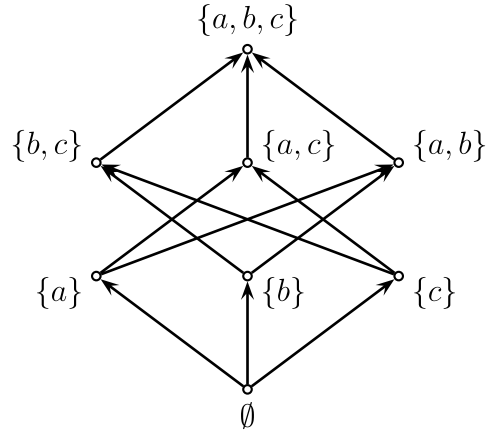

Figure 4.4: The lattice $2^{\{a,b,c\}}$

It is easy to show that any finite lattice is a complete lattice and that a finite poset is a lattice iff it has a least element and a greatest element.

The poset $\mathbb{N}_+$ under the divisibility ordering is a lattice! Indeed, it turns out that the meet operation corresponds to *greatest common divisor* and the join operation corresponds to *least common multiple*. However, it is not a complete lattice. The power set of any set, X, is a complete lattice under the subset ordering. Indeed, one will verify immediately that for any collection, $\mathcal{C}$, of subsets of X, the least upper bound of $\mathcal{C}$ is its union, $\bigcup \mathcal{C}$, and the greatest lower bound of $\mathcal{C}$ is its intersection, $\bigcap \mathcal{C}$. The least element of 2^X is $\emptyset$ and its greatest element is X itself.

The following proposition gathers some useful properties of meet and join.

Proposition 4.2.2 *If X is a lattice, then the following identities hold for all $a, b, c \in X$:*

$$\begin{array}{lll} L1 & a \vee b = b \vee a, & a \wedge b = b \wedge a \\ L2 & (a \vee b) \vee c = a \vee (b \vee c), & (a \wedge b) \wedge c = a \wedge (b \wedge c) \\ L3 & a \vee a = a, & a \wedge a = a \\ L4 & (a \vee b) \wedge a = a, & (a \wedge b) \vee a = a. \end{array}$$

Properties (L1) correspond to commutativity, properties (L2) to associativity, properties (L3) to idempotence and properties (L4) to absorption. Furthermore, for all $a, b \in X$, we have

$$a \leq b \quad \textit{iff} \quad a \vee b = b \quad \textit{iff} \quad a \wedge b = a,$$

called consistency.

Proof. The proof is left as an exercise to the reader. □

Properties (L1)-(L4) are algebraic properties that were found by Dedekind (1897). A pretty symmetry reveals itself in these identities: they all come in pairs, one involving $\wedge$, the other involving $\vee$. A useful consequence of this symmetry is *duality*, namely, that each equation derivable from (L1)-(L4) has a dual statement obtained by exchanging the symbols $\wedge$ and $\vee$. What is even more interesting is that it is possible to use these properties to define lattices. Indeed, if X is a set together with two operations, $\wedge$ and $\vee$, satisfying (L1)-(L4), we can define the relation $a \leq b$ by $a \vee b = b$ and then show that $\leq$ is a partial order such that $\wedge$ and $\vee$ are the corresponding meet and join. The first step is to show that

$$a \vee b = b \quad \text{iff} \quad a \wedge b = a.$$

If $a \vee b = b$, then substituting b for $a \vee b$ in (L4), namely

$$(a \vee b) \wedge a = a,$$

we get

$$b \wedge a = a,$$

which, by (L1), yields

$$a \wedge b = a,$$

as desired. Conversely, if $a \wedge b = a$, then by (L1) we have $b \wedge a = a$, and substituting a for $b \wedge a$ in the instance of (L4) where a and b are switched, namely

$$(b \wedge a) \vee b = b,$$

we get

$$a \vee b = b,$$

as claimed. Therefore, we can define $a \leq b$ as $a \vee b = b$ or equivalently as $a \wedge b = a$. After a little work, we obtain

Proposition 4.2.3 *Let X be a set together with two operations $\wedge$ and $\vee$ satisfying the axioms (L1)-(L4) of proposition 4.2.2. If we define the relation $\leq$ by $a \leq b$ iff $a \vee b = b$ (equivalently, $a \wedge b = a$), then $\leq$ is a partial order and $(X, \leq)$ is a lattice whose meet and join agree with the original operations $\wedge$ and $\vee$.*

The following proposition shows that the existence of arbitrary least upper bounds (or arbitrary greatest lower bounds) is already enough ensure that a poset is a complete lattice.

Proposition 4.2.4 *Let $\langle X, \leq \rangle$ be a poset. If X has a greatest element, $\top$, and if every nonempty subset, A, of X has a greatest lower bound, $\bigwedge A$, then X is a complete lattice. Dually, if X has a least element, $\perp$, and if every nonempty subset, A, of X has a least upper bound, $\bigvee A$, then X is a complete lattice*

Proof. Assume X has a greatest element, $\top$, and that every nonempty subset, A, of X has a greatest lower bound, $\bigwedge A$. We need to show that any subset, S, of X has a least upper bound. As X has a greatest element, $\top$, the set, U, of upper bounds of S is nonempty and so, $m = \bigwedge U$ exists. We claim that $\bigwedge U = \bigvee S$, i.e, m is the least upper bound of S. First, note that every element of S is a lower bound of U since U is the set of upper bounds of S. As $m = \bigwedge U$ is the greatest lower bound of U, we deduce that $s \leq m$ for all $s \in S$, i.e., m is an upper bound of S. Next, if b is any upper bound for S, then $b \in U$ and as m is a lower bound of U (the greatest one), we have $m \leq b$, i.e., m is the least upper bound of S. The other statement is proved by duality. $\square$

We are now going to prove a remarkable result due to A. Tarski (discovered in 1942, published in 1955). A special case (for power sets) was proved by B. Knaster (1928). First, we define fixed points.

Definition 4.2.5 Let $\langle X, \leq \rangle$ be a poset and let $f\colon X \to X$ be a function. An element, $x \in X$, is a *fixed point of f* (sometimes spelled *fixpoint*) iff

$$f(x) = x.$$

An element, $x \in X$, is a *least (resp. greatest) fixed point of f* if it is a fixed point of f and if $x \leq y$ (resp. $y \leq x$) for every fixed point y of f.

Fixed points play an important role in certain areas of mathematics (for example, topology, differential equations) and also in economics because they tend to capture the notion of stability or equilibrium.

We now prove the following pretty theorem due to Tarski and then immediately proceed to use it to give a very short proof of the Schröder-Bernstein Theorem (Theorem 2.9.9).

Theorem 4.2.6 *(Tarski's Fixed Point Theorem) Let $\langle X, \leq\rangle$ be a complete lattice and let $f\colon X \to X$ be any monotonic function. Then, the set, F, of fixed points of f is a complete lattice. In particular, f has a least fixed point,*

$$x_{\min} = \bigwedge\{x \in X \mid f(x) \leq x\}$$

and a greatest fixed point

$$x_{\max} = \bigvee\{x \in X \mid x \leq f(x)\}.$$

Proof. We proceed in three steps.

Step 1. We prove that $x_{\max}$ is the largest fixed point of f.

Since $x_{\max}$ is an upper bound of $A = \{x \in X \mid x \leq f(x)\}$ (the smallest one), we have $x \leq x_{\max}$ for all $x \in A$. By monotonicity of f, we get $f(x) \leq f(x_{\max})$ and since $x \in A$, we deduce

$$x \leq f(x) \leq f(x_{\max}) \quad \text{for all} \quad x \in A,$$

which shows that $f(x_{\max})$ is an upper bound of A. As $x_{\max}$ is the least upper bound of A, we get

$$x_{\max} \leq f(x_{\max}). \tag{$*$}$$

Again, by monotonicity, from the above inequality, we get

$$f(x_{\max}) \leq f(f(x_{\max})),$$

which shows that $f(x_{\max}) \in A$. As $x_{\max}$ is an upper bound of A, we deduce that

$$f(x_{\max}) \leq x_{\max}. \tag{$**$}$$

But then, $(*)$ and $(**)$ yield

$$f(x_{\max}) = x_{\max},$$

which shows that $x_{\max}$ is a fixed point of f. If x is any fixed point of f, that is, if $f(x) = x$, we also have $x \leq f(x)$, i.e., $x \in A$. As $x_{\max}$ is the least upper bound of A, we have $x \leq x_{\max}$, which proves that $x_{\max}$ is the greatest fixed point of f.

Step 2. We prove that $x_{\min}$ is the least fixed point of f.

This proof is dual to the proof given in Step 1.

Step 3. We know that the set of fixed points, F, of f has a least element and a greatest element, so by Proposition 4.2.4, it is enough to prove that any nonempty subset, $S \subseteq F$, has a greatest lower bound. If we let

$$I = \{x \in X \mid x \leq s \quad \text{for all} \quad s \in S \quad \text{and} \quad x \leq f(x)\},$$

then we claim that $a = \bigvee I$ is a fixed point of f and that it is the greatest lower bound of S.

The proof that $a = \bigvee I$ is a fixed point of f is analogous to the proof used in Step 1. Since a is an upper bound of I, we have $x \leq a$ for all $x \in I$. By monotonicity of f and the fact that $x \in I$, we get

$$x \leq f(x) \leq f(a).$$

Thus, $f(a)$ is an upper bound of I and so, as a is the least upper bound of I, we have

$$a \leq f(a). \tag{†}$$

By monotonicity of f, we get $f(a) \leq f(f(a))$. Now, to claim that $f(a) \in I$, we need to check that $f(a)$ is a lower bound of S. However, by definition of I, every element of S is an upper bound of I and since a is the least upper bound of I, we must have $a \leq s$ for all $s \in S$ i.e., a is a lower bound of S. By monotonicity of f and the fact that S is a set of fixed points, we get

$$f(a) \leq f(s) = s, \quad \text{for all} \quad \in S,$$

which shows that $f(a)$ is a lower bound of S and thus, $f(a) \in I$, as contended. As a is an upper bound of I and $f(a) \in I$, we must have

$$f(a) \leq a, \tag{††}$$

and together with (†), we conclude that $f(a) = a$, i.e., a is a fixed point of f.

Since we already proved that a is a lower bound of S it only remains to show that if x is any fixed point of f and x is a lower bound of S, then $x \leq a$. But, if x is any fixed point of f then $x \leq f(x)$ and since x is also a lower bound of S, then $x \in I$. As a is an upper bound of I, we do get $x \leq a$. □

It should be noted that the least upper bounds and the greatest lower bounds in F do not necessarily agree with those in X. In technical terms, F is generally not a sublattice of X.

Now, as promised, we use Tarski's Fixed Point Theorem to prove the Schröder-Bernstein Theorem.

Theorem 2.9.9 *Given any two sets, A and B, if there is an injection from A to B and an injection from B to A, then there is a bijection between A and B.*

Proof. Let $f\colon A \to B$ and $g\colon B \to A$ be two injections. We define the function, $\varphi\colon 2^A \to 2^A$, by

$$\varphi(S) = A - g(B - f(S)),$$

for any $S \subseteq A$. Because of the two complementations, it is easy to check that φ is monotonic (ckeck it). As 2^A is a complete lattice, by Tarski's fixed point theorem, the function φ has a fixed point, that is, there is some subset $C \subseteq A$ so that

$$C = A - g(B - f(C)).$$

By taking the complement of C in A, we get

$$A - C = g(B - f(C)).$$

Now, as f and g are injections, the restricted functions $f \restriction C \colon C \to f(C)$ and $g \restriction (B - f(C)) \colon (B - f(C)) \to (A - C)$ are bijections. Using these functions, we define the function, $h \colon A \to B$, as follows:

$$h(a) = \begin{cases} f(a) & \text{if } a \in C \\ (g \restriction (B - f(C))^{-1}(a) & \text{if } a \notin C. \end{cases}$$

The reader will check that h is indeed a bijection. □

The above proof is probably the shortest known proof of the Schröder-Bernstein Theorem because it uses Tarski's fixed point theorem, a powerful result. If one looks carefully at the proof, one realizes that there are two crucial ingredients:

1. The set C is closed under $g \circ f$, that is, $g \circ f(C) \subseteq C$.
2. $A - C \subseteq g(B)$.

Using these observations, it is possible to give a proof that circumvents the use of Tarski's theorem. Such a proof is given in Enderton [15], Chapter 6, and we give a sketch of this proof below.

Define a sequence of subsets, C_n, of A by recursion as follows:

$$\begin{aligned} C_0 &= A - g(B) \\ C_{n+1} &= (g \circ f)(C_n), \end{aligned}$$

and set

$$C = \bigcup_{n \geq 0} C_n.$$

Clearly, $A - C \subseteq g(B)$ and since direct images preserve unions, $(g \circ f)(C) \subseteq C$. The definition of h is similar to the one used in our proof:

$$h(a) = \begin{cases} f(a) & \text{if } a \in C \\ (g \restriction (A - C))^{-1}(a) & \text{if } a \notin C. \end{cases}$$

When $a \notin C$, i.e., $a \in A - C$, as $A - C \subseteq g(B)$ and g is injective, $g^{-1}(a)$ is indeed well-defined. As f and g are injective, so is g^{-1} on $A - C$. So, to check that h is injective, it is enough to prove that $f(a) = g^{-1}(b)$ with $a \in C$ and $b \notin C$ is impossible. However, if $f(a) = g^{-1}(b)$, then $(g \circ f)(a) = b$. Since $(g \circ f)(C) \subseteq C$ and $a \in C$, we get $b = (g \circ f)(a) \in C$, yet $b \notin C$, a contradiction. It is not hard to verify that h is surjective and therefore, h is a bijection between A and B. $\square$

The classical reference on lattices is Birkhoff [6]. We highly recommend this beautiful book (but it is not easy reading!).

We now turn to special properties of partial orders having to do with induction.

4.3 Well-Founded Orderings and Complete Induction

Have you ever wondered why induction on $\mathbb{N}$ actually "works"? The answer, of course, is that $\mathbb{N}$ was defined in such a way that, by Theorem 1.10.4, it is the "smallest" inductive set! But this is not a very illuminating answer. The key point is that every nonempty subset of $\mathbb{N}$ has a least element. This fact is intuitively clear since if we had some nonempty subset of $\mathbb{N}$ with no smallest element, then we could construct an infinite strictly decreasing sequence, $k_0 > k_1 > \cdots > k_n > \cdots$. But this is absurd, as such a sequence would eventually run into 0 and stop. It turns out that the deep reason why induction "works" on a poset is indeed that the poset ordering has a very special property and this leads us to the following definition:

Definition 4.3.1 Given a poset, $\langle X, \leq \rangle$, we say that $\leq$ is a *well-order (well ordering)* and that X is *well-ordered by* $\leq$ iff every nonempty subset of X has a least element.

When X is nonempty, if we pick any two-element subset, $\{a, b\}$, of X, since the subset $\{a, b\}$ must have a least element, we see that either $a \leq b$ or $b \leq a$, i.e., *every well-order is a total order*. First, let us confirm that $\mathbb{N}$ is indeed well-ordered.

Theorem 4.3.2 *(Well-Ordering of $\mathbb{N}$) The set of natural numbers, $\mathbb{N}$, is well-ordered.*

Proof. Not surprisingly we use induction, but we have to be a little shrewd. Let A be any nonempty subset of $\mathbb{N}$. We prove by contradiction that A has a least element. So, suppose A does not have a least element and let $P(m)$ be the predicate

$$P(m) \equiv (\forall k \in \mathbb{N})(k < m \Rightarrow k \notin A),$$

which says that no natural number strictly smaller than m is in A. We will prove by induction on m that $P(m)$ holds. But then, the fact that $P(m)$ holds for all m shows that $A = \emptyset$, a contradiction.

Let us now prove $P(m)$ by induction. The base case $P(0)$ holds trivially. Next, assume $P(m)$ holds; we want to prove that $P(m + 1)$ holds. Pick any $k < m + 1$. Then, either

(1) $k < m$, in which case, by the induction hypothesis, $k \notin A$; or

(2) $k = m$. By the induction hypothesis, $P(m)$ holds. Now, if m was in A, as $P(m)$ holds no $k < m$ would belong to A and m would be the least element of A, contradicting the assumption that A has no least element. Therefore, $m \notin A$.

Thus, in both cases, we proved that if $k < m + 1$, then $k \notin A$, establishing the induction hypothesis. This concludes the induction and the proof of Theorem 4.3.2. □

Theorem 4.3.2 yields another induction principle which is often more flexible that our original induction principle. This principle called *complete induction* (or sometimes *strong induction*) was already encountered in Section 2.3. It turns out that it is a special case of induction on a well-ordered set but it does not hurt to review it in the special case of the natural ordering on $\mathbb{N}$. Recall that $\mathbb{N}_+ = \mathbb{N} - \{0\}$.

Complete Induction Principle on $\mathbb{N}$.

In order to prove that a predicate, $P(n)$, holds for all $n \in \mathbb{N}$ it is enough to prove that

(1) $P(0)$ holds (the base case) and

(2) for every $m \in \mathbb{N}_+$, if $(\forall k \in \mathbb{N})(k < m \Rightarrow P(k))$ then $P(m)$.

As a formula, complete induction is stated as

$$P(0) \land (\forall m \in \mathbb{N}_+)[(\forall k \in \mathbb{N})(k < m \Rightarrow P(k)) \Rightarrow P(m)] \Rightarrow (\forall n \in \mathbb{N})P(n).$$

The difference between ordinary induction and complete induction is that in complete induction, the induction hypothesis, $(\forall k \in \mathbb{N})(k < m \Rightarrow P(k))$, assumes that $P(k)$ holds for all $k < m$ and not just for $m - 1$ (as in ordinary induction), in order to deduce $P(m)$. This gives us more proving power as we have more knowledge in order to prove $P(m)$.

We will have many occasions to use complete induction but let us first check that it is a valid principle. Even though we already sketched how the validity of complete induction is a consequence of the (ordinary) induction principle (Version 3) on $\mathbb{N}$ in Section 2.3 and we will soon give a more general proof of the validity of complete induction for a well-ordering, we feel that it is helpful to give the proof in the case of $\mathbb{N}$, as a warm-up.

Theorem 4.3.3 *The complete induction principle for $\mathbb{N}$ is valid.*

Proof. Let $P(n)$ be a predicate on $\mathbb{N}$ and assume that $P(n)$ satisfies conditions (1) and (2) of complete induction as stated above. We proceed by contradiction. So, assume that $P(n)$ fails for some $n \in \mathbb{N}$. If so, the set

$$F = \{n \in \mathbb{N} \mid P(n) = \textbf{false}\}$$

is nonempty. By Theorem 4.3.2, the set A has a least element, m, and thus,

$$P(m) = \textbf{false}.$$

Now, we can't have $m = 0$, as we assumed that $P(0)$ holds (by (1)) and since m is the least element for which $P(m) = \mathbf{false}$, we must have

$$P(k) = \mathbf{true} \quad \text{for all} \quad k < m.$$

But, this is exactly the premise in (2) and as we assumed that (2) holds, we deduce that

$$P(m) = \mathbf{true},$$

contradicting the fact that we already know that $P(m) = \mathbf{false}$. Therefore, $P(n)$ must hold for all $n \in \mathbb{N}$. $\square$

Remark: In our statement of the principle of complete induction, we singled out the base case, (1), and consequently, we stated the induction step (2) for every $m \in \mathbb{N}_+$, excluding the case $m = 0$, which is already covered by the base case. It is also possible to state the principle of complete induction in a more concise fashion as follows:

$$(\forall m \in \mathbb{N})[(\forall k \in \mathbb{N})(k < m \Rightarrow P(k)) \Rightarrow P(m)] \Rightarrow (\forall n \in \mathbb{N})P(n).$$

In the above formula, observe that when $m = 0$, which is now allowed, the premise $(\forall k \in \mathbb{N})(k < m \Rightarrow P(k))$ of the implication within the brackets is trivially true and so, $P(0)$ must still be established. In the end, exactly the same amount of work is required but some people prefer the second more concise version of the principle of complete induction. We feel that it would be easier for the reader to make the transition from ordinary induction to complete induction if we make explicit the fact that the base case must be established.

Let us illustrate the use of the complete induction principle by proving that every natural number factors as a product of primes. Recall that for any two natural numbers, $a, b \in \mathbb{N}$ with $b \neq 0$, we say that b *divides* a iff $a = bq$, for some $q \in \mathbb{N}$. In this case, we say that *a is divisible by b* and that *b is a factor of a*. Then, we say that a natural number, $p \in \mathbb{N}$, is a *prime number* (for short, a *prime*) if $p \geq 2$ and if p is only divisible by itself and by 1. Any prime number but 2 must be odd but the converse is false. For example, $2, 3, 5, 7, 11, 13, 17$ are prime numbers, but 9 is not. It can be shown that there are infinitely many prime numbers but to prove this, we need the following Theorem:

Theorem 4.3.4 *Every natural number, $n \geq 2$ can be factored as a product of primes, that is, n can be written as a product, $n = p_1^{m_1} \cdots p_k^{m_k}$, where the p_is are pairwise distinct prime numbers and $m_i \geq 1$ $(1 \leq i \leq k)$.*

Proof. We proceed by complete induction on $n \geq 2$. The base case, $n = 2$ is trivial, since 2 is prime.

Consider any $n > 2$ and assume that the induction hypothesis holds, that is, every m with $2 \leq m < n$ can be factored as a product of primes. There are two cases:

(a) The number n is prime. Then, we are done.

(b) The number n is not a prime. In this case, n factors as $n = n_1 n_2$, where $2 \leq n_1, n_2 < n$. By the induction hypothesis, n_1 has some prime factorization and so does n_2. If $\{p_1, \ldots, p_k\}$ is the union of all the primes occurring in these factorizations of n_1 and n_2, we can write

$$n_1 = p_1^{i_1} \cdots p_k^{i_k} \quad \text{and} \quad n_2 = p_1^{j_1} \cdots p_k^{j_k},$$

where $i_h, j_h \geq 0$ and, in fact, $i_h + j_h \geq 1$, for $1 \leq h \leq k$. Consequently, n factors as the product of primes,

$$n = p_1^{i_1+j_1} \cdots p_k^{i_k+j_k},$$

with $i_h + j_h \geq 1$, establishing the induction hypothesis. □

Remark: It can be shown that the prime factorization of a natural number is unique up to permutation of the primes $p_1, \ldots, p_k$ but this requires the Euclidean Division Lemma. However, we can prove right away that there are infinitely primes.

Theorem 4.3.5 *Given any natural number, $n \geq 1$, there is a prime number, p, such that $p > n$. Consequently, there are infinitely many primes.*

Proof. Let $m = n! + 1$. If m is prime, we are done. Otherwise, by Theorem 4.3.4, the number m has a prime decomposition. We claim that $p > n$ for every prime in this decomposition. If not, $2 \leq p \leq n$ and then p would divide both $n! + 1$ and $n!$, so p would divide 1, a contradiction. □

As an application of Theorem 4.3.2, we prove the "Euclidean Division Lemma" for the integers.

Theorem 4.3.6 *(Euclidean Division Lemma for $\mathbb{Z}$) Given any two integers, $a, b \in \mathbb{Z}$, with $b \neq 0$, there is some unique integer, $q \in \mathbb{Z}$ (the quotient), and some unique natural number, $r \in \mathbb{N}$ (the remainder or residue), so that*

$$a = bq + r \quad \textit{with} \quad 0 \leq r < |b|.$$

Proof. First, let us prove the existence of q and r with the required condition on r. We claim that if we show existence in the special case where $a, b \in \mathbb{N}$ (with $b \neq 0$), then we can prove existence in the general case. There are four cases:

1. If $a, b \in \mathbb{N}$, with $b \neq 0$, then we are done.

2. If $a \geq 0$ and $b < 0$, then $-b > 0$, so we know that there exist q, r with

$$a = (-b)q + r \quad \text{with} \quad 0 \leq r \leq -b - 1.$$

Then,

$$a = b(-q) + r \quad \text{with} \quad 0 \leq r \leq |b| - 1.$$

3. If $a < 0$ and $b > 0$, then $-a > 0$, so we know that there exist q, r with
$$-a = bq + r \quad \text{with} \quad 0 \leq r \leq b - 1.$$
Then,
$$a = b(-q) - r \quad \text{with} \quad 0 \leq r \leq b - 1.$$
If $r = 0$, we are done. Otherwise, $1 \leq r \leq b - 1$, which implies $1 \leq b - r \leq b - 1$, so we get
$$a = b(-q) - b + b - r = b(-(q + 1)) + b - r \quad \text{with} \quad 0 \leq b - r \leq b - 1.$$

4. If $a < 0$ and $b < 0$, then $-a > 0$ and $-b > 0$, so we know that there exist q, r with
$$-a = (-b)q + r \quad \text{with} \quad 0 \leq r \leq -b - 1.$$
Then,
$$a = bq - r \quad \text{with} \quad 0 \leq r \leq -b - 1.$$
If $r = 0$, we are done. Otherwise, $1 \leq r \leq -b - 1$, which implies $1 \leq -b - r \leq -b - 1$, so we get
$$a = bq + b - b - r = b(q + 1) + (-b - r) \quad \text{with} \quad 0 \leq -b - r \leq |b| - 1.$$

We are now reduced to proving the existence of q and r when $a, b \in \mathbb{N}$ with $b \neq 0$. Consider the set
$$R = \{a - bq \in \mathbb{N} \mid q \in \mathbb{N}\}.$$
Note that $a \in R$, by setting $q = 0$, since $a \in \mathbb{N}$. Therefore, R is nonempty. By Theorem 4.3.2, the nonempty set, R, has a least element, r. We claim that $r \leq b - 1$ (of course, $r \geq 0$ as $R \subseteq \mathbb{N}$). If not, then $r \geq b$, and so $r - b \geq 0$. As $r \in R$, there is some $q \in \mathbb{N}$ with $r = a - bq$. But now, we have
$$r - b = a - bq - b = a - b(q + 1)$$
and as $r - b \geq 0$, we see that $r - b \in R$ with $r - b < r$ (since $b \neq 0$), contradicting the minimality of r. Therefore, $0 \leq r \leq b - 1$, proving the existence of q and r with the required condition on r.

We now go back to the general case where $a, b \in \mathbb{Z}$ with $b \neq 0$ and we prove uniqueness of q and r (with the required condition on r). So, assume that
$$a = bq_1 + r_1 = bq_2 + r_2 \quad \text{with} \quad 0 \leq r_1 \leq |b| - 1 \quad \text{and} \quad 0 \leq r_2 \leq |b| - 1.$$
Now, as $0 \leq r_1 \leq |b| - 1$ and $0 \leq r_2 \leq |b| - 1$, we have $|r_1 - r_2| < |b|$, and from $bq_1 + r_1 = bq_2 + r_2$, we get
$$b(q_2 - q_1) = r_1 - r_2,$$

which yields

$$|b||q_2 - q_1| = |r_1 - r_2|.$$

Since $|r_1 - r_2| < |b|$, we must have $r_1 = r_2$. Then, from $b(q_2 - q_1) = r_1 - r_2 = 0$, as $b \neq 0$, we get $q_1 = q_2$, which concludes the proof. □

We will now show that complete induction holds for a very broad class of partial orders called *well-founded orderings* that subsume well-orderings.

Definition 4.3.7 Given a poset, $\langle X, \leq \rangle$, we say that $\leq$ is a *well-founded ordering (order)* and that X is *well-founded* iff X has **no** infinite strictly decreasing sequence
$x_0 > x_1 > x_2 > \cdots > x_n > x_{n+1} > \cdots$.

The following property of well-founded sets is fundamental:

Proposition 4.3.8 *A poset, $\langle X, \leq \rangle$, is well-founded iff every nonempty subset of X has a minimal element.*

Proof. First, assume that every nonempty subset of X has a minimal element. If we had an infinite strictly decreasing sequence, $x_0 > x_1 > x_2 > \cdots > x_n > \cdots$, then the set $A = \{x_n\}$ would have no minimal element, a contradiction. Therefore, X is well-founded.

Now, assume that X is well-founded. We prove that A has a least element by contradiction. So, let A be some nonempty subset of X and suppose A has no least element. This means that for every $a \in A$, there is some $b \in A$ with $a > b$. Using the Axiom of Choice (Graph Version), there is some function, $g\colon A \to A$, with the property that

$$a > g(a), \quad \text{for all} \quad a \in A.$$

Now, since A is nonempty, we can pick some element, say $a \in A$. By the recursion Theorem (Theorem 2.5.1), there is a unique function, $f\colon \mathbb{N} \to A$, so that

$$\begin{aligned} f(0) &= a, \\ f(n+1) &= g(f(n)) \qquad \text{for all} \quad n \in \mathbb{N}. \end{aligned}$$

But then, f defines an infinite sequence, $\{x_n\}$, with $x_n = f(n)$, so that $x_n > x_{n+1}$ for all $n \in \mathbb{N}$, contradicting the fact that X is well-founded. □

So, the seemingly weaker condition that there is **no** infinite strictly decreasing sequence in X is equivalent to the fact that every nonempty subset of X has a minimal element. If X is a total order, any minimal element is actually a least element and so, we get

Corollary 4.3.9 *A poset, $\langle X, \leq \rangle$, is well-ordered iff $\leq$ is total and X is well-founded.*

Note that the notion of a well-founded set is more general than that of a well-ordered set, since a well-founded set is not necessarily totally ordered.

Remark: Suppose we can prove some property, P, by (ordinary) induction on $\mathbb{N}$. Then, I claim that P can also be proved by complete induction on $\mathbb{N}$. To see this, observe first that the base step is identical. Also, for all $m \in \mathbb{N}_+$, the implication

$$(\forall k \in \mathbb{N})(k < m \Rightarrow P(k)) \Rightarrow P(m-1)$$

holds and since the induction step (in ordinary induction) consists in proving for all $m \in \mathbb{N}_+$ that

$$P(m-1) \Rightarrow P(m)$$

holds, from this implication and the previous implication we deduce that for all $m \in \mathbb{N}_+$, the implication

$$(\forall k \in \mathbb{N})(k < m \Rightarrow P(k)) \Rightarrow P(m)$$

holds, which is exactly the induction step of the complete induction method. So, we see that complete induction on $\mathbb{N}$ implies ordinary induction on $\mathbb{N}$. The converse is also true but we leave it as a fun exercise. But now, by Theorem 4.3.2, (ordinary) induction on $\mathbb{N}$ implies that $\mathbb{N}$ is well-ordered and by Theorem 4.3.3, the fact that $\mathbb{N}$ is well-ordered implies complete induction on $\mathbb{N}$. Since we just showed that complete induction on $\mathbb{N}$ implies (ordinary) induction on $\mathbb{N}$, we conclude that all three are equivalent, that is

(ordinary) induction on $\mathbb{N}$ is valid
iff
complete induction on $\mathbb{N}$ is valid
iff
$\mathbb{N}$ is well-ordered.

These equivalences justify our earlier claim that the ability to do induction hinges on some key property of the ordering, in this case, that it is a well-ordering.

We finally come to the principle of *complete induction* (also called *transfinite induction* or *structural induction)*, which, as we shall prove, is valid for all well-founded sets. Since every well-ordered set is also well-founded, complete induction is a very general induction method.

Let $(X, \leq)$ be a well-founded poset and let P be a predicate on X (i.e., a function $P\colon X \to \{\mathbf{true}, \mathbf{false}\}$).

Principle of Complete Induction on a Well-Founded Set.

To prove that a property P holds for all $z \in X$, it suffices to show that, for every $x \in X$,

$(*)$ if x is minimal or $P(y)$ holds for all $y < x$,

$(**)$ then $P(x)$ holds.

The statement $(*)$ is called the *induction hypothesis*, and the implication

for all x, $(*)$ implies $(**)$ is called the *induction step*. Formally, the induction principle can be stated as:

$$(\forall x \in X)[(\forall y \in X)(y < x \Rightarrow P(y)) \Rightarrow P(x)] \Rightarrow (\forall z \in X)P(z) \tag{CI}$$

Note that if x is minimal, then there is no $y \in X$ such that $y < x$, and $(\forall y \in X)(y < x \Rightarrow P(y))$ is true. Hence, we must show that $P(x)$ holds for every minimal element, x. These cases are called the *base cases*.

Complete induction is not valid for arbitrary posets (see the problems) but holds for well-founded sets as shown in the following theorem.

Theorem 4.3.10 *The principle of complete induction holds for every well-founded set.*

Proof. We proceed by contradiction. Assume that (CI) is false. Then,

$$(\forall x \in X)[(\forall y \in X)(y < x \Rightarrow P(y)) \Rightarrow P(x)] \tag{1}$$

holds and

$$(\forall z \in X)P(z) \tag{2}$$

is false, that is, there is some $z \in X$ so that

$$P(z) = \mathbf{false}.$$

Hence, the subset F of X defined by

$$F = \{x \in X \mid P(x) = \mathbf{false}\}$$

is nonempty. Since X is well founded, by Proposition 4.3.8, F has some minimal element, b. Since (1) holds for all $x \in X$, letting $x = b$, we see that

$$[(\forall y \in X)(y < b \Rightarrow P(y)) \Rightarrow P(b)] \tag{3}$$

holds. If b is also minimal in X, then there is no $y \in X$ such that $y < b$ and so,

$$(\forall y \in X)(y < b \Rightarrow P(y))$$

holds trivially and (3) implies that $P(b) = \mathbf{true}$, which contradicts the fact that $b \in F$. Otherwise, for every $y \in X$ such that $y < b$, $P(y) = \mathbf{true}$, since otherwise y would belong to F and b would not be minimal. But then,

$$(\forall y \in X)(y < b \Rightarrow P(y))$$

also holds and (3) implies that $P(b) = \mathbf{true}$, contradicting the fact that $b \in F$. Hence, complete induction is valid for well-founded sets. □

As an illustration of well-founded sets, we define the *lexicographic ordering* on pairs. Given a partially ordered set $\langle X, \leq \rangle$, the *lexicographic ordering*, $<<$, on $X \times X$ induced by $\leq$ is defined a follows: For all $x, y, x', y' \in X$,

$$(x, y) << (x', y') \quad \text{iff either}$$

$$\begin{aligned}
&x = x' \quad \text{and} \quad y = y' \quad \text{or}\\
&x < x' \quad \text{or}\\
&x = x' \quad \text{and} \quad y < y'.
\end{aligned}$$

We leave it as an exercise to check that $<<$ is indeed a partial order on $X \times X$. The following proposition will be useful.

Proposition 4.3.11 *If $\langle X, \leq \rangle$ is a well-founded set, then the lexicographic ordering $<<$ on $X \times X$ is also well founded.*

Proof. We proceed by contradiction. Assume that there is an infinite decreasing sequence $(\langle x_i, y_i \rangle)_i$ in $X \times X$. Then, either,

(1) There is an infinite number of distinct x_i, or

(2) There is only a finite number of distinct x_i.

In case (1), the subsequence consisting of these distinct elements forms a decreasing sequence in X, contradicting the fact that $\leq$ is well founded. In case (2), there is some k such that $x_i = x_{i+1}$, for all $i \geq k$. By definition of $<<$, the sequence $(y_i)_{i \geq k}$ is a decreasing sequence in X, contradicting the fact that $\leq$ is well founded. Hence, $<<$ is well founded on $X \times X$. □

As an illustration of the principle of complete induction, consider the following example in which it is shown that a function defined recursively is a total function.

Example (Ackermann's function) The following function, $A\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, known as *Ackermann's function* is well known in recursive function theory for its extraordinary rate of growth. It is defined recursively as follows:

$$\begin{aligned}
A(x, y) = \;& \textbf{if } x = 0 \textbf{ then } y + 1\\
& \textbf{else if } y = 0 \textbf{ then } A(x - 1, 1)\\
& \textbf{else } A(x - 1, A(x, y - 1)).
\end{aligned}$$

We wish to prove that A is a total function. We proceed by complete induction over the lexicographic ordering on $\mathbb{N} \times \mathbb{N}$.

1. The base case is $x = 0$, $y = 0$. In this case, since $A(0, y) = y + 1$, $A(0, 0)$ is defined and equal to 1.

2. The induction hypothesis is that for any (m, n), $A(m', n')$ is defined for all $(m', n') << (m, n)$, with $(m, n) \neq (m', n')$.

3. For the induction step, we have three cases:

   (a) If $m = 0$, since $A(0, y) = y + 1$, $A(0, n)$ is defined and equal to $n + 1$.

   (b) If $m \neq 0$ and $n = 0$, since $(m - 1, 1) << (m, 0)$ and $(m - 1, 1) \neq (m, 0)$, by the induction hypothesis, $A(m - 1, 1)$ is defined, and so $A(m, 0)$ is defined since it is equal to $A(m - 1, 1)$.

   (c) If $m \neq 0$ and $n \neq 0$, since $(m, n - 1) << (m, n)$ and $(m, n - 1) \neq (m, n)$, by the induction hypothesis, $A(m, n - 1)$ is defined. Since $(m - 1, y) << (m, z)$ and $(m - 1, y) \neq (m, z)$ no matter what y and z are, $(m - 1, A(m, n - 1)) << (m, n)$ and $(m - 1, A(m, n - 1)) \neq (m, n)$, and by the induction hypothesis, $A(m - 1, A(m, n - 1))$ is defined. But this is precisely $A(m, n)$, and so $A(m, n)$ is defined. This concludes the induction step.

Hence, $A(x, y)$ is defined for all $x, y \geq 0$. □

4.4 Unique Prime Factorization in $\mathbb{Z}$ and GCD's

In the previous section, we proved that every natural number, $n \geq 2$, can be factored as a product of primes numbers. In this section, we use the Euclidean Division Lemma to prove that such a factorization is unique. For this, we need to introduce greatest common divisors (gcd's) and prove some of their properties.

In this section, it will be convenient to allow 0 to be a divisor. So, given any two integers, $a, b \in \mathbb{Z}$, we will say that b *divides* a *and that* a *is a multiple of* b iff $a = bq$, for some $q \in \mathbb{Z}$. Contrary to our previous definition, $b = 0$ is allowed as a divisor. However, this changes very little because if 0 divides a, then $a = 0q = 0$, that is, *the only integer divisible by* 0 *is* 0.

We begin by introducing a very important notion in algebra, that of an ideal, and prove a fundamental property of the ideals of $\mathbb{Z}$.

Definition 4.4.1 An *ideal of* $\mathbb{Z}$ is any nonempty subset, $\mathfrak{I}$, of $\mathbb{Z}$ satisfying the following two properties:

(ID1) If $a, b \in \mathfrak{I}$, then $b - a \in \mathfrak{I}$.

(ID2) If $a \in \mathfrak{I}$, then $ak \in \mathfrak{I}$ for every $k \in \mathbb{Z}$.

An ideal, $\mathfrak{I}$, is a *principal ideal* if there is some $a \in \mathfrak{I}$, *called a generator*, such that $\mathfrak{I} = \{ak \mid k \in \mathbb{Z}\}$. The equality $\mathfrak{I} = \{ak \mid k \in \mathbb{Z}\}$ is also written as $\mathfrak{I} = a\mathbb{Z}$ or as $\mathfrak{I} = (a)$. The ideal $\mathfrak{I} = (0) = \{0\}$ is called the *null ideal*.

Note that if $\mathfrak{I}$ is an ideal, then $\mathfrak{I} = \mathbb{Z}$ iff $1 \in \mathfrak{I}$. Since by definition, an ideal $\mathfrak{I}$ is nonempty, there is some $a \in \mathfrak{I}$, and by (ID1) we get $0 = a - a \in \mathfrak{I}$. Then, for every $a \in \mathfrak{I}$, since $0 \in \mathfrak{I}$, by (ID1) we get $-a \in \mathfrak{I}$.

Theorem 4.4.2 *Every ideal, $\mathfrak{I}$, of $\mathbb{Z}$, is a principal ideal, i.e., $\mathfrak{I} = m\mathbb{Z}$ for some unique $m \in \mathbb{N}$, with $m > 0$ iff $\mathfrak{I} \neq (0)$.*

Proof. Note that $\mathfrak{I} = (0)$ iff $\mathfrak{I} = 0\mathbb{Z}$ and the theorem holds in this case. So, assume that $\mathfrak{I} \neq (0)$. Then, our previous observation that $-a \in \mathfrak{I}$ for every $a \in \mathfrak{I}$ implies that some positive integer belongs to $\mathfrak{I}$ and so, the set $\mathfrak{I} \cap \mathbb{N}_+$ is nonempty. As $\mathbb{N}$ is well-ordered, this set has a smallest element, say $m > 0$. We claim that $\mathfrak{I} = m\mathbb{Z}$.

As $m \in \mathfrak{I}$, by (ID2), $m\mathbb{Z} \subseteq \mathfrak{I}$. Conversely, pick any $n \in \mathfrak{I}$. By the Euclidean division Theorem, there are unique $q \in \mathbb{Z}$ and $r \in \mathbb{N}$ so that $n = mq + r$, with $0 \leq r < m$. If $r > 0$, since $m \in \mathfrak{I}$, by (ID2), $mq \in \mathfrak{I}$ and by (ID1), we get $r = n - mq \in \mathfrak{I}$. Yet $r < m$, contradicting the minimality of m. Therefore, $r = 0$, so $n = mq \in m\mathbb{Z}$, establishing that $\mathfrak{I} \subseteq m\mathbb{Z}$ and thus, $\mathfrak{I} = m\mathbb{Z}$, as claimed. As to uniqueness, clearly $(0) \neq m\mathbb{Z}$ if $m \neq 0$, so assume $m\mathbb{Z} = m'\mathbb{Z}$, with $m > 0$ and $m' > 0$. Then, m divides m' and m' divides m, but we already proved earlier that this implies $m = m'$. $\square$

Theorem 4.4.2 is often phrased: $\mathbb{Z}$ is a *principal ideal domain*, for short, a *PID*. Note that the natural number m such that $\mathfrak{I} = m\mathbb{Z}$ is a divisor of every element in $\mathfrak{I}$.

Corollary 4.4.3 *For any two integers, $a, b \in \mathbb{Z}$, there is a unique natural number, $d \in \mathbb{N}$, and some integers, $u, v \in \mathbb{Z}$, so that d divides both a and b and*

$$ua + vb = d.$$

(The above is called the Bezout identity.) Furthermore, $d = 0$ iff $a = 0$ and $b = 0$.

Proof. It is immediately verified that

$$\mathfrak{I} = \{ha + kb \mid h, k \in \mathbb{Z}\}$$

is an ideal of $\mathbb{Z}$ with $a, b \in \mathfrak{I}$. Therefore, by Theorem 4.4.2, there is a unique, $d \in \mathbb{N}$, so that $\mathfrak{I} = d\mathbb{Z}$. We already observed that d divides every number in $\mathfrak{I}$ so, as $a, b \in \mathfrak{I}$, we see that d divides a and b. If $d = 0$, as d divides a and b, we must have $a = b = 0$. Conversely, if $a = b = 0$, then $d = ua + bv = 0$. $\square$

The natural number d of corollary 4.4.3 divides both a and b. Moreover, every divisor of a and b divides $d = ua + vb$. This motivates the definition:

Definition 4.4.4 Given any two integers, $a, b \in \mathbb{Z}$, an integer, $d \in \mathbb{Z}$, is a *greatest common divisor of a and b* (for short, a *gcd of a and b*) if d divides a and b and, for any integer, $h \in \mathbb{Z}$, if h divides a and b, then h divides d. We say that a and b are *relatively prime* if 1 is a gcd of a and b.

Remarks:

1. Assume $a = b = 0$. Then, any integer, $d \in \mathbb{Z}$, is a divisor of 0. In particular, 0 divides 0. According to Definition 4.4.4, this implies $\gcd(0, 0) = 0$. If $(a, b) \neq (0, 0)$, then 1 divides a and b, so $\gcd(a, b) = d > 0$. In this case, if d' is any other gcd of a and b, then $d = qd'$ and $d' = q'd$ for some $q, q' \in \mathbb{Z}$. So, $d = qq'd$ which implies $qq' = 1$ (since $d \neq 0$) and thus, $d' = \pm d$. So, according to the above definition, when $(a, b) \neq (0, 0)$, gcd's are not unique. However, exactly one of d or $-d$ is positive, so we will refer to this positive gcd as "the" gcd of a and b and write $d = \gcd(a, b)$.

2. Observe that $d = \gcd(a, b)$ is indeed the largest positive common divisor of a and b since every divisor of a and b must divide d. However, we did not use this property as one of the conditions for being a gcd because such a condition does not generalize to other rings where a total order is not available. Another minor reason is that if we had used in the definition of a gcd the condition that $\gcd(a, b)$ should be the largest common divisor of a and b, as every integer divides 0, $\gcd(0, 0)$ would be undefined!

3. Our definition of the gcd makes sense even if we allow $a = b = 0$. In this case, $\gcd(0, 0) = 0$. If we did not allows 0 to be a divisor, the situation would be different. Indeed, if we had $\gcd(0, 0) = d$ for some $d > 0$, as every other positive integer, d', divides 0, every integer d' would have to divide d, which is absurd. This is why we relaxed our definition to allow 0 to be a divisor. Nevertheless, the cases where $a = 0$ or $b = 0$ are somewhat degenerate cases so we prefer to stick to the simpler situation where we only consider gcd's for two nonzero integers.

Let $p \in \mathbb{N}$ be a prime number. Then, note that for any other integer, n, if p does not divide n, then $\gcd(p, n) = 1$, as the only divisors of p are 1 and p.

Proposition 4.4.5 *Given any two integers, $a, b \in \mathbb{Z}$, a natural number, $d \in \mathbb{N}$, is the greatest common divisor of a and b iff d divides a and b and if there are some integers, $u, v \in \mathbb{Z}$, so that*

$$ua + vb = d. \qquad \text{(Bezout Identity)}$$

In particular, a and b are relatively prime iff there are some integers, $u, v \in \mathbb{Z}$, so that

$$ua + vb = 1. \qquad \text{(Bezout Identity)}$$

Proof. We already observed that half of Proposition 4.4.5 holds, namely if $d \in \mathbb{N}$ divides a and b and if there are some integers, $u, v \in \mathbb{Z}$, so that $ua + vb = d$, then, d is the gcd of a and b. Conversely, assume that $d = \gcd(a, b)$. If $d = 0$, then $a = b = 0$ and the proposition holds trivially. So, assume $d > 0$, in which case $(a, b) \neq (0, 0)$. By Corollary 4.4.3, there is a unique $m \in \mathbb{N}$ with $m > 0$ that divides a and b and there are some integers, $u, v \in \mathbb{Z}$, so that

$$ua + vb = m.$$

But now, m is also the (positive) gcd of a and b, so $d = m$ and our Proposition holds. Now, a and b are relatively prime iff $\gcd(a, b) = 1$ in which case the condition that $d = 1$ divides a and b is trivial. □

Remark: The gcd of two nonzero integers can be found using a method involving Euclidean division and so can the numbers u and v.

Proposition 4.4.5 implies a very crucial property of divisibility in any PID.

Proposition 4.4.6 *(Euclid's proposition) Let $a, b, c \in \mathbb{Z}$ be any integers. If a divides bc and a is relatively prime to b, then a divides c.*

Proof. From Proposition 4.4.5, a and b are relatively prime iff there exist some integers, $u, v \in \mathbb{Z}$ such that

$$ua + vb = 1.$$

Then, we have

$$uac + vbc = c,$$

and since a divides bc, it divides both uac and vbc and so, a divides c. □

In particular, if p is a prime number and if p divides ab, where $a, b \in \mathbb{Z}$ are nonzero, then either p divides a or p divides b since if p does not divide a, by a previous remark, then p and a are relatively prime, so Proposition 4.4.6 implies that p divides c.

Proposition 4.4.7 *Let $a, b_1, \ldots, b_m \in \mathbb{Z}$ be any integers. If a and b_i are relatively prime for all i, with $1 \leq i \leq m$, then a and $b_1 \cdots b_m$ are relatively prime.*

Proof. We proceed by induction on m. The case $m = 1$ is trivial. Let $c = b_2 \cdots b_m$. By the induction hypothesis, a and c are relatively prime. Let d the gcd of a and b_1c. We claim that d is relatively prime to b_1. Otherwise, d and b_1 would have some gcd $d_1 \neq 1$ which would divide both a and b_1, contradicting the fact that a and b_1 are relatively prime. Now, by Proposition 4.4.6, since d divides b_1c and d and b_1 are relatively prime, d divides $c = b_2 \cdots b_m$. But then, d is a divisor of a and c, and since a and c are relatively prime, $d = 1$, which means that a and $b_1 \cdots b_m$ are relatively prime. □

We can now prove the uniqueness of prime factorizations in $\mathbb{N}$.

Theorem 4.4.8 *(Unique Prime Factorization in $\mathbb{N}$) For every nonzero natural number, $a \geq 2$, there exists a unique set, $\{\langle p_1, k_1\rangle, \ldots, \langle p_m, k_m\rangle\}$, where the p_i's are distinct prime numbers and the k_i's are (not necessarily distinct) integers, with $m \geq 1$, $k_i \geq 1$, and*

$$a = p_1^{k_1} \cdots p_m^{k_m}.$$

Proof. The existence of such a factorization has already been proved in Theorem 4.3.4.

Let us now prove uniqueness. Assume that

$$a = p_1^{k_1} \cdots p_m^{k_m} \quad \text{and} \quad a = q_1^{h_1} \cdots q_n^{h_n}.$$

Thus, we have

$$p_1^{k_1} \cdots p_m^{k_m} = q_1^{h_1} \cdots q_n^{h_n}.$$

We prove that $m = n$, $p_i = q_i$ and $h_i = k_i$, for all i, with $1 \leq i \leq n$. The proof proceeds by induction on $h_1 + \cdots + h_n$.

If $h_1 + \cdots + h_n = 1$, then $n = 1$ and $h_1 = 1$. Then,

$$p_1^{k_1} \cdots p_m^{k_m} = q_1,$$

and since q_1 and the p_i are prime numbers, we must have $m = 1$ and $p_1 = q_1$ (a prime is only divisible by 1 or itself).

If $h_1 + \cdots + h_n \geq 2$, since $h_1 \geq 1$, we have

$$p_1^{k_1} \cdots p_m^{k_m} = q_1 q,$$

with

$$q = q_1^{h_1 - 1} \cdots q_n^{h_n},$$

where $(h_1 - 1) + \cdots + h_n \geq 1$ (and $q_1^{h_1 - 1} = 1$ if $h_1 = 1$). Now, if q_1 is not equal to any of the p_i, by a previous remark, q_1 and p_i are relatively prime, and by Proposition 4.4.7, q_1 and $p_1^{k_1} \cdots p_m^{k_m}$ are relatively prime. But this contradicts the fact that q_1 divides $p_1^{k_1} \cdots p_m^{k_m}$. Thus, q_1 is equal to one of the p_i. Without loss of generality, we can assume that $q_1 = p_1$. Then, as $q_1 \neq 0$, we get

$$p_1^{k_1 - 1} \cdots p_m^{k_m} = q_1^{h_1 - 1} \cdots q_n^{h_n},$$

where $p_1^{k_1 - 1} = 1$ if $k_1 = 1$, and $q_1^{h_1 - 1} = 1$ if $h_1 = 1$. Now, $(h_1 - 1) + \cdots + h_n < h_1 + \cdots + h_n$, and we can apply the induction hypothesis to conclude that $m = n$, $p_i = q_i$ and $h_i = k_i$, with $1 \leq i \leq n$. $\square$

Theorem 4.4.8 is a basic but very important result of number theory and it has many applications. It also reveals the importance of the primes as the building blocks of all numbers.

Remark: Theorem 4.4.8 also applies to any nonzero integer $a \in \mathbb{Z} - \{-1, +1\}$, by adding a suitable sign in front of the prime factorization. That is, we have a unique prime factorization of the form

$$a = \pm p_1^{k_1} \cdots p_m^{k_m}.$$

Theorem 4.4.8 shows that $\mathbb{Z}$ is a *unique factorization domain*, for short, a *UFD*. Such rings play an important role because every nonzero element which is not a unit (i.e., which is not invertible) has a unique factorization (up to some unit factor) into so-called *irreducible elements* which generalize the primes.

We now take a well-deserved break from partial orders and induction and study equivalence relations, an equally important class of relations.

4.5 Equivalence Relations and Partitions

Equivalence relations basically generalize the identity relation. Technically, the definition of an equivalence relation is obtained from the definition of a partial order (Definition 4.1.1) by changing the third condition, antisymmetry, to *symmetry*.

Definition 4.5.1 A binary relation, R, on a set, X, is an *equivalence relation* iff it is *reflexive*, *transitive* and *symmetric*, that is:

(1) (*Reflexivity*): aRa, for all $a \in X$;

(2) (*Transitivity*): If aRb and bRc, then aRc, for all $a, b, c \in X$.

(3) (*symmetry*): If aRb, then bRa, for all $a, b \in X$.

Here are some examples of equivalence relations.

1. The identity relation, id_X, on a set X is an equivalence relation.

2. The relation $X \times X$ is an equivalence relation.

3. Let S be the set of students in CIS260. Define two students to be equivalent iff they were born the same year. It is trivial to check that this relation is indeed an equivalence relation.

4. Given any natural number, $p \geq 1$, define a relation on $\mathbb{Z}$ as follows:

$$m \equiv n \pmod{p}$$

iff $p \mid m - n$, i.e., p divides $m - n$. It is an easy exercise to check that this is indeed an equivalence relation called *congruence modulo p*.

5. Equivalence of propositions is the relation defined so that $P \equiv Q$ iff $P \Rightarrow Q$ and $Q \Rightarrow P$ are both provable (say, classically). It is easy to check that logical equivalence is an equivalence relation.

6. Suppose $f\colon X \to Y$ is a function. Then, we define the relation $\equiv_f$ on X by
$$x \equiv_f y \quad \text{iff} \quad f(x) = f(y).$$
It is immediately verified that $\equiv_f$ is an equivalence relation. Actually, we are going to show that every equivalence relation arises in this way, in terms of (surjective) functions.

The crucial property of equivalence relations is that they *partition* their domain, X, into pairwise disjoint nonempty blocks. Intuitively, they carve out X into a bunch of puzzle pieces.

Definition 4.5.2 Given an equivalence relation, R, on a set, X, for any $x \in X$, the set
$$[x]_R = \{y \in X \mid xRy\}$$
is the *equivalence class of* x. Each equivalence class, $[x]_R$, is also denoted $\overline{x}_R$ and the subscript R is often omitted when no confusion arises. The set of equivalence classes of R is denoted by X/R. The set X/R is called the *quotient of* X *by* R or *quotient of* X *modulo* R. The function, $\pi\colon X \to X/R$, given by
$$\pi(x) = [x]_R, \quad x \in X,$$
is called the *canonical projection* (or *projection*) of X onto X/R.

Since every equivalence relation is reflexive, i.e., xRx for every $x \in X$, observe that $x \in [x]_R$ for any $x \in R$, that is, every equivalence class is *nonempty*. It is also clear that the projection, $\pi\colon X \to X/R$, is surjective. The main properties of equivalence classes are given by

Proposition 4.5.3 *Let R be an equivalence relation on a set, X. For any two elements $x, y \in X$, we have*
$$xRy \quad \textit{iff} \quad [x] = [y].$$
Moreover, the equivalences classes of R satisfy the following properties:

(1) $[x] \neq \emptyset$, for all $x \in X$;

(2) If $[x] \neq [y]$ then $[x] \cap [y] = \emptyset$;

(3) $X = \bigcup_{x \in X} [x]$.

Proof. First, assume that $[x] = [y]$. We observed that by reflexivity, $y \in [y]$. As $[x] = [y]$, we get $y \in [x]$ and by definition of $[x]$, this means that xRy.

Next, assume that xRy. Let us prove that $[y] \subseteq [x]$. Pick any $z \in [y]$; this means that yRz. By transitivity, we get xRz, ie., $z \in [x]$, proving that $[y] \subseteq [x]$. Now, as R is symmetric, xRy implies that yRx and the previous argument yields $[x] \subseteq [y]$. Therefore, $[x] = [y]$, as needed.

Property (1) follows from the fact that $x \in [x]$ (by reflexivity).

Let us prove the contrapositive of (2). So, assume $[x] \cap [y] \neq \emptyset$. Thus, there is some z so that $z \in [x]$ and $z \in [y]$, i.e.,

$$xRz \quad \text{and} \quad yRz.$$

By symmetry, we get zRy and by transitivity, xRy. But then, by the first part of the proposition, we deduce $[x] = [y]$, as claimed.

The third property follows again from the fact that $x \in [x]$. $\square$

A useful way of interpreting Proposition 4.5.3 is to say that the equivalence classes of an equivalence relation form a partition, as defined next.

Definition 4.5.4 Given a set, X, a *partition of* X is any family, $\Pi = \{X_i\}_{i \in I}$, of subsets of X such that

(1) $X_i \neq \emptyset$, for all $i \in I$ (each X_i is nonempty);

(2) If $i \neq j$ then $X_i \cap X_j = \emptyset$ (the X_i are pairwise disjoint);

(3) $X = \bigcup_{i \in I} X_i$ (the family is exhaustive).

Each set X_i is called a *block* of the partition.

In the example where equivalence is determined by the same year of birth, each equivalence class consists of those students having the same year of birth. Let us now go back to the example of congruence modulo p (with $p > 0$) and figure out what are the blocks of the corresponding partition. Recall that

$$m \equiv n \pmod{p}$$

iff $m - n = pk$ for some $k \in \mathbb{Z}$. By the division Theorem (Theorem 4.3.6), we know that there exist some unique q, r, with $m = pq + r$ and $0 \leq r \leq p - 1$. Therefore, for every $m \in \mathbb{Z}$,

$$m \equiv r \pmod{p} \quad \text{with} \quad 0 \leq r \leq p - 1,$$

which shows that there are p equivalence classes, $[0], [1], \ldots, [p-1]$, where the equivalence class, $[r]$ (with $0 \leq r \leq p - 1$), consists of all integers of the form $pq + r$, where $q \in \mathbb{Z}$, i.e., those integers whose residue modulo p is r.

Proposition 4.5.3 defines a map from the set of equivalence relations on X to the set of partitions on X. Given any set, X, let $\mathrm{Equiv}(X)$ denote the set of equivalence relations on X and let $\mathrm{Part}(X)$ denote the set of partitions on X. Then, Proposition 4.5.3 defines the function, $\Pi\colon \mathrm{Equiv}(X) \to \mathrm{Part}(X)$, given by,

$$\Pi(R) = X/R = \{[x]_R \mid x \in X\},$$

where R is any equivalence relation on X. We also write Π_R instead of $\Pi(R)$.

There is also a function, $\mathcal{R}\colon \mathrm{Part}(X) \to \mathrm{Equiv}(X)$, that assigns an equivalence relation to a partition a shown by the next proposition.

Proposition 4.5.5 *For any partition, $\Pi = \{X_i\}_{i\in I}$, on a set, X, the relation, $\mathcal{R}(\Pi)$, defined by*

$$x\mathcal{R}(\Pi)y \quad \textit{iff} \quad (\exists i \in I)(x, y \in X_i),$$

is an equivalence relation whose equivalence classes are exactly the blocks X_i.

Proof. We leave this easy proof as an exercise to the reader. □

Putting Propositions 4.5.3 and 4.5.5 together we obtain the useful fact there is a bijection between $\mathrm{Equiv}(X)$ and $\mathrm{Part}(X)$. Therefore, in principle, it is a matter of taste whether we prefer to work with equivalence relations or partitions. In computer science, it is often preferable to work with partitions, but not always.

Proposition 4.5.6 *Given any set, X, the functions $\Pi\colon \mathrm{Equiv}(X) \to \mathrm{Part}(X)$ and $\mathcal{R}\colon \mathrm{Part}(X) \to \mathrm{Equiv}(X)$ are mutual inverses, that is,*

$$\mathcal{R} \circ \Pi = \mathrm{id} \quad \textit{and} \quad \Pi \circ \mathcal{R} = \mathrm{id}.$$

Consequently, there is a bijection between the set, $\mathrm{Equiv}(X)$, of equivalence relations on X and the set, $\mathrm{Part}(X)$, of partitions on X.

Proof. This is a routine verication left to the reader. □

Now, if $f\colon X \to Y$ is a surjective function, we have the equivalence relation, $\equiv_f$, defined by

$$x \equiv_f y \quad \text{iff} \quad f(x) = f(y).$$

It is clear that the equivalence class of any $x \in X$ is the inverse image, $f^{-1}(f(x))$, of $f(x) \in Y$. Therefore, there is a bijection between $X/\equiv_f$ and Y. Thus, we can identify f and the projection, π, from X onto $X/\equiv_f$. If f is not surjective, note that f is surjective onto $f(X)$ and so, we see that f can be written as the composition

$$f = i \circ \pi,$$

where $\pi\colon X \to f(X)$ is the canonical projection and $i\colon f(X) \to Y$ is the *inclusion function* mapping $f(X)$ into Y (i.e., $i(y) = y$, for every $y \in f(X)$).

Given a set, X, the inclusion ordering on $X \times X$ defines an ordering on binary relations on X, namely,

$$R \leq S \quad \text{iff} \quad (\forall x, y \in X)(xRy \Rightarrow xSy).$$

When $R \leq S$, we say that R *refines* S. If R and S are equivalence relations and $R \leq S$, we observe that every equivalence class of R is contained in some equivalence class of S. Actually, in view of Proposition 4.5.3, we see that *every equivalence class of S is the union of equivalence classes of R.* We also note that id_X is the least equivalence relation on X and $X \times X$ is the largest equivalence relation on X. This suggests the following question: Is $\mathrm{Equiv}(X)$ a lattice under refinement?

The answer is yes. It is easy to see that the meet of two equivalence relations is $R \cap S$, their intersection. But beware, their join is not $R \cup S$, because in general, $R \cup S$ is not transitive. However, there is a least equivalence relation containing R and S, and this is the join of R and S. This leads us to look at various closure properties of relations.

4.6 Transitive Closure, Reflexive and Transitive Closure, Smallest Equivalence Relation

Let R be any relation on a set X. Note that R is reflexive iff $\mathrm{id}_X \subseteq R$. Consequently, the smallest reflexive relation containing R is $\mathrm{id}_X \cup R$. This relation is called the *reflexive closure of R*.

Note that R is transitive iff $R \circ R \subseteq R$. This suggests a way of making the smallest transitive relation containing R (if R is not already transitive). Define R^n by induction as follows:

$$\begin{aligned} R^0 &= \mathrm{id}_X \\ R^{n+1} &= R^n \circ R. \end{aligned}$$

Definition 4.6.1 Given any relation, R, on a set, X, the *transitive closure of R* is the relation, R^+, given by

$$R^+ = \bigcup_{n \geq 1} R^n.$$

The *reflexive and transitive closure of R* is the relation, R^*, given by

$$R^* = \bigcup_{n \geq 0} R^n = \mathrm{id}_X \cup R^+.$$

The proof of the following proposition is left an an easy exercise.

Proposition 4.6.2 *Given any relation, R, on a set, X, the relation R^+ is the smallest transitive relation containing R and R^* is the smallest reflexive and transtive relation containing R.*

If R is reflexive, then it is easy to see that $R \subseteq R^2$ and so, $R^k \subseteq R^{k+1}$ for all $k \geq 0$. From this, we can show that if X is a finite set, then there is a smallest k so that $R^k = R^{k+1}$. In this case, R^k is the reflexive and transitive closure of R. If X has n elements it can be shown that $k \leq n - 1$.

Note that a relation, R, is symmetric iff $R^{-1} = R$. As a consequence, $R \cup R^{-1}$ is the smallest symmetric relation containing R. This relation is called the *symmetric closure of* R. Finally, given a relation, R, what is the smallest equivalence relation containing R? The answer is given by

Proposition 4.6.3 *For any relation, R, on a set, X, the relation*

$$(R \cup R^{-1})^*$$

is the smalest equivalence relation containing R.

4.7 Distributive Lattices, Boolean Algebras, Heyting Algebras

If we go back to one of our favorite examples of a lattice, namely, the power set, 2^X, of some set, X, we observe that it is more than a lattice. For example, if we look at Figure 4.4, we can check that the two identities D1 and D2 stated in the next definition hold.

Definition 4.7.1 We say that a lattice, X, is a *distributive lattice* if (D1) and (D2) hold:

$$\begin{array}{ll} D1 & a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \\ D2 & a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c). \end{array}$$

Remark: Not every lattice is distributive but many lattices of interest are distributive.

It is a bit surprising that in a lattice, (D1) and (D2) are actually equivalent, as we now show. Suppose (D1) holds, then

$$\begin{aligned} (a \vee b) \wedge (a \vee c) &= ((a \vee b) \wedge a) \vee ((a \vee b) \wedge c) && \text{(D1)} \\ &= a \vee ((a \vee b) \wedge c) && \text{(L4)} \\ &= a \vee ((c \wedge (a \vee b)) && \text{(L1)} \\ &= a \vee ((c \wedge a) \vee (c \wedge b)) && \text{(D1)} \\ &= a \vee ((a \wedge c) \vee (b \wedge c)) && \text{(L1)} \\ &= (a \vee (a \wedge c)) \vee (b \wedge c) && \text{(L2)} \\ &= ((a \wedge c) \vee a) \vee (b \wedge c) && \text{(L1)} \\ &= a \vee (b \wedge c) && \text{(L4)} \end{aligned}$$

which is (D2). Dually, (D2) implies (D1).

The reader should prove that every totally ordered poset is a distributive lattice. The lattice $\mathbb{N}_+$ under the divisibility ordering also turns out to be a distributive lattice.

Another useful fact about distributivity which is worth noting is that in any lattice

$$a \wedge (b \vee c) \geq (a \wedge b) \vee (a \wedge c).$$

This is because in any lattice, $a \wedge (b \vee c) \geq a \wedge b$ and $a \wedge (b \vee c) \geq a \wedge c$. Therefore, in order to establish associativity in a lattice it suffices to show that

$$a \wedge (b \vee c) \leq (a \wedge b) \vee (a \wedge c).$$

Another important property of distributive lattices is the following:

Proposition 4.7.2 *In a distributive lattice, X, if $z \wedge x = z \wedge y$ and $z \vee x = z \vee y$, then $x = y$ (for all $x, y, z \in X$).*

Proof. We have

$$\begin{aligned}
x &= (x \vee z) \wedge x && \text{(L4)}\\
&= x \wedge (z \vee x) && \text{(L1)}\\
&= x \wedge (z \vee y) && \\
&= (x \wedge z) \vee (x \wedge y) && \text{(D1)}\\
&= (z \wedge x) \vee (x \wedge y) && \text{(L1)}\\
&= (z \wedge y) \vee (x \wedge y) && \\
&= (y \wedge z) \vee (y \wedge x) && \text{(L1)}\\
&= y \wedge (z \vee x) && \text{(D1)}\\
&= y \wedge (z \vee y) && \\
&= (y \vee z) \wedge y && \text{(L1)}\\
&= y, && \text{(L4)}
\end{aligned}$$

that is, $x = y$, as claimed. □

The power set lattice has yet some additional properties having to do with complementation. First, the power lattice 2^X has a least element $0 = \emptyset$ and a greatest element, $1 = X$. If a lattice, X, has a least element, 0, and a greatest element, 1, the following properties are clear: For all $a \in X$, we have

$$\begin{aligned}
a \wedge 0 &= 0 \qquad & a \vee 0 &= a\\
a \wedge 1 &= a \qquad & a \vee 1 &= 1.
\end{aligned}$$

More importantly, for any subset, $A \subseteq X$, we have the complement, $\overline{A}$, of A in X, which satisfies the identities:

$$A \cup \overline{A} = X, \qquad A \cap \overline{A} = \emptyset.$$

Moreover, we know that the de Morgan identities hold. The generalization of these properties leads to what is called a complemented lattice.

Definition 4.7.3 Let X be a lattice and assume that X has a least element, 0, and a greatest element, 1 (we say that X is a *bounded lattice*). For any $a \in X$, a *complement of* a is any element, $b \in X$, so that

$$a \vee b = 1 \quad \text{and} \quad a \wedge b = 0.$$

If every element of X has a complement, we say that X is a *complemented lattice*.

Remarks:

1. When $0 = 1$, the lattice X collapses to the degenerate lattice consisting of a single element. As this lattice is of little interest, from now on, we will always assume that $0 \neq 1$.
2. In a complemented lattice, complements are generally not unique. However, as the next proposition shows, this is the case for distributive lattices.

Proposition 4.7.4 *Let X be a lattice with least element 0 and greatest element 1. If X is distributive, then complements are unique if they exist. Moreover, if b is the complement of a, then a is the complement of b.*

Proof. If a has two complements, b_1 and b_2, then $a \wedge b_1 = 0$, $a \wedge b_2 = 0$, $a \vee b_1 = 1$, and $a \vee b_2 = 1$. By commutativity, if follows that $b_1 \wedge a = b_2 \wedge a = 0$ and $b_1 \vee a = b_2 \vee a = 1$. By Proposition 4.7.2, we deduce that $b_1 = b_2$, that is, a has a unique complement.

By commutativity, the equations

$$a \vee b = 1 \quad \text{and} \quad a \wedge b = 0$$

are equivalent to the equations

$$b \vee a = 1 \quad \text{and} \quad b \wedge a = 0,$$

which shows that a is indeed a complement of b. By uniqueness, a is *the* complement of b. □

In view of Proposition 4.7.4, if X is a complemented distributive lattice, we denote the complement of any element, $a \in X$, by $\overline{a}$. We have the identities

$$\begin{aligned} a \vee \overline{a} &= 1 \\ a \wedge \overline{a} &= 0 \\ \overline{\overline{a}} &= a. \end{aligned}$$

We also have the following proposition about the de Morgan laws.

Proposition 4.7.5 *Let X be a lattice with least element 0 and greatest element 1. If X is distributive and complemented, then the de Morgan laws hold:*

$$\begin{aligned}\overline{a \vee b} &= \overline{a} \wedge \overline{b} \\ \overline{a \wedge b} &= \overline{a} \vee \overline{b}.\end{aligned}$$

Proof. We prove that

$$\overline{a \vee b} = \overline{a} \wedge \overline{b},$$

leaving the dual identity as an easy exercise. Using the uniqueness of complements, it is enough to check that $\overline{a} \wedge \overline{b}$ works, i.e., satisfies the conditions of Definition 4.7.3. For the first condition, we have

$$\begin{aligned}(a \vee b) \vee (\overline{a} \wedge \overline{b}) &= ((a \vee b) \vee \overline{a}) \wedge ((a \vee b) \vee \overline{b}) \\ &= (a \vee (b \vee \overline{a})) \wedge (a \vee (b \vee \overline{b})) \\ &= (a \vee (\overline{a} \vee b)) \wedge (a \vee 1) \\ &= ((a \vee \overline{a}) \vee b) \wedge 1 \\ &= (1 \vee b) \wedge 1 \\ &= 1 \wedge 1 = 1.\end{aligned}$$

For the second condition, we have

$$\begin{aligned}(a \vee b) \wedge (\overline{a} \wedge \overline{b}) &= (a \wedge (\overline{a} \wedge \overline{b})) \vee (b \wedge (\overline{a} \wedge \overline{b})) \\ &= ((a \wedge \overline{a}) \wedge \overline{b}) \vee (b \wedge (\overline{b} \wedge \overline{a})) \\ &= (0 \wedge \overline{b}) \vee ((b \wedge \overline{b}) \wedge \overline{a}) \\ &= 0 \vee (0 \wedge \overline{a}) \\ &= 0 \vee 0 = 0.\end{aligned}$$

□

All this leads to the definition of a boolean lattice

Definition 4.7.6 A *Boolean lattice* is a lattice with a least element, 0, a greatest element, 1, and which is distributive and complemented.

Of course, every power set is a boolean lattice, but there are boolean lattices that are not power sets. Putting together what we have done, we see that a boolean lattice is a set, X, with two special elements, 0, 1, and three operations, $\wedge$, $\vee$ and $a \mapsto \overline{a}$ satisfying the axioms stated in

Proposition 4.7.7 *If X is a boolean lattice, then the following equations hold for all $a, b, c \in X$:*

$$\begin{array}{lll}
L1 & a \vee b = b \vee a, & a \wedge b = b \wedge a \\
L2 & (a \vee b) \vee c = a \vee (b \vee c), & (a \wedge b) \wedge c = a \wedge (b \wedge c) \\
L3 & a \vee a = a, & a \wedge a = a \\
L4 & (a \vee b) \wedge a = a, & (a \wedge b) \vee a = a \\
D1\text{-}D2 & a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c), & a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \\
LE & a \vee 0 = a, & a \wedge 0 = 0 \\
GE & a \vee 1 = 1, & a \wedge 1 = a \\
C & a \vee \overline{a} = 1, & a \wedge \overline{a} = 0 \\
I & \overline{\overline{a}} = a & \\
dM & \overline{a \vee b} = \overline{a} \wedge \overline{b}, & \overline{a \wedge b} = \overline{a} \vee \overline{b}.
\end{array}$$

Conversely, if X is a set together with two special elements, 0, 1, and three operations, $\wedge$, $\vee$ and $a \mapsto \overline{a}$ satisfying the axioms above, then it is a boolean lattice under the ordering given by $a \leq b$ iff $a \vee b = b$.

In view of Proposition 4.7.7, we make the definition:

Definition 4.7.8 A set, X, together with two special elements, 0, 1, and three operations, $\wedge$, $\vee$ and $a \mapsto \overline{a}$ satisfying the axioms of Proposition 4.7.7 is called a *Boolean algebra*.

Proposition 4.7.7 shows that the notions of a Boolean lattice and of a Boolean algebra are equivalent. The first one is order-theoretic and the second one is algebraic.

Remarks:

1. As the name indicates, Boolean algebras were invented by G. Boole (1854). One of the first comprehensive accounts is due to E. Schröder (1890-1895).
2. The axioms for Boolean algebras given in Proposition 4.7.7 are not independent. There is a set of independent axioms known as the *Huntington axioms* (1933).

Let p be any integer with $p \geq 2$. Under the division ordering, it turns out that the set, $\mathrm{Div}(p)$, of divisors of p is a distributive lattice. In general not every integer, $k \in \mathrm{Div}(p)$, has a complement but when it does, $\overline{k} = p/k$. It can be shown that $\mathrm{Div}(p)$ is a Boolean algebra iff p is not divisible by any square integer (an integer of the form m^2, with $m > 1$).

Classical logic is also a rich source of Boolean algebras. Indeed, it is easy to show that logical equivalence is an equivalence relation and, as Homework problems, you have shown (with great pain) that all the axioms of Proposition 4.7.7 are provable equivalences (where

$\vee$ is disjunction, $\wedge$ is conjunction, $\overline{P} = \neg P$, i.e., negation, $0 = \perp$ and $1 = \top$). Furthermore, again, as a Homework problem, you have shown that logical equivalence is compatible with $\vee, \wedge, \neg$ in the following sense: If $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$, then

$$\begin{aligned} (P_1 \vee P_2) &\equiv (Q_1 \vee Q_2) \\ (P_1 \wedge P_2) &\equiv (Q_1 \wedge Q_2) \\ \neg P_1 &\equiv \neg Q_1. \end{aligned}$$

Consequently, for any set, T, of propositions we can define the relation, $\equiv_T$, by

$$P \equiv_T Q \quad \text{iff} \quad T \vdash P \equiv Q,$$

i.e., iff $P \equiv Q$ is provable from T (as explained in Section 1.9). Clearly, $\equiv_T$ is an equivalence relation on propositions and so, we can define the operations $\vee, \wedge$ and - on the set of equivalence classes, $\mathbf{B}_T$, of propositions as follows:

$$\begin{aligned} [P] \vee [Q] &= [P \vee Q] \\ [P] \wedge [Q] &= [P \wedge Q] \\ \overline{[P]} &= [\neg P]. \end{aligned}$$

We also let $0 = [\perp]$ and $1 = [\top]$. Then, we get the Boolean algebra, $\mathbf{B}_T$, called the *Lindenbaum algebra* of T.

It also turns out that Boolean algebras are just what's needed to give truth-value semantics to classical logic. Let B be any Boolean algebra. A *truth assignment* is any function, v, from the set $\mathbf{PS} = \{\mathbf{P}_1, \mathbf{P}_2, \cdots\}$ of propositional symbols to B. Then, we can evaluate recursively the truth value, $P_B[v]$, in B of any proposition, P, with respect to the truth assigment, v, as follows:

$$\begin{aligned} (\mathbf{P}_i)_B[v] &= v(P) \\ \perp_B [v] &= 0 \\ \top_B[v] &= 1 \\ (P \vee Q)_B[v] &= P_B[v] \vee P_B[v] \\ (P \wedge Q)_B[v] &= P_B[v] \wedge P_B[v] \\ (\neg P)_B[v] &= \overline{P[v]_B}. \end{aligned}$$

In the equations above, on the right hand side, $\vee$ and $\wedge$ are the lattice operations of the Boolean algebra, B. We say that a proposition, P, is *valid in the Boolean algebra B (or B-valid)* if $P_B[v] = 1$ for all truth assigments, v. We say that P is *(classially) valid* if P is B-valid in all Boolean algebras, B. It can be shown that every provable proposition is valid. This property is called *soundness*. Conversely, if P is valid, then it is provable. This second property is called *completeness*. Actually completeness holds in a much stronger sense: If a proposition is valid in the two element Boolean algebra, $\{0, 1\}$, then it is provable!

One might wonder if there are certain kinds of algebras similar to Boolean algebras well suited for intuitionistic logic. The answer is yes: Such algebras are called *Heyting algebras*.

In our study of intuitionistic logic, we learned that negation is not a primary connective but instead it is defined in terms of implication by $\neg P = P \Rightarrow \perp$. This suggests adding to the two lattice operations $\vee$ and $\wedge$ a new operation, $\rightarrow$, that will behave like $\Rightarrow$. The trick is, what kind of axioms should we require on $\rightarrow$ to "capture" the properties of intuitionistic logic? Now, if X is a lattice with 0 and 1, given any two elements, $a, b \in X$, experience shows that $a \rightarrow b$ should be the largest element, c, such that $c \wedge a \leq b$. This leads to

Definition 4.7.9 A lattice, X, with 0 and 1 is a *Heyting lattice* iff it has a third binary operation, $\rightarrow$, such that

$$c \wedge a \leq b \quad \text{iff} \quad c \leq (a \rightarrow b)$$

for all $a, b, c \in X$. We define the *negation (or pseudo-complement) of* a as $\overline{a} = (a \rightarrow 0)$.

At first glance, it is not clear that a Heyting lattice is distributive but in fact, it is. The following proposition (stated without proof) gives an algebraic characterization of Heyting lattices which is useful to prove various properties of Heyting lattices.

Proposition 4.7.10 *Let X be a lattice with* 0 *and* 1 *and with a binary operation,* $\rightarrow$*. Then, X is a Heyting lattice iff the following equations hold for all $a, b, c \in X$:*

$$\begin{aligned} a \rightarrow a &= 1 \\ a \wedge (a \rightarrow b) &= a \wedge b \\ b \wedge (a \rightarrow b) &= b \\ a \rightarrow (b \wedge c) &= (a \rightarrow b) \wedge (a \rightarrow c). \end{aligned}$$

A lattice with 0 and 1 and with a binary operation, $\rightarrow$, satisfying the equations of Proposition 4.7.10 is called a *Heyting algebra*. So, we see that Proposition 4.7.10 shows that the notions of Heyting lattice and Heyting algebra are equivalent (this is analogous to Boolean lattices and Boolean algebras).

The reader will notice that these axioms are propositions that were shown to be provable intuitionistically in Homework Problems! The proof of Proposition 4.7.10 is not really difficult but it is a bit tedious so we will omit it. Let us simply show that the fourth equation implies that for any fixed $a \in X$, the map $b \mapsto (a \rightarrow b)$ is monotonic. So, assume $b \leq c$, i.e., $b \wedge c = b$. Then, we get

$$a \rightarrow b = a \rightarrow (b \wedge c) = (a \rightarrow b) \wedge (a \rightarrow c),$$

which means that $(a \rightarrow b) \leq (a \rightarrow c)$, as claimed.

The following theorem shows that every Heyting algebra is distributive, as we claimed earlier. This theorem also shows "how close" to a Boolean algebra a Heyting algebra is.

Theorem 4.7.11 *(a) Every Heyting algebra is distributive.*

(b) A Heyting algebra, X, is a boolean algebra iff $\overline{\overline{a}} = a$ for all $a \in X$.

Proof. (a) From a previous remark, to show distributivity, it is enough to show the inequality

$$a \wedge (b \vee c) \leq (a \wedge b) \vee (a \wedge c).$$

Observe that from the property characterizing $\rightarrow$, we have

$$b \leq a \rightarrow (a \wedge b) \quad \text{iff} \quad b \wedge a \leq a \wedge b$$

which holds, by commutativity of $\wedge$. Thus, $b \leq a \rightarrow (a \wedge b)$ and similarly, $c \leq a \rightarrow (a \wedge c)$.

Recall that for any fixed a, the map $x \mapsto (a \rightarrow x)$ is monotonic. Since $a \wedge b \leq (a \wedge b) \vee (a \wedge c)$ and $a \wedge c \leq (a \wedge b) \vee (a \wedge c)$, we get

$$a \rightarrow (a \wedge b) \leq a \rightarrow ((a \wedge b) \vee (a \wedge c)) \quad \text{and} \quad a \rightarrow (a \wedge c) \leq a \rightarrow ((a \wedge b) \vee (a \wedge c)).$$

These two inequalities imply $(a \rightarrow (a \wedge b)) \vee (a \rightarrow (a \wedge c)) \leq a \rightarrow ((a \wedge b) \vee (a \wedge c))$, and since we also have $b \leq a \rightarrow (a \wedge b)$ and $c \leq a \rightarrow (a \wedge c)$, we deduce that

$$b \vee c \leq a \rightarrow ((a \wedge b) \vee (a \wedge c)),$$

which, using the fact that $(b \vee c) \wedge a = a \wedge (b \vee c)$, means that

$$a \wedge (b \vee c) \leq (a \wedge b) \vee (a \wedge c),$$

as desired.

(b) We leave this part as an exercise. The trick is to see that the de Morgan laws hold and to apply one of them to $a \wedge \overline{a} = 0$. □

Remarks:

1. Heyting algebras were invented by A. Heyting in 1930. Heyting algebras are sometimes known as "Brouwerian lattices".

2. Every Boolean algebra is automatically a Heyting algebra: Set $a \rightarrow b = \overline{a} \vee b$.

3. It can be shown that every finite distributive lattice is a Heyting algebra.

We conclude this brief exposition of Heyting algebras by explaining how they provide a truth semantics for intuitionistic logic analogous to the thuth semantics that Boolean algebras provide for classical logic.

As in the classical case, it is easy to show that intuitionistic logical equivalence is an equivalence relation and you have shown (with great pain) that all the axioms of Heyting algebras are intuitionistically provable equivalences (where $\vee$ is disjunction, $\wedge$ is conjunction, and $\rightarrow$ is $\Rightarrow$). Furthermore, you have also shown that intuitionistic logical equivalence is compatible with $\vee, \wedge, \Rightarrow$ in the following sense: If $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$, then

$$\begin{aligned}(P_1 \vee P_2) &\equiv (Q_1 \vee Q_2)\\ (P_1 \wedge P_2) &\equiv (Q_1 \wedge Q_2)\\ (P_1 \Rightarrow P_2) &\equiv (Q_1 \Rightarrow Q_2).\end{aligned}$$

Consequently, for any set, T, of propositions we can define the relation, $\equiv_T$, by

$$P \equiv_T Q \quad \text{iff} \quad T \vdash P \equiv Q,$$

i.e., iff $P \equiv Q$ is provable intuitionistically from T (as explained in Section 1.9). Clearly, $\equiv_T$ is an equivalence relation on propositions and we can define the operations $\vee, \wedge$ and $\rightarrow$ on the set of equivalence classes, $\mathbf{H}_T$, of propositions as follows:

$$\begin{aligned}[P] \vee [Q] &= [P \vee Q]\\ [P] \wedge [Q] &= [P \wedge Q]\\ [P] \rightarrow [Q] &= [P \Rightarrow Q].\end{aligned}$$

We also let $0 = [\perp]$ and $1 = [\top]$. Then, we get the Heyting algebra, $\mathbf{H}_T$, called the *Lindenbaum algebra* of T, as in the classical case.

Now, let H be any Heyting algebra. By analogy with the case of Boolean algebras, a *truth assignment* is any function, v, from the set $\mathbf{PS} = \{\mathbf{P}_1, \mathbf{P}_2, \cdots\}$ of propositional symbols to H. Then, we can evaluate recursively the truth value, $P_H[v]$, in H of any proposition, P, with respect to the truth assigment, v, as follows:

$$\begin{aligned}(\mathbf{P}_i)_H[v] &= v(P)\\ \perp_H [v] &= 0\\ \top_H[v] &= 1\\ (P \vee Q)_H[v] &= P_H[v] \vee P_H[v]\\ (P \wedge Q)_H[v] &= P_H[v] \wedge P_H[v]\\ (P \Rightarrow Q)_H[v] &= (P_H[v] \rightarrow P_H[v])\\ (\neg P)_H[v] &= (P_H[v] \rightarrow 0).\end{aligned}$$

In the equations above, on the right hand side, $\vee$, $\wedge$ and $\rightarrow$ are the operations of the Heyting algebra, H. We say that a proposition, P, is *valid in the Heyting algebra H (or H-valid)* if $P_H[v] = 1$ for all truth assigments, v. We say that P is *HA-valid (or intuitionistically valid)* if P is H-valid in all Heyting algebras, H. As in the classical case, it can be shown that every intuitionistically provable proposition is HA-valid. This property is called *soundness*. Conversely, if P is HA-valid, then it is intuitionistically provable. This second property is called *completeness*. A stronger completeness result actually holds: If a proposition is H-valid in all *finite* Heyting algebras, H, then it is intuitionistically provable. As a consequence, if a proposition is *not* provable intuitionistically, then it can be falsified in some finite Heyting algebra.

Remark: If X is any set, a *topology on* X is a family, $\mathcal{O}$, of subsets of X satisfying the following conditions:

(1) $\emptyset \in \mathcal{O}$ and $X \in \mathcal{O}$;

(2) For every family (even infinite), $(U_i)_{i \in I}$, of sets $U_i \in \mathcal{O}$, we have $\bigcup_{i \in I} U_i \in \mathcal{O}$.

(3) For every *finite* family, $(U_i)_{1 \leq i \leq n}$, of sets $U_i \in \mathcal{O}$, we have $\bigcap_{1 \leq i \leq n} U_i \in \mathcal{O}$.

Every subset in $\mathcal{O}$ is called an *open subset* of X (in the topology $\mathcal{O}$) . The pair, $\langle X, \mathcal{O} \rangle$, is called a *topological space*. Given any subset, A, of X, the union of all open subsets contained in A is the largest open subset of A and is denoted $\overset{\circ}{A}$.

Given a topological space, $\langle X, \mathcal{O} \rangle$, we claim that $\mathcal{O}$ with the inclusion ordering is a Heyting algebra with $0 = \emptyset$; $1 = X$; $\vee = \cup$ (union); $\wedge = \cap$ (intersection); and with

$$(U \rightarrow V) = \overbrace{(X - U) \cup V}^{\circ}.$$

(Here, $X - U$ is the complement of U in X.) In this Heyting algebra, we have

$$\overline{U} = \overbrace{X - U}^{\circ}.$$

Since $X - U$ is usually not open, we generally have $\overline{\overline{U}} \neq U$. Therefore, we see that topology yields another supply of Heyting algebras.

Chapter 5

Graphs, Basic Notions

5.1 Why Graphs? Some Motivations

Graphs are mathematical structures that have many applications to computer science, electrical engineering and more widely to engineering as a whole, but also to sciences such as biology, linguistics, and sociology, among others. For example, relations among objects can usually be encoded by graphs. Whenever a system has a notion of state and state transition function, graph methods may be applicable. Certain problems are naturally modeled by undirected graphs whereas others require directed graphs. Let us give a concrete example.

Suppose a city decides to create a public-transportation system. It would be desirable if this system allowed transportation between certain locations considered important. Now, if this system consists of buses, the traffic will probably get worse so the city engineers decide that the traffic will be improved by making certain streets one-way streets. The problem then is, given a map of the city consisting of the important locations and of the two-way streets linking them, find an orientation of the streets so that it is still possible to travel between any two locations. The problem requires finding a directed graph, given an undirected graph. Figure 5.1 shows the undirected graph corresponding to the city map and Figure 5.2 shows a proposed choice of one-way streets. Did the engineers do a good job or are there locations such that it is impossible to travel from one to the other while respecting the one-way signs?

The answer to this puzzle will be revealed in Section 5.3.

There is a peculiar aspect of graph theory having to do with its terminology. Indeed, unlike most branches of mathematics, it appears that the terminology of graph theory is not standardized, yet. This can be quite confusing to the beginner who has to struggle with many different and often inconsistent terms denoting the same concept, one of the worse being the notion of a *path*. Our attitude has been to use terms that we feel are as simple as possible. As a result, we have not followed a single book. Among the many books on graph theory, we have been inspired by the classic texts, Harary [29], Berge [3] and Bollobas [7]. This chapter on graphs is heavily inspired by Sakarovitch [38], because we find Sakarovitch's book extremely clear and because it has more emphasis on applications than the previous

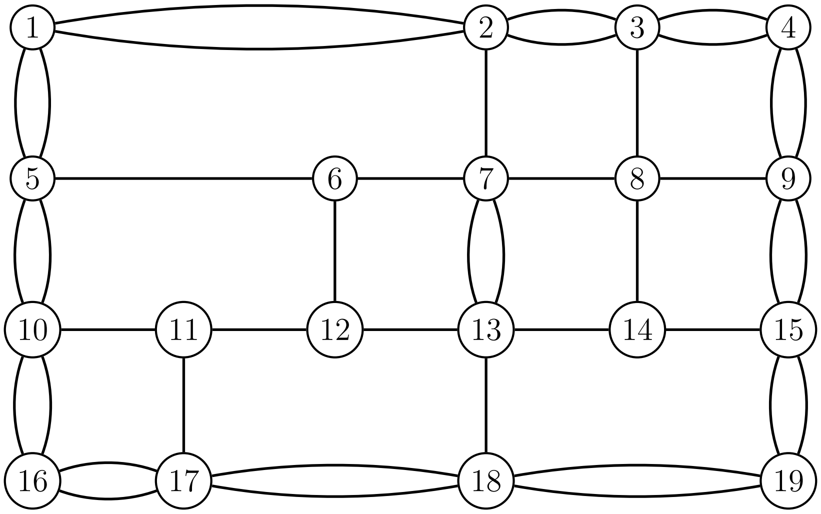

Figure 5.1: An undirected graph modeling a city map

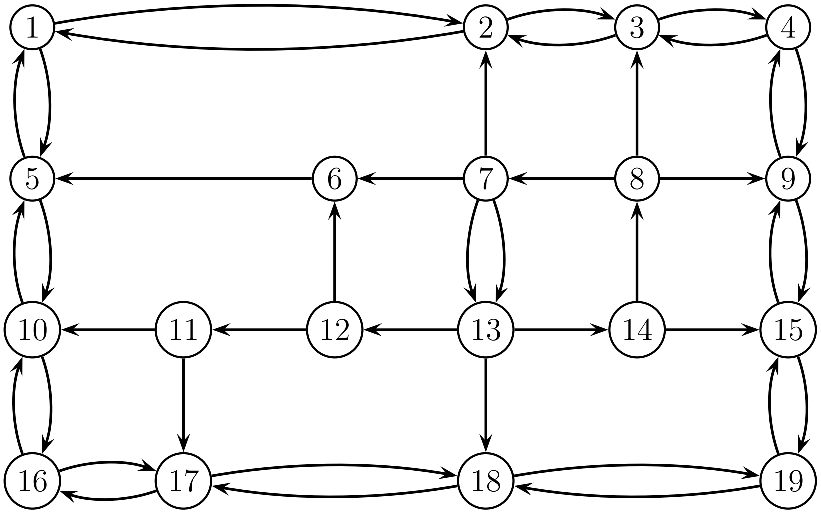

Figure 5.2: A choice of one-way streets

two. Another more recent (and more advanced) text which is also excellent is Diestel [14].

Many books begin by discussing undirected graphs and introduce directed graph only later on. We disagree with this approach. Indeed, we feel that the notion of a directed graph is more fundamental than the notion of an undirected graph. For one thing, a unique undirected graph is obtained from a directed graph by forgetting the direction of the arcs, whereas there are many ways of orienting an undirected graph. Also, in general, we believe that most definitions about directed graphs are cleaner than the corresponding ones for undirected graphs (for instance, we claim that the definition of a directed graph is simpler than the definition of an undirected graph, and similarly for paths). Thus, we begin with directed graphs.

5.2 Directed Graphs

Informally, a directed graph consists of a set of nodes together with a set of oriented arcs (also called edges) between these nodes. Every arc has a single source (or initial point) and a single target (or endpoint), both of which are nodes. There are various ways of formalizing what a directed graph is and some decisions must be made. Two issues must be confronted:

1. Do we allow "loops," that is, arcs whose source and target are identical?
2. Do we allow "parallel arcs," that is distinct arcs having the same source and target?

Since every binary relation on a set can be represented as a directed graph with loops, our definition allows loops. Since the directed graphs used in automata theory must accomodate parallel arcs (usually labeled with different symbols), our definition also allows parallel arcs. Thus, we choose a more inclusive definition in order to accomodate as many applications as possible, even though some authors place restrictions on the definition of a graph, for example, forbidding loops and parallel arcs (we will call such graphs, simple graphs). Before giving a formal definition, let us say that graphs are usually depicted by drawings (graphs!) where the nodes are represented by circles containing the node name and oriented line segments labeled with their arc name (see Figure 5.3).

Definition 5.2.1 A *directed graph* (or *digraph*) is a quadruple, $G = (V, E, s, t)$, where V is a set of *nodes or vertices*, E is a set of *arcs or edges* and $s, t\colon E \to V$ are two functions, s being called the *source function* and t the *target function*. Given an edge $e \in E$, we also call $s(e)$ the *origin* or *source* of e, and $t(e)$ the *endpoint* or *target* of e.

If the context makes it clear that we are dealing only with directed graphs, we usually say simply "graph" instead of "directed graph". A directed graph, $G = (V, E, s, t)$, is *finite* iff both V and E are finite. In this case, $|V|$, the number of nodes of G is called the *order* of G.

Example: Let G_1 be the directed graph defined such that

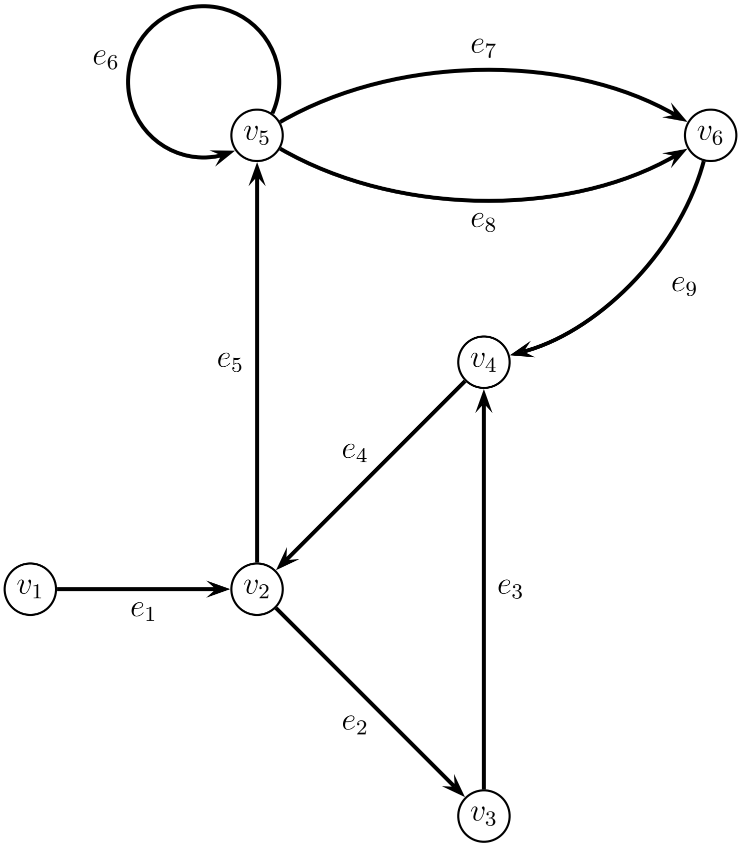

Figure 5.3: A directed graph, G_1

$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$,
$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, and

$$\begin{aligned}
&s(e_1) = v_1, s(e_2) = v_2, s(e_3) = v_3, s(e_4) = v_4,\\
&s(e_5) = v_2, s(e_6) = v_5, s(e_7) = v_5, s(e_8) = v_5, s(e_9) = v_6\\
&t(e_1) = v_2, t(e_2) = v_3, t(e_3) = v_4, t(e_4) = v_2,\\
&t(e_5) = v_5, t(e_6) = v_5, t(e_7) = v_6, t(e_8) = v_6, t(e_9) = v_4.
\end{aligned}$$

The graph G_1 is represented by the diagram shown in Figure 5.3.

It should be noted that there are many different ways of "drawing" a graph. Obviously, we would like as much as possible to avoid having too many intersecting arrows but this is not always possible if we insist in drawing a graph on a sheet of paper (on the plane).

Definition 5.2.2 Given a directed graph, G, an edge, $e \in E$, such that $s(e) = t(e)$ is called a *loop* (or *self-loop*). Two edges, $e, e' \in E$ are said to be *parallel edges* iff $s(e) = s(e')$ and $t(e) = t(e')$. A directed graph is *simple* iff it has no parallel edges.

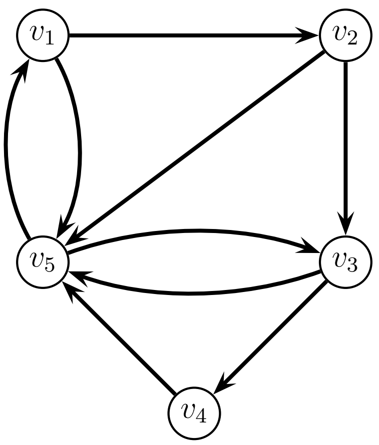

Figure 5.4: A directed graph, G_2

Remarks:

1. The functions s, t need not be injective or surjective. Thus, we allow "isolated vertices", that is, vertices that are not the source or the target of any edge.

2. When G is simple, every edge, $e \in E$, is uniquely determined by the ordered pair of vertices, (u, v), such that $u = s(e)$ and $v = t(e)$. In this case, we may denote the edge e by (uv) (some books also use the notation uv). Also, a graph without parallel edges can be defined as a pair, (V, E), with $E \subseteq V \times V$. In other words, a simple graph is equivalent to a binary relation on a set ($E \subseteq V \times V$). This definition is often the one used to define directed graphs.

3. Given any edge, $e \in E$, the nodes $s(e)$ and $t(e)$ are often called the *boundaries* of e and the expression $t(e) - s(e)$ is called the *boundary of* e.

4. Given a graph, $G = (V, E, s, t)$, we may also write $V(G)$ for V and $E(G)$ for E. Sometimes, we even drop s and t and write simply $G = (V, E)$ instead of $G = (V, E, s, t)$.

5. Some authors define a simple graph to be a graph without loops and without parallel edges.

Observe that the graph G_1 has the loop e_6 and the two parallel edges e_7 and e_8. When we draw pictures of graphs, we often omit the edge names (sometimes even the node names) as illustrated in Figure 5.4.

Definition 5.2.3 Given a directed graph, G, for any edge $e \in E$, if $u = s(e)$ and $v = t(e)$, we say that

(i) The nodes u and v are *adjacent*

(ii) The nodes u and v are *incident to the arc* e

(iii) The arc e is *incident to the nodes* u *and* v

(iv) Two edges, $e, e' \in E$ are *adjacent* if they are incident to some common node (that is, either $s(e) = s(e')$ or $t(e) = t(e')$ or $t(e) = s(e')$ or $s(e) = t(e')$).

For any node, $u \in V$, set

(a) $d^+_G(u) = |\{e \in E \mid s(e) = u\}|$, the *outer half-degree or outdegree of* u

(b) $d^-_G(u) = |\{e \in E \mid t(e) = u\}|$, the *inner half-degree or indegree of* u

(c) $d_G(u) = d^+_G(u) + d^-_G(u)$, the *degree of* u.

A graph is *regular* iff every node has the same degree.

Note that d^+_G (respectively $d^-_G(u)$) counts the number of arcs "coming out from u", that is, whose source is u (resp. counts the number of arcs "coming into u", that is, whose target is u). For example, in the graph of Figure 5.4, $d^+_{G_2}(v_1) = 2$, $d^-_{G_2}(v_1) = 1$, $d^+_{G_2}(v_5) = 2$, $d^-_{G_2}(v_5) = 4$, $d^+_{G_2}(v_3) = 2$, $d^-_{G_2}(v_3) = 2$. Neither G_1 nor G_2 are regular graphs.

The first result of graph theory is the following simple but very useful proposition:

Proposition 5.2.4 *For any finite graph, $G = (V, E, s, t)$, we have*

$$\sum_{u\in V} d^+_G(u) = \sum_{u\in V} d^-_G(u).$$

Proof. Every arc, $e \in E$, has a single source and a single target and each side of the above equations simply counts the number of edges in the graph. □

Corollary 5.2.5 *For any finite graph, $G = (V, E, s, t)$, we have*

$$\sum_{u\in V} d_G(u) = 2|E|,$$

that is, the sum of the degrees of all the nodes is equal to twice the number of edges.

Corollary 5.2.6 *For any finite graph, $G = (V, E, s, t)$, there is an even number of nodes with an odd degree.*

The notion of homomorphism and isomorphism of graphs is fundamental.

Definition 5.2.7 Given two directed graphs, $G_1 = (V_1, E_1, s_1, t_1)$ and $G_2 = (V_2, E_2, s_2, t_2)$, a *homomorphism* (or *morphism*), $f\colon G_1 \to G_2$, *from* G_1 *to* G_2 is a pair, $f = (f^v, f^e)$, with $f^v\colon V_1 \to V_2$ and $f^e\colon E_1 \to E_2$ preserving incidence, that is, for every edge, $e \in E_1$, we have

$$s_2(f^e(e)) = f^v(s_1(e)) \quad \text{and} \quad t_2(f^e(e)) = f^v(t_1(e)).$$

These conditions can also be expressed by saying that the following two diagrams commute:

$$\begin{tikzcd} E_1 \arrow[r, "f^e"] \arrow[d, "s_1"'] & E_2 \arrow[d, "s_2"] \\ V_1 \arrow[r, "f^v"'] & V_2 \end{tikzcd} \qquad \begin{tikzcd} E_1 \arrow[r, "f^e"] \arrow[d, "t_1"'] & E_2 \arrow[d, "t_2"] \\ V_1 \arrow[r, "f^v"'] & V_2. \end{tikzcd}$$

Given three graphs, G_1, G_2, G_3 and two homomorphisms, $f\colon G_1 \to G_2$ and $g\colon G_2 \to G_3$, with $f = (f^v, f^e)$ and $g = (g^v, g^e)$, it is easily checked that $(g^v \circ f^v, g^e \circ f^e)$ is a homomorphism from G_1 to G_3. The homomorphism $(g^v \circ f^v, g^e \circ f^e)$ is denoted $g \circ f$. Also, for any graph, G, the map $\mathrm{id}_G = (\mathrm{id}_V, \mathrm{id}_E)$ is a homomorphism called the *identity homomorphism*. Then, a homomorphism, $f\colon G_1 \to G_2$, is an *isomorphism* iff there is a homomorphism, $g\colon G_2 \to G_1$, such that

$$g \circ f = \mathrm{id}_{G_1} \quad \text{and} \quad f \circ g = \mathrm{id}_{G_2}.$$

In this case, g is unique and it is called the *inverse* of f and denoted f^{-1}. If $f = (f^v, f^e)$ is an isomorphism, we see immediately that f^v and f^e are bijections. Checking whether two finite graphs are isomorphic is not as easy as it looks. In fact, no general efficient algorithm for checking graph isomorphism is known at this time and determining the exact complexity of this problem is a major open question in computer science. For example, the graphs G_3 and G_4 shown in Figure 5.5 are isomorphic. The bijection f^v is given by $f^v(v_i) = w_i$, for $i = 1, \ldots, 6$ and the reader will easily figure out the bijection on arcs. As we can see, isomorphic graphs can look quite different.

5.3 Paths in Digraphs; Strongly Connected Components

Many problems about graphs can be formulated as path existence problems. Given a directed graph, G, intuitively, a path from a node u to a node v is a way to travel from u in v by following edges of the graph that "link up correctly". Unfortunately, if we look up the definition of a path in two different graph theory books, we are almost guaranteed to find different and usually clashing definitions! This has to do with the fact that for some authors, a path may not use the same edge more than once and for others, a path may not pass through the same node more than once. Moreover, when parallel edges are present (*i.e.*, when a graph is not simple), a sequence of nodes does not define a path unambiguously!.

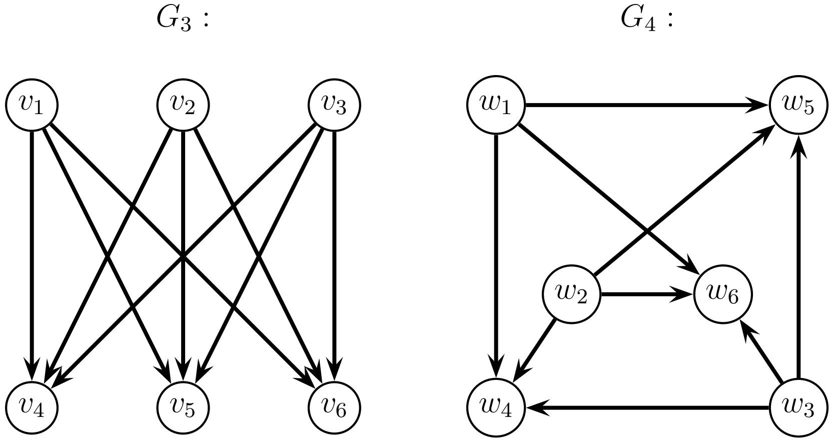

Figure 5.5: Two isomorphic graphs, G_3 and G_4

The terminology that we have chosen may not be standard, but it is used by a number of authors (some very distinguished, for example, Fields medalists!) and we believe that it is less taxing on one's memory (however, this point is probably the most debatable).

Definition 5.3.1 Given any digraph, $G = (V, E, s, t)$, and any two nodes, $u, v \in V$, a *path from u to v* is a triple, $\pi = (u, e_1 \cdots e_n, v)$, where $n \geq 1$ and $e_1 \cdots e_n$ is a sequence of edges, $e_i \in E$ (i.e., a nonempty string in E^*), such that

$$s(e_1) = u;\ t(e_n) = v;\ t(e_i) = s(e_{i+1}),\ 1 \leq i \leq n - 1.$$

We call n the *length of the path* π and we write $|\pi| = n$. When $n = 0$, we have the *null path*, (u, ϵ, u), from u to u (recall, ϵ denotes the empty string); the null path has length 0. If $u = v$, then π is called a *closed path*, else an *open path*. The path, $\pi = (u, e_1 \cdots e_n, v)$, determines the sequence of nodes, $\text{nodes}(\pi) = \langle u_0, \ldots, u_n \rangle$, where $u_0 = u$, $u_n = v$ and $u_i = t(e_i)$, for $1 \leq i \leq n - 1$. We also set $\text{nodes}((u, \epsilon, u)) = \langle u, u \rangle$. A path, $\pi = (u, e_1 \cdots e_n, v)$, is *simple* iff $e_i \neq e_j$ for all $i \neq j$ (i.e., no edge in the path is used twice). A path, π, from u to v is *elementary* iff no vertex in $\text{nodes}(\pi)$ occurs twice, except possibly for u if π is closed. Equivalently, if $\text{nodes}(\pi) = \langle u_0, \ldots, u_n \rangle$, then π is elementary iff either

1. $u_i \neq u_j$ for all i, j with $i \neq j$ and $0 \leq i, j \leq n$, or π is closed ($u_0 = u_n$) in which case

2. $u_i \neq u_0$ and $u_i \neq u_n$ for all i with $1 \leq i \leq n - 1$, and $u_i \neq u_j$ for all i, j with $i \neq j$ and $1 \leq i, j \leq n - 1$.

The null path, (u, ϵ, u), is considered simple and elementary.

Remarks:

1. Other authors use the term *walk* for what we call a path. These authors also use the term *trail* for what we call a simple path and the term *path* for what we call an elementary path.

2. If a digraph is not simple, then even if a sequence of nodes is of the form nodes(π) for some path, that sequence of nodes does not uniquely determine a path. For example, in the graph of Figure 5.3, the sequence $\langle v_2, v_5, v_6\rangle$ corresponds to the two distinct paths (v_2, e_5e_7, v_6) and (v_2, e_5e_8, v_6).

In the graph G_1 from Figure 5.3,

$$(v_2, e_5e_7e_9e_4e_5e_8, v_6)$$

is a path from v_2 to v_6 which is neither simple nor elementary,

$$(v_2, e_2e_3e_4e_5, v_5)$$

is a simple path from v_2 to v_5 which is not elementary and

$$(v_2, e_5e_7e_9, v_4), \qquad (v_2, e_5e_7e_9e_4, v_2)$$

are elementary paths, the first one open and the second one closed.

Recall the notion of subsequence of a sequence defined just before stating Theorem 2.9.8. Then, if $\pi = (u, e_1 \cdots e_n, v)$ is any path from u to v in a digraph, G, a *subpath* of π is any path $\pi' = (u, e'_1 \cdots e'_m, v)$ such that $e'_1, \ldots, e'_m$ is a subsequence of $e_1, \ldots, e_n$. The following simple proposition is actually very important:

Proposition 5.3.2 *Let G be any digraph. (a) For any two nodes, u, v, in G, every non-null path, π, from u to v contains an elementary non-null subpath.*

(b) If $|V| = n$, then every open elementary path has length at most $n-1$ and every closed elementary path has length at most n.

Proof. (a) Let π be any non-null path from u to v in G and let

$$S = \{k \in \mathbb{N} \mid k = |\pi'|, \quad \pi' \text{ is a non-null subpath of } \pi\}.$$

The set $S \subseteq \mathbb{N}$ is nonempty since $|\pi| \in S$ and as $\mathbb{N}$ is well-ordered, S has a least element, say $m \geq 1$. We claim that any subpath of π of length m is elementary. Consider any such path, say $\pi' = (u, e'_1 \cdots e'_m, v)$, let

$$\text{nodes}(\pi') = \langle v_0, \ldots, v_m\rangle,$$

and assume that π' is not elementary. There are two cases:

(1) $u \neq v$. Then, some node occurs twice in nodes(π'), say $v_i = v_j$, with $i < j$. Then, we can delete the path $(v_i, e'_{i+1} \cdots, e'_j, v_j)$ from π' to obtain a non-null (because $u \neq v$) subpath π'' of π' from u to v with $|\pi''| = |\pi'| - (j - i)$ and since $i < j$, we see that $|\pi''| < |\pi'|$, contradicting the minimality of m. Therefore, π' is a non-null elementary subpath of π.

(2) $u = v$. In this case, either some node occurs twice in the sequence $\langle v_0, \ldots, v_{m-1}\rangle$ or some node occurs twice in the sequence $\langle v_1, \ldots, v_m\rangle$. In either case, as in (1), we can strictly shorten the path from v_0 to v_{m-1} or the path from v_1 to v_m. Even though the resulting path may be the null path, as one of two two edges e_1' or e_m' remains from the original path π', we get a non-null path from u to u strictly shorter than π', contradicting the minimality of π'.

(b) As in (a), let π' be an open elementary path from u to v and let

$$\mathrm{nodes}(\pi') = \langle v_0, \ldots, v_m\rangle.$$

If $m \geq n = |V|$, as the above sequence has $m+1 > n$ nodes, by the Pigeonhole Principle, some node must occur twice, contradicting the fact that π' is an open elementary path. If π' is a non-null closed path and $m \geq n+1$, then by the Pigeonhole Principle, either some node occurs twice in $\langle v_0, \ldots, v_m\rangle$ or some node occurs twice in $\langle v_1, \ldots, v_{m+1}\rangle$. In either case, this contradicts the fact that π' is a non-null elementary path. □

Like strings, paths can be concatenated.

Definition 5.3.3 Two paths, $\pi = (u, e_1 \cdots e_m, v)$ and $\pi' = (u', e_1' \cdots e_n', v')$ in a digraph G can be *concatenated* iff $v = u'$ in which case their *concatenation*, $\pi\pi'$, is the path

$$\pi\pi' = (u, e_1 \cdots e_m e_1' \cdots e_n', v').$$

We also let

$$(u, \epsilon, u)\pi = \pi = \pi(v, \epsilon, v).$$

Concatenation of paths is obviously associative and observe that $|\pi\pi'| = |\pi| + |\pi'|$.

Definition 5.3.4 Let $G = (V, E, s, t)$ be a digraph. We define the binary relation, $\widehat{C}_G$, on V as follows: For all $u, v \in V$,

$$u\widehat{C}_G v \quad \text{iff} \quad \text{there is a path from } u \text{ to } v \text{ and there is a path from } v \text{ to } u.$$

When $u\widehat{C}_G v$, we say that *u and v are strongly connected*.

Oberve that the relation $\widehat{C}_G$ is an equivalence relation. It is reflexive because we have the null path from u to u, symmetric by definition, and transitive because paths can be concatenated. The equivalence classes of the relation $\widehat{C}_G$ are called the *strongly connected components of G (SCC's)*. A graph is *strongly connected* iff it has a single strongly connected component.

For example, we see that the graph, G_1, of Figure 5.3 has two strongly connected components

$$\{v_1\}, \quad \{v_2, v_3, v_4, v_5, v_6\},$$

since there is a closed path

$$(v_4, e_4e_2e_3e_4e_5e_7e_9, v_4).$$

The graph G_2 of Figure 5.4 is strongly connected.

Let us give a simple algorithm for computing the strongly connected components of a graph since this is often the key to solving many problems. The algorithm works as follows: Given some vertex, $u \in V$, the algorithm computes the two sets, $X^+(u)$ and $X^-(u)$, where

$$\begin{aligned} X^+(u) &= \{v \in V \mid \text{there exists a path from } u \text{ to } v\} \\ X^-(u) &= \{v \in V \mid \text{there exists a path from } v \text{ to } u\}. \end{aligned}$$

Then, it is clear that the connected component, $C(u)$, or u, is given by $C(u) = C^+(u) \cap X^-(u)$. For simplicity, we assume that $X^+(u), X^-(u)$ and $C(u)$ are represented by linear arrays. In order to make sure that the algorithm makes progress, we used a simple marking scheme. We use the variable *total* to count how many nodes are in $X^+(u)$ (or in $X^-(u)$) and the variable *marked* to keep track of how many nodes in $X^+(u)$ (or in $X^-(u)$) have been processed so far. Whenever the algorithm considers some unprocessed node, the first thing it does is to increment *marked* by 1. Here is the algorithm in high-level form.

```
function strcomp($G$: graph; $u$: node): set
  begin
    $X^+(u)[1] := u$; $X^-(u)[1] := u$; $total := 1$; $marked := 0$;
    while $marked < total$ do
      $marked := marked + 1$; $v := X^+(u)[marked]$;
      for each $e \in E$
        if $(s(e) = v) \wedge (t(e) \notin X^+(u))$ then
          $total := total + 1$; $X^+(u)[total] := t(e)$ endif
      endfor
    endwhile;
    $total := 1$; $marked := 0$;
    while $marked < total$ do
      $marked := marked + 1$; $v := X^-(u)[marked]$;
      for each $e \in E$
        if $(t(e) = v) \wedge (s(e) \notin X^-(u))$ then
          $total := total + 1$; $X^-(u)[total] := s(e)$ endif
      endfor
    endwhile;
    $C(u) = X^+(u) \cap X^-(u)$; $strcomp := C(u)$
  end
```

If we want to obtain all the strongly connected components (SCC's) of a finite graph, G, we proceed as follows: Set $V_1 = V$, pick any node, v_1, in V_1 and use the above algorithm to compute the strongly connected component, C_1, of v_1. If $V_1 = C_1$, stop. Otherwise, let $V_2 = V_1 - C_1$. Again, pick any node, v_2 in V_2 and determine the strongly connected component, C_2, of v_2. If $V_2 = C_2$, stop. Otherwise, let $V_3 = V_2 - C_2$, pick v_3 in V_3, and continue in the same manner as before. Ultimately, this process will stop and produce all the strongly connected components $C_1, \ldots, C_k$ of G.

It should be noted that the function *strcomp* and the simple algorithm that we just described are "naive" algorithms that are not particularly efficient. Their main advantage is their simplicity. There are more efficient algorithms, in particular, there is a beautiful algorithm for computing the SCC's due to Robert Tarjan.

Going back to our city traffic problem from Section 5.1, if we compute the strongly connected components for the proposed solution shown in Figure 5.2 , we find three SCC's:

$$\{6, 7, 8, 12, 13, 14\}, \quad \{11\}, \quad \{1, 2, 3, 4, 5, 9, 10, 15, 16, 17, 18, 19\}.$$

Therefore, the city engineers did not do a good job! We will show after proving Proposition 5.3.8 how to "fix" this faulty solution.

Closed simple paths also play an important role.

Definition 5.3.5 Let $G = (V, E, s, t)$ be a digraph. A *circuit* is a closed simple path (i.e., no edge occurs twice) and an *elementary circuit* is an elementary closed path. The null path, (u, ϵ, u), is an elementary circuit.

Remark: A closed path is sometimes called a *pseudo-circuit*. In a pseudo-circuit, some edge may occur move than once.

The significance of elementary circuits is revealed by the next proposition.

Proposition 5.3.6 *Let G be any digraph. (a) Every circuit, π, in G is the concatenation of pairwise edge-disjoint elementary circuits.*

(b) A circuit is elementary iff it is a minimal circuit, that is, iff it does not contain any proper circuit.

Proof. We proceed by induction on the length of π. The proposition is trivially true if π is the null path. Next, let $\pi = (u, e_1 \cdots e_m, u)$ be any non-null circuit and let

$$\text{nodes}(\pi) = \langle v_0, \ldots, v_m \rangle.$$

If π is an elementary circuit, we are done. Otherwise, some node occurs twice in the sequence $\langle v_0, \ldots, v_{m-1} \rangle$ or in the sequence $\langle v_1, \ldots, v_m \rangle$. Let us consider the first case, the second one being similar. Pick two occurrences of the same, node, say $v_i = v_j$, with $i < j$, such that $j - i$ is minimal. Then, due to the minimality of $j - i$, no node occurs twice in $\langle v_i, \ldots, v_{j-1} \rangle$ or $\langle v_{i+1}, \ldots, v_j \rangle$, which shows that $\pi_1 = (v_i, e_{i+1} \cdots e_j, v_i)$ is an elementary circuit. Now, we can write $\pi = \pi'\pi_1\pi''$, with $|\pi'| < |\pi|$ and $|\pi''| < |\pi|$. Thus, we can apply the induction hypothesis to both π' and π'', which shows that π' and π'' are concatenations of elementary circuits. Then, π itself is the concatenation of elementary circuits. All these elementary circuits are pairwise edge-disjoint since π has no repeated edges.

(b) This is clear by definition of an elementary circuit. □

Remarks:

1. If u and v are two nodes that belong to a circuit, π, in G, (i.e., both u and v are incident to some edge in π), then u and v are strongly connected. Indeed, u and v are connected by a portion of the circuit π, and v and u are connected by the complementary portion of the circuit.

2. If π is a pseudo-circuit, the above proof shows that it is still possible to decompose π into elementary circuits, but it may not be possible to write π as the concatenation of pairwise edge-disjoint elementary circuits.

Given a graph, G, we can form a new and simpler graph from G by connecting the strongly connected components of G as shown below.

Definition 5.3.7 Let $G = (V, E, s, t)$ be a digraph. The *reduced graph*, $\widehat{G}$, is the simple digraph whose set of nodes, $\widehat{V} = V/\widehat{C}_G$, is the set of strongly connected components of V and whose set of edges, $\widehat{E}$, is defined as follows:

$$(\widehat{u}, \widehat{v}) \in \widehat{E} \quad \text{iff} \quad (\exists e \in E)(s(e) \in \widehat{u} \quad \text{and} \quad t(e) \in \widehat{v}),$$

where we denote the strongly connected component of u by $\widehat{u}$.

That $\widehat{G}$ is "simpler" than G is the object of the next proposition.

Proposition 5.3.8 *Let G be any digraph. The reduced graph, $\widehat{G}$, contains no circuits.*

Proof. Suppose that u and v are nodes of G and that u and v belong to two disjoint strongly connected components that belong to a circuit, $\widehat{\pi}$, in $\widehat{G}$. Then, the circuit, $\widehat{\pi}$, yields a closed sequence of edges $e_1, \ldots, e_n$ between strongly connected components and we can arrange the numbering so that these components are $C_0, \ldots, C_n$, with $C_n = C_0$, with e_i an edge between $s(e_i) \in C_i$ and $t(e_i) \in C_{i+1}$ for $0 \leq i \leq n-1$, e_n an edge between between $s(e_n) \in C_n$ and $t(e_n) \in C_0$, $\widehat{u} = C_p$ and $\widehat{v} = C_q$, for some $p < q$. Now, we have $t(e_i) \in C_{i+1}$ and $s(e_{i+1}) \in C_{i+1}$ for $0 \leq i \leq n-1$ and $t(e_n) \in C_0$ and $s(e_1) \in C_0$ and as each C_i is strongly connected, we have elementary paths from $t(e_i)$ to $s(e_{i+1})$ and from $t(e_n)$ to $s(e_1)$. Also, as $\widehat{u} = C_p$ and $\widehat{v} = C_q$ for some $p < q$, we have some elementary paths from u to $s(e_p)$ and from $t(e_{q-1})$ to v. By concatenating the appropriate paths, we get a circuit in G containing u and v, showing that u and v are strongly connected, contradicting that u and v belong to two disjoint strongly connected components. □

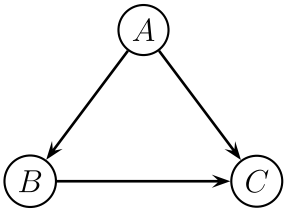

Figure 5.6: The reduced graph of the graph in Figure 5.2

Remark: Digraphs without circuits are called *DAG's*. Such graphs have many nice properties. In particular, it is easy to see that any finite DAG has nodes with no incoming edges. Then, it is easy to see that finite DAG's are basically collections of trees with shared nodes.

The reduced graph of the graph shown in Figure 5.2 is showed in Figure 5.6, where its SCC's are labeled A, B and C as shown below:

$$A = \{6, 7, 8, 12, 13, 14\}, \quad B = \{11\}, \quad C = \{1, 2, 3, 4, 5, 9, 10, 15, 16, 17, 18, 19\}.$$

The locations in the component A are inaccessible. Observe that changing the direction of the street between 13 and 18 yields a solution, that is, a strongly connected graph. So, the engineers were not too far off after all! The solution to our traffic problem is shown in Figure 5.7.

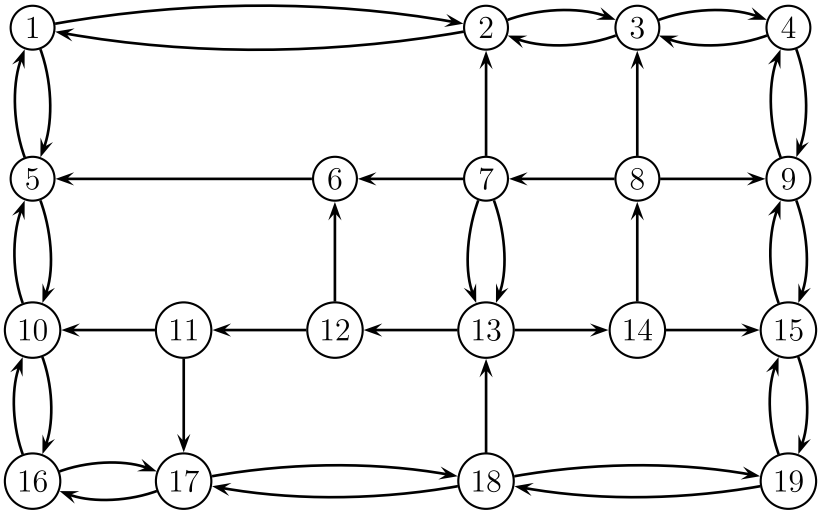

Figure 5.7: A good choice of one-way streets

Before discussing undirected graphs, let us collect various definitions having to do with the notion of subgraph.

Definition 5.3.9 Given any two digraphs, $G = (V, E, s, t)$ and $G' = (V', E', s', t')$, we say that G' *is a subgraph of* G iff $V' \subseteq V$, $E' \subseteq E$, s' is the retriction of s to E' and t' is the retriction of t to E'. If G' is a subgraph of G and $V' = V$, we say that G' is a *spanning subgraph of* G. Given any subset, V', of V, the *induced subgraph,* $G\langle V'\rangle$, *of* G is the graph whose set of edges is

$$E_{V'} = \{e \in E \mid s(e) \in V'; t(e) \in V'\}.$$

(Clearly, s' and t' are the restrictions of s and t to $E_{V'}$, respectively.) Given any subset, $E' \subseteq E$, the graph $G' = (V, E', s', t')$, where s' and t' are the restrictions of s and t to E', respectively, is called the *partial graph of* G *generated by* E'. The graph, $(V', E' \cap V_{V'}, s', t')$, is a *partial subgraph of* G (here, s' and t' are the restrictions of s and t to $E' \cap V_{V'}$, respectively.)

5.4 Undirected Graphs, Chains, Cycles, Connectivity

The edges of a graph express relationships among its nodes. Sometimes, these relationships are not symmetric, in which case it is desirable to use directed arcs, as we have in the previous sections. However, there is a class of problems where these relationships are naturally symmetric or where there is no a priori preferred orientation of the arcs. For example, if V is the population of individuals that were students at Penn between 1900 until now and if we are interested in the relation where two people A and B are related iff they had the same professor in some course, then this relation is clearly symmetric. As a consequence, if we want to find the set of individuals that are related to a given individual, A, it seems unnatural and, in fact, counter-productive, to model this relation using a directed graph.

As another example suppose we want to investigate the vulnerabilty of an internet network under two kinds of attacks: (1) disabling a node; (2) cutting a link. Again, whether of not a link between two sites is oriented is irrelevant. What is important is that the two sites are either connected or disconnected.

These examples suggest that we should consider an “unoriented” version of a graph. How should we proceed?

One way to proceed is to still assume that we have a directed graph but to modify certain notions such as paths and circuits to account for the fact that such graphs are really “unoriented.” In particular, we should redefine paths to allow edges to be traversed in the “wrong direction”. Such an approach is possible but slightly akward and ultimately it is really better to define undirected graphs. However, to show that this approach is feasible, let us give a new definition of a path that corresponds to the notion of path in an undirected graph.

Definition 5.4.1 Given any digraph, $G = (V, E, s, t)$, and any two nodes, $u, v \in V$, a *chain* (or *walk*) *from* u *to* v is a sequence $\pi = (u_0, e_1, u_1, e_2, u_2, \ldots, u_{n-1}, e_n, u_n)$, where $n \geq 1$; $u_i \in V$; $e_i \in E$ and

$$u_0 = u;\ u_n = v \quad \text{and} \quad \{s(e_i), t(e_i)\} = \{u_{i-1}, u_i\}, \quad 1 \leq i \leq n.$$

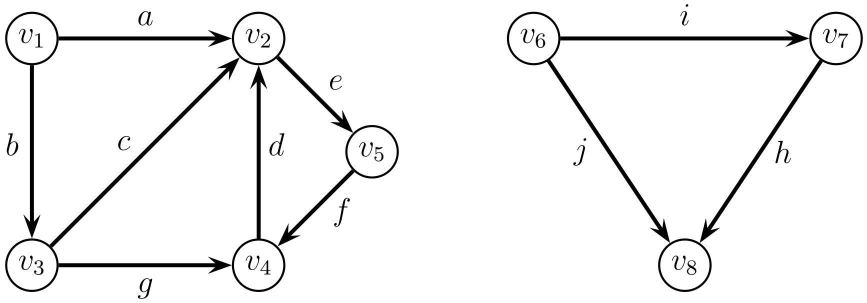

Figure 5.8: Graph G_5

We call n the *length of the chain* π and we write $|\pi| = n$. When $n = 0$, we have the *null chain*, (u, ϵ, u), from u to u, a chain of length 0. If $u = v$, then π is called a *closed chain*, else an *open chain*. The chain, π, determines the sequence of nodes, nodes$(\pi) = \langle u_0, \ldots, u_n \rangle$, with nodes$((u, \epsilon, u)) = \langle u, u \rangle$. A chain, π, is *simple* iff $e_i \neq e_j$ for all $i \neq j$ (i.e., no edge in the chain is used twice). A chain, π, from u to v is *elementary* iff no vertex in nodes(π) occurs twice, except possibly for u if π is closed. The null chain, (u, ϵ, u), is considered simple and elementary.

The main difference between Definition 5.4.1 and Definition 5.3.1 is that Definition 5.4.1 ignores the orientation: in a chain, an edge may be traversed backwards, from its endpoint back to its source. This implies that the reverse of a chain

$$\pi^R = (u_n, e_n, u_{n-1}, , \ldots, u_2, e_2, u_1, e_1, u_0)$$

is a chain from $v = u_n$ to $u = u_0$. In general, this fails for paths. Note, as before, that if G is a simple graph, then a chain is more simply defined by a sequence of nodes

$$(u_0, u_1, \ldots, u_n).$$

For example, in the gaph G_5 shown in Figure 5.8, we have the chains

$$(v_1, a, v_2, d, v_4, f, v_5, e, v_2, d, v_4, g, v_3),\ (v_1, a, v_2, d, v_4, f, v_5, e, v_2, c, v_3),\ (v_1, a, v_2, d, v_4, g, v_3)$$

from v_1 to v_3, The second chain is simple and the third is elementary. Note that none of these chains are paths.

Chains are concatenated the same way as paths and the notion of subchain is analogous to the notion of subpath. The undirected version of Proposition 5.3.2 also holds. The proof is obtained by changing the word "path" to "chain".

Proposition 5.4.2 *Let G be any digraph. (a) For any two nodes, u, v, in G, every non-null chain, π, from u to v contains an elementary non-null subchain.*

(b) If $|V| = n$, then every open elementary chain has length at most $n - 1$ and every closed elementary chain has length at most n.

The undirected version of strong connectivity is the following:

Definition 5.4.3 Let $G = (V, E, s, t)$ be a digraph. We define the binary relation, $\widetilde{C}_G$, on V as follows: For all $u, v \in V$,

$$u\widetilde{C}_G v \quad \text{iff} \quad \text{there is a chain from } u \text{ to } v.$$

When $u\widetilde{C}_G v$, we say that *u and v are connected*.

Oberve that the relation $\widetilde{C}_G$ is an equivalence relation. It is reflexive because we have the null chain from u to u, symmetric because the reverse of a chain is also a chain and transitive because chains can be concatenated. The equivalence classes of the relation $\widetilde{C}_G$ are called the *connected components of G (CC's)*. A graph is *connected* iff it has a single connected component.

Observe that strong connectivity implies connectively but the converse is false. For example, the graph G_1 of Figure 5.3 is connected but it is not strongly connected. The function *strcomp* and the method for computing the strongly connected components of a graph can easily be adapted to compute the connected components of a graph.

The undirected version of a circuit is the following:

Definition 5.4.4 Let $G = (V, E, s, t)$ be a digraph. A *cycle* is a closed simple chain (i.e., no edge occurs twice) and an *elementary cycle* is an elementary closed chain. The null chain, (u, ϵ, u), is an elementary cycle.

Remark: A closed cycle is sometimes called a *pseudo-cycle*. The undirected version of Proposition 5.3.6 also holds. Again, the proof consist in changing the word "circuit" to "cycle".

Proposition 5.4.5 *Let G be any digraph. (a) Every cycle, π, in G is the concatenation of pairwise edge-disjoint elementary cycles.*

(b) A cycle is elementary iff it is a minimal cycle, that is, iff it does not contain any proper cycle.

The reader should now be convinced that it is actually possible to use the notion of a directed graph to model a large class of problems where the notion of orientation is irrelevant. However, this is somewhat unnatural and often inconvenient, so it is desirable to introduce the notion of an undirected graph as a "first-class" object. How should we do that?

We could redefine the set of edges of an undirected graph to be of the form $E^+ \cup E^-$, where $E^+ = E$ is the original set of edges of a digraph and with

$$E^- = \{e^- \mid e^+ \in E^+,\ s(e^-) = t(e^+),\ t(e^-) = s(e^+)\},$$

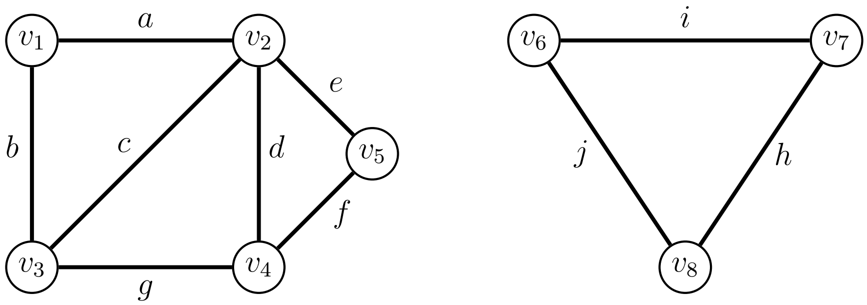

Figure 5.9: The Undirected Graph G_6

each edge, e^-, being the "anti-edge" (opposite edge) of e^+. Such an approach is workable but experience shows that it not very satisfactory.

The solution adopted by most people is to relax the condition that every edge, $e \in E$, is assigned an *ordered pair*, $\langle u, v\rangle$, of nodes (with $u = s(e)$ and $v = t(e)$) to the condition that every edge, $e \in E$, is assigned a *set*, $\{u, v\}$ of nodes (with $u = v$ allowed). To this effect, let $[V]^2$ denote the subset of the power set consisting of all two-element subsets of V (the notation $\binom{V}{2}$ is sometimes used instead of $[V]^2$) :

$$[V]^2 = \{\{u, v\} \in 2^V \mid u \neq v\}.$$

Definition 5.4.6 A *graph* is a triple, $G = (V, E, st)$, where V is a set of *nodes or vertices*, E is a set of *arcs or edges* and $st\colon E \to V \cup [V]^2$ is a function that assigns a set of *endpoints* (or *endnodes*) to every edge.

When we want to stress that we are dealing with an undirected graph as opposed to a digraph, we use the locution *undirected graph*. When we draw an undirected graph we suppress the tip on the extremity of an arc. For example, the undirected graph, G_6, corresponding to the directed graph G_5 is shown in Figure 5.9.

Definition 5.4.7 Given a graph, G, an edge, $e \in E$, such that $st(e) \in V$ is called a *loop* (or *self-loop*). Two edges, $e, e' \in E$ are said to be *parallel edges* iff $st(e) = st(e')$. A graph is *simple* iff it has no loops and no parallel edges.

Remarks:

1. The functions st need not be injective or surjective.

2. When G is simple, every edge, $e \in E$, is uniquely determined by the set of vertices, $\{u, v\}$, such that $\{u, v\} = st(e)$. In this case, we may denote the edge e by $\{u, v\}$ (some books also use the notation (uv) or even uv).

3. Some authors call a graph with no loops but possibly parallel edges a *multigraph* and a graph with loops and parallel edges a *pseudograph*. We prefer to use the term graph for the most general concept.

4. Given an undirected graph, $G = (V, E, st)$, we can form directed graphs from G by assigning an arbitrary orientation to the edges of G. This means that we assign to every set, $st(e) = \{u, v\}$, where $u \neq v$, one of the two pairs (u, v) or (v, u) and define s and t such that $s(e) = u$ and $t(e) = v$ in the first case or such that $s(e) = v$ and $t(e) = u$ in the second case (when $u = v$, we have $s(e) = t(e) = u$)).

5. When a graph is simple, the function st is often omitted and we simply write (V, E), with the understanding that E is a set of two-elements subsets of V.

6. The concepts or adjacency and incidence transfer immediately to (undirected) graphs.

It is clear that the Definition of chain, connectivity, and cycle (Definitions 5.4.1, 5.4.3 and 5.4.4) immediately apply to (undirected) graphs. However, only the notion of *degree* (or *valency*) of a node applies to undirected graph where it is given by

$$d_G(u) = |\{e \in E \mid u \in st(e)\}|.$$

We can check immediately that Corollary 5.2.5 and Corollary 5.2.6 apply to undirected graphs.

Remark: When it is clear that we are dealing with undirected graphs, we will sometimes allow ourselves some abuse of language. For example, we will occasionally use the term path instead of chain.

The notion of homomorphism and isomorphism also makes sense for undirected graphs. In order to adapt Definition 5.2.7, observe that any function, $g \colon V_1 \to V_2$, can be extended in a natural way to a function from $V_1 \cup [V_1]^2$ to $V_2 \cup [V_2]^2$, also denoted g, so that

$$g(\{u, v\}) = \{g(u), g(v)\},$$

for all $\{u, v\} \in [V_1]^2$.

Definition 5.4.8 Given two graphs, $G_1 = (V_1, E_1, st_1)$ and $G_2 = (V_2, E_2, st_2)$, a *homomorphism* (or *morphism*), $f \colon G_1 \to G_2$, *from* G_1 *to* G_2 is a pair, $f = (f^v, f^e)$, with $f^v \colon V_1 \to V_2$ and $f^e \colon E_1 \to E_2$ preserving incidence, that is, for every edge, $e \in E_1$, we have

$$st_2(f^e(e)) = f^v(st_1(e)).$$

These conditions can also be expressed by saying that the following diagram commute:

$$\begin{tikzcd}
E_1 \arrow[r, "f^e"] \arrow[d, "st_1"'] & E_2 \arrow[d, "st_2"] \\
V_1 \cup [V_1]^2 \arrow[r, "f^v"'] & V_2 \cup [V_2]^2.
\end{tikzcd}$$

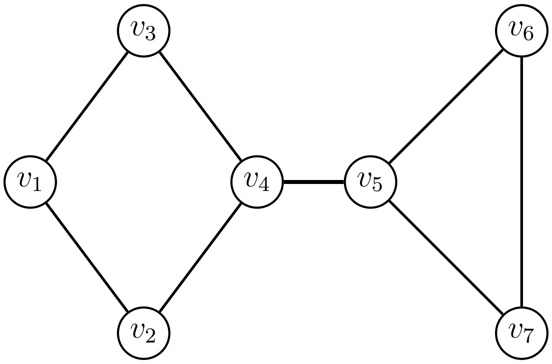

Figure 5.10: A bridge in the graph G_7

As for directed graphs, we can compose homomorphisms of undirected graphs and the definition of an isomorphism of undirected graphs is the same as the definition of an isomorphism of digraphs.

We are now going to investigate the properties of a very important subclass of graphs, trees.

5.5 Trees and Arborescences

In this section, until further notice, we will be dealing with undirected graphs. Given a graph, G, edges having the property that their deletion increases the number of connected components of G play an important role and we would like to characterize such edges.

Definition 5.5.1 Given any graph, $G = (V, E, st)$, any edge, $e \in E$, whose deletion increases the number of connected components of G (i.e., $(V, E - \{e\}, st \upharpoonright (E - \{e\}))$ has more connected components than G) is called a *bridge*.

For example, the edge $(v_4 v_5)$ in the graph shown in Figure 5.10 is a bridge.

Proposition 5.5.2 *Given any graph, $G = (V, E, st)$, adjunction of a new edge, e, between u and v (this means that st is extended to st_e, with $st_e(e) = \{u, v\}$) to G has the following effect:*

1. *Either the number of components of G decreases by 1, in wich case the edge e does not belong to any cycle of $G' = (V, E \cup \{e\}, st_e)$, or*

2. *The number of components of G is unchanged, in wich case the edge e belongs to some cycle of $G' = (V, E \cup \{e\}, st_e)$.*

Proof. Two mutually exclusive cases are possible:

(a) The endpoints u and v (of e) belong to two disjoint connected components of G. In G', these components are merged. The edge e can't belong to a cycle of G' because the chain obtained by deleting e from this cycle would connect u and v in G, a contradiction.

(b) The endpoints u and v (of e) belong to the same connected component of G. Then, G' has the same connected components as G. Since u and v are connected, there is an elementary chain from u to v (by Proposition 5.4.2) and by adding e to this elementary chain, we get a cycle of G' containing e. $\square$

Corollary 5.5.3 *Given any graph, $G = (V, E, st)$, an edge, $e \in E$, is a bridge iff it does not belong to any cycle of G.*

Theorem 5.5.4 *Let G be a finite graph and let $m = |V| \geq 1$. The following properties hold:*

(i) If G is connected, then $|E| \geq m - 1$.

(ii) If G has no cycle, then $|E| \leq m - 1$.

Proof. We can build the graph G progressively by adjoining edges one at a time starting from the graph $(V, \emptyset)$, which has m connected components.

(i) Every time a new edge is added, the number of connected components decreases by at most 1. Therefore, it will take at least $m - 1$ steps to get a connected graph.

(ii) If G has no cycle, then every spannning graph has no cycle. Therefore, at every step, we are in case (1) of Proposition 5.5.2 and the number of connected components decreases by exactly 1. As G has at least one connected component, the number of steps (i.e., of edges) is at most $m - 1$. $\square$

In view of Theorem 5.5.4, it makes sense to define the following kind of graphs:

Definition 5.5.5 A *tree* is a graph that is connected and acyclic (i.e., has no cycles). A *forest* is a graph whose connected components are trees.

The picture of a tree is shown in Figure 5.11.

Our next theorem gives several equivalent characterizations of a tree.

Theorem 5.5.6 *Let G be a finite graph with $m = |V| \geq 2$ nodes. The following properties characterize trees:*

(1) G is connected and acyclic.

(2) G is connected and minimal for this property (if we delete any edge of G, then the resulting graph is no longer connected).

(3) G is connected and has $m - 1$ edges.

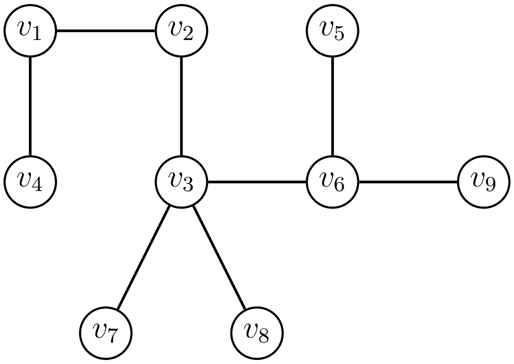

Figure 5.11: A Tree, T_1

(4) G is acyclic and maximal for this property (if we add any edge to G, then the resulting graph is no longer acyclic).

(5) G is acyclic and has $m - 1$ edges.

(6) Any two nodes of G are joined by a unique chain.

Proof. The implications

$$\begin{aligned}(1) &\Longrightarrow (3), (5)\\ (3) &\Longrightarrow (2)\\ (5) &\Longrightarrow (4)\end{aligned}$$

all follow immediately from Theorem 5.5.4.

(4) $\Longrightarrow$ (3). If G was not connected, we could add an edge between to disjoint connected components without creating any cycle in G, contradicting the maximality of G with respect to acyclicity. By Theorem 5.5.4. as G is connected and acyclic, it must have $m - 1$ edges.

(2) $\Longrightarrow$ (6). As G is connected, there is a chain joining any two nodes of G. If, for two nodes u and v, we had two distinct chains from u to v, deleting any edge from one of these two chains would not destroy the connectivity of G contradicting the fact that G is minimal with respect to connectivity.

(6) $\Longrightarrow$ (1). If G had a cycle, then there would be at least two distinct chains joining two nodes in this cycle, a contradiction.

The reader should then draw the directed graph of implications that we just established and check that this graph is strongly connected! Indeed, we have the cycle of implications

$$(1) \Longrightarrow (5) \Longrightarrow (4) \Longrightarrow (3) \Longrightarrow (2) \Longrightarrow (6) \Longrightarrow (1).$$

□

Remark: The equivalence of (1) and (6) holds for infinite graphs too.

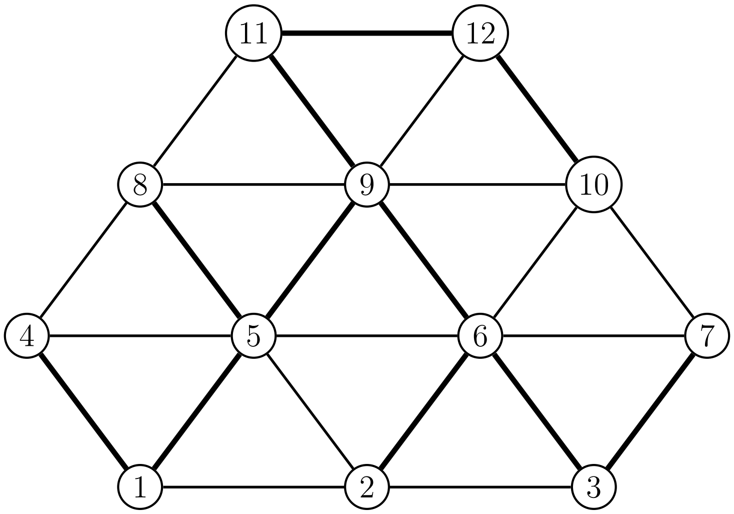

Figure 5.12: A Spanning Tree

Corollary 5.5.7 *For any tree, G, adding a new edge, e, to G yields a graph, G', with a unique cycle.*

Proof. Because G is a tree, all cycles of G' must contain e. If G' had two distinct cycles, there would be two distinct chains in G joining the endpoints of e, contradicting property (6) of Theorem 5.5.6. □

Corollary 5.5.8 *Every finite connected graph possesses a spanning tree.*

Proof. This is a consequence of property (2) of Theorem 5.5.6. Indeed, if there is some edge, $e \in E$, such that deleting e yields a connected graph, G_1, we consider G_1 and repeat this deletion procedure. Eventually, we will get a minimal connected graph that must be a tree. □

An example of a spanning tree (shown in thicker lines) in a graph is shown in Figure 5.12.

An *endpoint* or *leaf* in a graph is a node of degree 1.

Proposition 5.5.9 *Every finite tree with $m \geq 2$ nodes has at least two endpoints.*

Proof. By Theorem 5.5.6, our tree has $m - 1$ edges and by the version of Proposition 5.2.5 for undirected graphs,

$$\sum_{u \in V} d_G(u) = 2(m-1).$$

If we had $d_G(u) \geq 2$ except for a single node u_0, we would have

$$\sum_{u \in V} d_G(u) \geq 2m - 1,$$

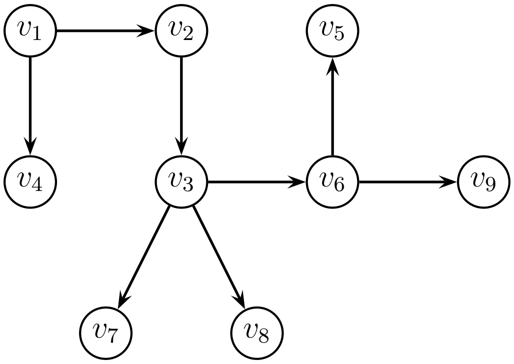

Figure 5.13: An Arborescence, T_2

contradicting the above. □

Remark: A forest with m nodes and p connected components has $m - p$ edges. Indeed, if each connected component has m_i nodes, then the total number of edges is

$$(m_1 - 1) + (m_2 - 1) + \cdots + (m_p - 1) = m - p.$$

We now consider briefly directed versions of a tree.

Definition 5.5.10 Given a digraph, $G = (V, E, s, t)$, a node, $a \in V$ is a *root* (resp. *anti-root*) iff for every node $u \in V$, there is a path from a to u (resp. there is a path from u to a). A digraph with at least two nodes is an *arborescence with root* a iff

1. The node a is a root of G
2. G is a tree (as an undirected graph).

A digraph with at least two nodes is an *anti-arborescence with anti-root* a iff

1. The node a is an anti-root of G
2. G is a tree (as an undirected graph).

Note that orienting the edges in a tree does not necessarily yield an arborescence (or an anti-arborescence). Also, if we reverse the orientation of the arcs of an arborescence we get an anti-arborescence. An arborescence is shown is Figure 5.13.

There is a version of Theorem 5.5.6 giving several equivalent characterizations of an arborescence. The proof of this theorem is left as an exercise to the reader.

Theorem 5.5.11 *Let G be a finite digraph with $m = |V| \geq 2$ nodes. The following properties characterize arborescences with root a:*

(1) G is a tree (as undirected graph) with root a.

(2) For every $u \in V$, there is a unique path from a to u.

(3) G has a as a root and is minimal for this property (if we delete any edge of G, then a is not a root any longer).

(4) G is connected (as undirected graph) and moreover

$$(*)\left\{\begin{array}{l} d_G^-(a) = 0 \\ d_G^-(u) = 1, \textit{ for all } u \in V,\ u \neq a. \end{array}\right.$$

(5) G is acyclic (as undirected graph) and the properties $(*)$ are satisfied.

(6) G is acyclic (as undirected graph) and has a as a root.

(7) G has a as a root and has $m - 1$ arcs.

5.6 Minimum (or Maximum) Weight Spanning Trees

For a certain class of problems, it is necessary to consider undirected graphs (without loops) whose edges are assigned a "cost" or "weight".

Definition 5.6.1 A *weighted graph* is a finite graph without loops, $G = (V, E, st)$, together with a function, $c\colon E \to \mathbb{R}$, called a *weight function* (or *cost function*). We will denote a weighted graph by (G, c). Given any set of edges, $E' \subseteq E$, we define the *weight (or cost)* of E' by

$$c(E') = \sum_{e \in E'} c(e).$$

Given a weighted graph, (G, c), an important problem is to find a spanning tree, T such that $c(T)$ is maximum (or minimum). This problem is called the *maximal weight spanning tree* (resp. *minimal weight spanning tree*). Actually, it is easy to see that any algorithm solving any one of the two problems can be converted to an algorithm solving the other problem. For example, if we can solve the maximal weight spanning tree, we can solve the mimimal weight spanning tree by replacing every weight, $c(e)$, by $-c(e)$, and by looking for a spanning tree, T, that is a maximal spanning tree, since

$$\min_{T \subseteq G} c(T) = -\max_{T \subseteq G} -c(T).$$

There are several algorithms for finding such spanning trees, including one due to Kruskal and another one due to Prim. The fastest known algorithm at the present is due to Bernard Chazelle (1999).

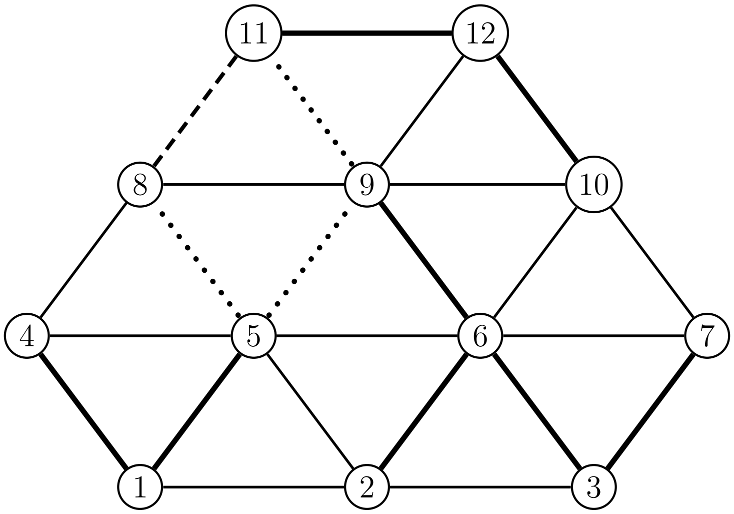

Figure 5.14: The set C_e associated with an edge $e \in G - T$

Since every spannning tree of a given graph, $G = (V, E, st)$, has the same number of edges (namely, $|V| - 1$), adding the same constant to the weight of every edge does not affect the maximal nature a spanning tree, that is, the set of maximal weight spanning trees is preserved. Therefore, we may assume that all the weights are non-negative.

In order to justify the correctness of Kruskal's algorithm, we need two definitions. Let $G = (V, E, st)$ be any connected weighted graph and let T be any spanning tree of G. For every edge, $e \in E - T$, let C_e be the set of edges belonging to the unique chain joining the endpoints of e (the vertices in $st(e)$). For example, in the graph shown in Figure 5.14, the set $C_{\{8,11\}}$ associated with the edge $\{8, 11\}$ (shown as a dashed line) corresponds to the following set of edges (shown as dotted lines) in T:

$$C_{\{8,11\}} = \{\{8, 5\}, \{5, 9\}, \{9, 11\}\}.$$

Also, given any edge, $e \in T$, observe that the result of deleting e yields a graph denoted $T - e$ consisting of two disjoint subtrees of T. We let Ω_e be the set of edges, $e' \in G - T$, such that if $st(e') = \{u, v\}$, then u and v belong to the two distinct connected components of $T - \{e\}$. For example, in Figure 5.15, deleting the edge $\{5, 9\}$ yields the set of edges (shown as dotted lines)

$$\Omega_{\{5,9\}} = \{\{1, 2\}, \{5, 2\}, \{5, 6\}, \{8, 9\}, \{8, 11\}\}.$$

Observe that in the first case, deleting any edge from C_e and adding the edge $e \in E - T$ yields a new spanning tree and in the second case, deleting any edge $e \in T$ and adding any edge in Ω_e also yields a new spanning tree. These observations are crucial ingredients in the proof of the following theorem:

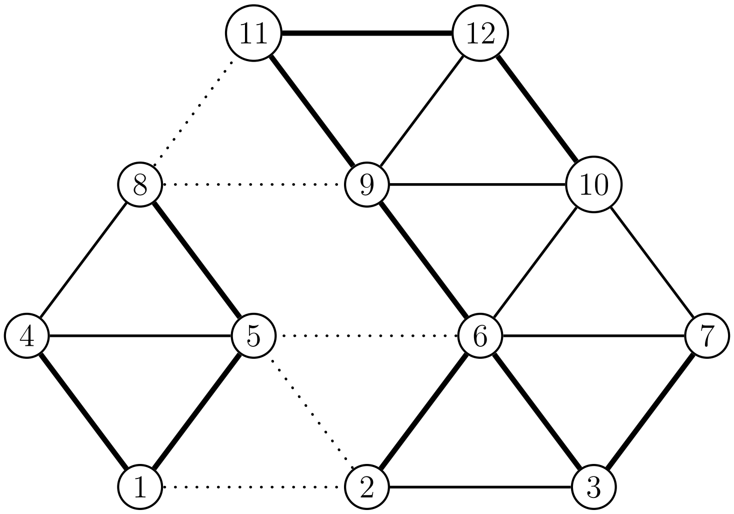

Figure 5.15: The set $\Omega_{\{5,9\}}$ obtained by deleting the edge $\{5,9\}$ from the spanning tree.

Theorem 5.6.2 *Let (G, c) be any connected weighted graph and let T be any spanning tree of G. (1) The tree T is a maximal weight spanning tree iff any of the following (equivalent) conditions hold:*

(i) For every $e \in E - T$,

$$c(e) \leq \min_{e' \in C_e} c(e')$$

(ii) For every $e \in T$,

$$c(e) \geq \max_{e' \in \Omega_e} c(e').$$

(2) The tree T is a minimal weight spanning tree iff any of the following (equivalent) conditions hold:

(i) For every $e \in E - T$,

$$c(e) \geq \max_{e' \in C_e} c(e')$$

(ii) For every $e \in T$,

$$c(e) \leq \min_{e' \in \Omega_e} c(e').$$

Proof. (1) First, assume that T is a maximal weight spanning tree. Observe that

(a) For any $e \in E - T$ and any $e' \in C_e$, the graph $T' = (V, (T \cup \{e\}) - \{e'\})$ is acyclic and has $|V| - 1$ edges, so it is a spanning tree. Then, (i) must hold, as otherwise we would have $c(T') > c(T)$, contradicting the maximality of T.

(b) For any $e \in T$ and any $e' \in \Omega_e$, the graph $T' = (V, (T \cup \{e'\}) - \{e\})$ is connected and has $|V| - 1$ edges, so it is a spanning tree. Then, (ii) must hold, as otherwise we would have $c(T') > c(T)$, contradicting the maximality of T.

Let us now assume that (i) holds. We proceed by contradiction. Let T be a spanning tree satisfying condition (i) and assume there is another spanning tree, T', with $c(T') > c(T)$. Since there are only finitely many spanning trees of G, we may assume that T' is maximal. Consider any edge $e \in T' - T$ and let $st(e) = \{u, v\}$. In T, there is a unique chain, C_e, joining u and v and this chain must contain some edge, $e' \in T$, joining the two connected components of $T' - e$, that is, $e' \in \Omega_e$. As (i) holds, we get $c(e) \leq c(e')$. However, as T' is maximal, (ii) holds (as we just proved), so $c(e) \geq c(e')$. Therefore, we get

$$c(e) = c(e').$$

Consequently, if we form the graph $T_2 = (T' \cup \{e'\}) - \{e\})$, we see that T_2 is a spanning tree having some edge from T and $c(T_2) = c(T')$. We can repeat this process of edge substitution with T_2 and T and so on. Ultimately, we will obtain the tree T with the weight $c(T') > c(T)$, which is absurd. Therefore, T is indeed maximal.

Finally, assume that (ii) holds. The proof is analogous to the previous proof: We pick some edge $e' \in T - T'$ and e is some edge in $\Omega_{e'}$ belonging to the chain joining the endpoints of e' in T'.

(2) The proof of (2) is analogous to the proof of (1) but uses 2(a) and 2(b) instead of 1(a) and 1(b). □

We are now in the position to present a version of Kruskal's algorithm and to prove its correctness. Here is a version of Kruskal's algorithm for finding a minimal weight spanning tree using criterion 2(a). Let n be the number of edges of the weighted graph, (G, c), where $G = (V, E, st)$.

```
function Kruskal((G, c): weighted graph): tree
  begin
    Sort the edges in non-decreasing order of weights:
    $c(e_1) \leq c(e_2) \leq \cdots \leq c(e_n)$;
    $T := \emptyset$;
    for $i := 1$ to $n$ do
      if $(V, T \cup \{e_i\})$ is acyclic then $T := T \cup \{e_i\}$
      endif
    endfor;
    Kruskal := T
  end
```

We admit that the above description of Kruskal's algorithm is a bit sketchy as we have not explicitly specified how we check that adding an edge to a tree preserves acyclicity. On the other hand, it is quite easy to prove the correctness of the above algorithm. It is not difficult to refine the above "naive" algorithm to make it totally explicit but this involves a good choice of data structures. We leave these considerations to an algorithms course.

Clearly, the graph T returned by the algorithm is acyclic, but why is it connected? Well, suppose T is not connected and consider two of its connected components, say T_1 and T_2. Being acyclic and connected, T_1 and T_2 are trees. Now, as G itself it connected, for any node of T_1 and nay node of T_2, there is some chain connecting these nodes. Consider such a chain, C, of minimal length. Then, as T_1 is a tree, the first edge, e_j, of C cannot belong to T_1 since otherwise, we would get an even shorter chain connecting T_1 and T_2 by deleting e_j. Furthermore, e_j does not belong to any other connected component of T, as these connected components are pairwise disjoint. But then, $T + e_j$ is acyclic, which means that when we considered the addition of edge e_j to the current graph, $T^{(j)}$, the test should have been positive and e_j should have been added to $T^{(j)}$. Therefore, T is connected and so, it is a spanning tree. Now, observe that as the edges are sorted in non-decreasing order of weight, condition 2(a) is enforced and by Theorem 5.6.2, T is a minimal weight spanning tree.

We can easily design a version of Kruskal's algorithm based on considion 2(b). This time, we sort the edges in non-increasing order of weights and, starting with G, we attempt to delete each edge, e_j, as long as the remaining graph is still connected. We leave the design of this algorithm as an exercise to the reader.

Prim's algorithm is based on a rather different observation. For any node, $v \in V$, let U_v be the set of edges incident with v that are not loops,

$$U_v = \{e \in E \mid v \in st(e),\ st(e) \in [V]^2\}.$$

Choose in U_v some edge of minimum weight which we will (ambiguously) denote by $e(v)$.

Proposition 5.6.3 *Let (G, c) be a connected weighted graph with $G = (V, E, st)$. For every vertex, $v \in V$, there is a minimum weight spanning tree, T, so that $e(v) \in T$.*

Proof. Let T' be a minimum weight spanning tree of G and assume that $e(v) \notin T'$. Let C be the chain in T' that joins the endpoints of $e(v)$ and let e the edge of C incident with v. Then, the graph $T'' = (V, (T' \cup \{e(v)\}) - \{e\})$ is a spanning tree of weight less that or equal to the weight of T' and as T' has minimum weight, do does T''. By construction, $e(v) \in T''$. □

Prim's algorithm uses an edge-contraction operation described below:

Definition 5.6.4 Let $G = (V, E, st)$ be a graph, and let $e \in E$ be some edge which is not a loop, i.e., $st(e) = \{u, v\}$, with $u \neq v$. The graph, $C_e(G)$, obtained by *contracting the edge e* is the graph obtained by merging u and v into a single node and deleting e. More precisely, $C_e(G) = ((V - \{u, v\}) \cup \{w\}, E - \{e\}, st_e)$, where w is any new node not in V and where

1. $st_e(e') = st(e')$ iff $u \notin st(e')$ and $v \notin st(e')$
2. $st_e(e') = \{w, z\}$ iff $st(e') = \{u, z\}$, with $z \notin st(e)$
3. $st_e(e') = \{z, w\}$ iff $st(e') = \{z, v\}$, with $z \notin st(e)$
4. $st_e(e') = z$ iff $st(e') = \{u, v\}$.

Proposition 5.6.5 *Let $G = (V, E, st)$ be a graph. For any edge, $e \in E$, the graph G is a tree iff $C_e(G)$ is a tree.*

Proof. Proposition 5.6.5 follows from Theorem 5.5.6. Observe that G is connected iff $C_e(G)$ is connected. Moreover, if G is a tree, the number of nodes of $C_e(G)$ is $n_e = |V| - 1$ and the number of edges of $C_e(G)$ is $m_e = |E| - 1$. Since $|E| = |V| - 1$, we get $m_e = n_e - 1$ and $C_e(G)$ is a tree. Conversely, if $C_e(G)$ is a tree, then $m_e = n_e - 1$, $|V| = n_e + 1$ and $|E| = m_e + 1$, so $m = n - 1$ and G is a tree. □

Here is a "naive" version of Prim's algorithm.

```
function Prim((G = (V, E, st), c): weighted graph): tree
  begin
    T := ∅;
    while |V| ≥ 2 do
      pick any vertex v ∈ V;
      pick any edge (not a loop), e, in U_v of minimum weight;
      T := T ∪ {e}; G := C_e(G)
    endwhile;
    Prim := T
  end
```

The correctness of Prim's algorithm is an immediate consequence of Proposition 5.6.3 and Proposition 5.6.5, the details are left to the reader.

5.7 Γ-Cycles, Cocycles, Cotrees, Flows and Tensions

In this section, we take a closer look at the structure of cycles in a finite graph, G. It turns out that there is a dual notion to that of a cycle, the notion of a *cocycle*. Assuming any orientation of our graph, it is possible to associate a vector space, $\mathcal{F}$, to the set of cycles in G, another vector space, $\mathcal{T}$, to the set of cocycles in G, and these vector spaces are mutually orthogonal (for the usual inner product). Furthermore, these vector spaces do not depend on the orientation chosen, up to isomorphism. In fact, if G has m nodes, n edges and p connected components, we will prove that $\dim \mathcal{F} = n - m + p$ and $\dim \mathcal{T} = m - p$. These vector spaces are the *flows* and the *tensions* of the graph G, and these notions are important in combinatorial optimization and the study of networks. This chapter assumes some basic knowledge of linear algebra.

Recall that if G is a directed graph, then a cycle, C, is a closed simple chain, which means that C is a sequence of the form $C = (u_0, e_1, u_1, e_2, u_2, \ldots, u_{n-1}, e_n, u_n)$, where $n \geq 1$; $u_i \in V$; $e_i \in E$ and

$$u_0 = u_n; \quad \{s(e_i), t(e_i)\} = \{u_{i-1}, u_i\}, \quad 1 \leq i \leq n \quad \text{and} \quad e_i \neq e_j \quad \text{for all } i \neq j.$$

The cycle, C, induces the sets C^+ and C^- where C^+ consists of the edges whose orientation agrees with the order of traversal induced by C and where C^- consists of the edges whose orientation is the inverse of the order of traversal induced by C. More precisely,

$$C^+ = \{e_i \in C \mid s(e_i) = u_{i-1},\, t(e_i) = u_i\}$$

and

$$C^- = \{e_i \in C \mid s(e_i) = u_i,\, t(e_i) = u_{i-1}\}.$$

For the rest of this section, we assume that G is a finite graph and that its edges are named, $\mathbf{e}_1, \ldots, \mathbf{e}_n$[1].

Definition 5.7.1 Given any finite directed graph, G, with n edges, to every cycle, C, is associated a *representative vector*, $\gamma(C) \in \mathbb{R}^n$, defined so that for every i, with $1 \leq i \leq n$,

$$\gamma(C)_i = \begin{cases} +1 & \text{if } e_i \in C^+ \\ -1 & \text{if } e_i \in C^- \\ 0 & \text{if } e_i \notin C. \end{cases}$$

For example, if $G = G_8$ is the graph of Figure 5.16, the cycle

$$C = (v_3, e_7, v_4, e_6, v_5, e_5, v_2, e_1, v_1, e_2, v_3)$$

corresponds to the vector

$$\gamma(C) = (-1, 1, 0, 0, -1, -1, 1).$$

Observe that distinct cycles may yield the same representative vector unless they are elementary cycles. For example, the cycles

$$C_1 = (v_2, e_5, v_5, e_6, v_4, e_4, v_2, e_1, v_1, e_2, v_3, e_3, v_2)$$

and

$$C_2 = (v_2, e_1, v_1, e_2, v_3, e_3, v_2, e_5, v_5, e_6, v_4, e_4, v_2)$$

[1] We use boldface notation for the edges in E in order to avoid confusion with the edges occurring in a cycle or in a chain; those are denoted in italic.

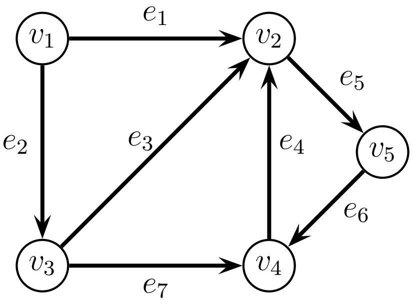

Figure 5.16: Graph G_8

yield the same representative vector

$$\gamma = (-1, 1, 1, 1, 1, 1, 0).$$

In order to obtain a bijection between representative vectors and "cycles", we introduce the notion of a "Γ-cycle" (some authors redefine the notion of cycle and call "cycle" what we call a Γ-cycle, but we find this practice confusing).

Definition 5.7.2 Given a finite directed graph, $G = (V, E, s, t)$, a Γ-*cycle* is any set of edges, $\Gamma = \Gamma^+ \cup \Gamma^-$, such that there is some cycle, C, in G with $\Gamma^+ = C^+$ and $\Gamma^- = C^-$; we say that the cycle, C, *induces the* Γ-*cycle,* Γ. The *representative vector*, $\gamma(\Gamma)$, (for short, γ) associated with Γ is the vector, $\gamma(C)$, from Definition 5.7.1, where C is any cycle inducing Γ. We say that a Γ-cycle, Γ, is a Γ-*circuit* iff either $\Gamma^+ = \emptyset$ or $\Gamma^- = \emptyset$ and that Γ is *elementary* iff Γ arises from an elementary cycle.

Remarks:

1. Given a Γ-cycle, $\Gamma = \Gamma^+ \cup \Gamma^-$, we have the subgraphs $G^+ = (V, \Gamma^+, s, t)$ and $G^- = (V, \Gamma^-, s, t)$. Then, for every $u \in V$, we have
$$d^+_{G^+}(u) - d^-_{G^+}(u) - d^+_{G^-}(u) + d^-_{G^-}(u) = 0.$$

2. If Γ is an elementary Γ-cycle, then every vertex of the graph (V, Γ, s, t) has degree 0 or 2.

3. When the context is clear and no confusion may arise, we often drop the "Γ" is Γ-cycle and simply use the term "cycle'.

Proposition 5.7.3 *If G is any finite directed graph, then any* Γ*-cycle,* Γ*, is the disjoint union of elementary* Γ*-cycles.*

Proof. This is an immediate consequence of Proposition 5.4.5. □

Corollary 5.7.4 *If G is any finite directed graph, then any Γ-cycle, Γ, is elementary iff it is minimal, i.e., if there is no Γ-cycle, Γ', such that $\Gamma' \subseteq \Gamma$ and $\Gamma' \neq \Gamma$.*

We now consider a concept which will turn out to be dual to the notion of Γ-cycle.

Definition 5.7.5 Let G be a finite directed graph, $G = (V, E, s, t)$, with n edges. For any subset of nodes, $Y \subseteq V$, define the sets of edges, $\Omega^+(Y)$ and $\Omega^-(Y)$ by

$$\begin{aligned}
\Omega^+(Y) &= \{e \in E \mid s(e) \in Y,\ t(e) \notin Y\}\\
\Omega^-(Y) &= \{e \in E \mid s(e) \notin Y,\ t(e) \in Y\}\\
\Omega(Y) &= \Omega^+(Y) \cup \Omega^-(Y).
\end{aligned}$$

Any set, Ω, of edges of the form $\Omega = \Omega(Y)$, for some set of nodes, $Y \subseteq V$, is called a *cocycle* (or *cutset*). To every, cocycle, Ω, we associate the *representative vector*, $\omega(\Omega) \in \mathbb{R}^n$, defined so that

$$\omega(\Omega)_i = \begin{cases} +1 & \text{if } e_i \in \Omega^+\\ -1 & \text{if } e_i \in \Omega^-\\ 0 & \text{if } e_i \notin \Omega,\end{cases}$$

with $1 \leq i \leq n$. We also write $\omega(Y)$ for $\omega(\Omega)$ when $\Omega = \Omega(Y)$. If either $\Omega^+(Y) = \emptyset$ or $\Omega^-(Y) = \emptyset$, then Ω is called a *cocircuit* and an *elementary cocycle* (or *bond*) is a minimal cocycle (i.e., there is no cocycle, Ω', such that $\Omega' \subseteq \Omega$ and $\Omega' \neq \Omega$).

In the graph, G_8, of Figure 5.16,

$$\Omega = \{e_5\} \cup \{e_1, e_2, e_6\}$$

is a cocycle induced by the set of nodes, $Y = \{v_2, v_3, v_4\}$ and it corresponds to the vector

$$\omega(\Omega) = (-1, -1, 0, 0, 1, -1, 0).$$

This is not an elementary cocycle because

$$\Omega' = \{e_5\} \cup \{e_6\}$$

is also a cocycle (induced by $Y' = \{v_1, v_2, v_3, v_4\}$). Observe that Ω' is a minimal cocycle, so it is an elementary cocycle. Observe that the inner product

$$\gamma(C_1) \cdot \omega(\Omega) = (-1, 1, 1, 1, 1, 1, 0) \cdot (-1, -1, 0, 0, 1, -1, 0) = 1 - 1 + 0 + 0 + 1 - 1 + 0 = 0$$

is zero. This is a general property that we will prove shortly.

Observe that a cocycle, Ω, is the set of edges of G that join the vertices in a set, Y, to the vertices in its complement, $V - Y$. Consequently, deletetion of all the edges in Ω

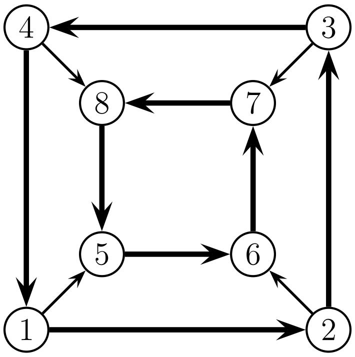

Figure 5.17: A coycle, Ω, equal to the edge set of a cycle, Γ

will increase the number of connected components of G. We say that Ω is a *cutset* of G. Generally, a set of edges, $K \subseteq E$, is a *cutset* of G if the graph $(V, E - K, s, t)$ has more connected components than G.

It should be noted that a cocycle, $\Omega = \Omega(Y)$, may coincide with the set of edges of some cycle, Γ. For example, in the graph displayed in Figure 5.17, the cocycle, $\Omega = \Omega(\{1, 3, 5, 7\})$, shown in thicker lines, is equal to the set of edges of the cycle,

$$(1,2), (2,3), (3,4), (4,1), (5,6), (6,7), (7,8), (8,5).$$

If the edges of the graph are listed in the order

$$(1,2), (2,3), (3,4), (4,1), (5,6), (6,7), (7,8), (8,5), (1,5), (2,6), (3,7), (4,8)$$

the reader should check that the vectors

$$\gamma = (1,1,1,1,1,1,1,1,0,0,0,0) \in \mathcal{F} \quad \text{and} \quad \omega = (1,-1,1,-1,1,-1,1,-1,0,0,0,0,) \in \mathcal{T}$$

correspond to Γ and Ω, respectively.

We now give several characterizations of elementary cocycles.

Proposition 5.7.6 *Given a finite directed graph, $G = (V, E, s, t)$, a set of edges, $S \subseteq E$, is an elementary cocycle iff it is a minimal cutset.*

Proof. We already observed that every cocycle is a cutset. Furthermore, we claim that every cutset contains a cocyle. To prove this, it is enough to consider a minimal cutset, S, and to prove the following satement:

Claim. Any minimal cutset, S, is the set of edges of G that join two nonempty sets of vertices, Y_1 and Y_2, such that

(i) $Y_1 \cap Y_2 = \emptyset$;

(ii) $Y_1 \cup Y_2 = C$, some connected component of G;

(iii) The subgraphs G_{Y_1} and G_{Y_2}, induced by Y_1 and Y_2 are connected.

Indeed, if S is a minimal cutset, it disconnects a unique connected component of G, say C. Let $C_1, \ldots, C_k$ be the connected components of the graph, $C - S$, obtained from C by deleting the edges in S. Adding any edge, $e \in S$, to $C - S$, must connect two components of C since otherwise, $S - \{e\}$ would disconnect C, contradicting the minimality of C. Furthermore, $k = 2$, since otherwise, again, $S - \{e\}$ would disconnect C. Then, if Y_1 is the set of nodes of C_1 and Y_2 is the set of nodes of C_2, it is clear that the Claim holds.

Now, if S is a minimal cutset, the above argument shows that S contains a cocyle and this cocycle must be elementary (i.e., minimal as a cocycle) as it is a cutset. Conversely, if S is an elementary cocycle, i.e., minimal as a cocycle, it must be a minimal cutset since otherwise, S would contain a strictly smaller cutset which would then contain a cocycle strictly contained in S. □

Proposition 5.7.7 *Given a finite directed graph, $G = (V, E, s, t)$, a set of edges, $S \subseteq E$, is an elementary cocycle iff S is the set of edges of G that join two nonempty sets of vertices, Y_1 and Y_2, such that*

(i) $Y_1 \cap Y_2 = \emptyset$;

(ii) $Y_1 \cup Y_2 = C$, some connected component of G;

(iii) The subgraphs G_{Y_1} and G_{Y_2}, induced by Y_1 and Y_2 are connected.

Proof. It is clear that if S satisfies (i)–(iii), then S is a minimal cutset and by Proposition 5.7.7, it is an elementary cocycle.

Let us first assume that G is connected and that $S = \Omega(Y)$ is an elementary cocycle, i.e., is minimal as a cocycle. If we let $Y_1 = Y$ and $Y_2 = X - Y_1$, it is clear that (i) and (ii) are satisfied. If G_{Y_1} or G_{Y_2} is not connected, then if Z is a connected component of one of these two graphs, we see that $\Omega(Z)$ is a cocycle strictly contained in $S = \Omega(Y_1)$, a contradiction. Therefore, (iii) also holds. If G is not connected, as S is a minimal cocycle it is a minimal cutset and so, it is contained in some connected component, C, of G and we apply the above argument to C. □

The following proposition is the analog of Proposition 5.7.3 for cocycle:

Proposition 5.7.8 *Given a finite directed graph, $G = (V, E, s, t)$, every cocycle, $\Omega = \Omega(Y)$, is the disjoint union of elementary cocycles.*

Proof. We give two proofs.

Proof 1: (Claude Berge) Let $Y_1, \ldots, Y_k$ be the connected components of the subgraph of G induced by Y. Then, it is obvious that

$$\Omega(Y) = \Omega(Y_1) \cup \cdots \cup \Omega(Y_k),$$

where the $\Omega(Y_i)$ are pairwise disjoint. So, it is enough to show that each $\Omega(Y_i)$ is the union of disjoint elementary cycles.

Let C be the connected component of G that contains Y_i and let $C_1, \ldots, C_m$ be the connected components of the subgraph, $C - Y$, obtained from C by deleting the nodes in Y_i and the edges incident to these nodes. Observe that the set of edges that are deleted when the nodes in Y_i are deleted is the union of $\Omega(Y_i)$ and the edges of the connected subgraph induced by Y_i. As a consequence, we see that

$$\Omega(Y_i) = \Omega(C_1) \cup \cdots \cup \Omega(C_m),$$

where $\Omega(C_k)$ is the set of edges joining C_k and nodes from Y_i in the connected subgraph induced by the nodes in $Y_i \cup \bigcup_{j \neq k} C_j$. By Proposition 5.7.8, the set $\Omega(C_k)$ is an elementary cocycle and it is clear that the sets $\Omega(C_k)$ are pairwise disjoint since the C_k are disjoint.

Proof 2: (Michel Sakarovitch) Let $\Omega = \Omega(Y)$ be a cocycle in G. Now, Ω is a cutset and we can pick some minimal cocycle, $\Omega_1 = \Omega(Z)$, contained in Ω_1. We proceed by induction on $|\Omega - \Omega_1|$. If $\Omega = \Omega_1$, we are done. Otherwise, we claim that $E_1 = \Omega - \Omega_1$ is a cutset in G. If not, let e be any edge in E_1; we may assume that $a = s(e) \in Y$ and $b = t(e) \in V - Y$. As E_1 is not a cutset, there is a chain, C, from a to b in $(V, E - E_1, s, t)$ and as Ω is a cutset, this chain must contain some edge e', in Ω, so $C = C_1(x, e', y)C_2$, where C_1 is a chain from a to x and C_2 is a chain from y to b. Then, since C has its edges in $E - E_1$ and $E_1 = \Omega - \Omega_1$, we must have $e' \in \Omega_1$. We may assume that $x = s(e') \in Z$ and $y = t(e') \in V - Z$. But, we have the chain, $C_1^R(a, e, b)C_2^R$, joining x and y in $(V, E - \Omega_1)$, a contradiction. Therefore, E_1 is indeed a cutset of G. Now, there is some minimal cocycle, Ω_2, contained in E_1, and if we let $E_2 = E_1 - \Omega_1$, we can show as we just did that E_2 is a cutset of G with $|E_2| < |E_1$. Thus, we finish the proof by applying the induction hypothesis to E_2. □

We now prove the key property of orthogonality between cycles and cocycles.

Proposition 5.7.9 *Given any finite directed graph, $G = (V, E, s, t)$, if $\gamma = \gamma(C)$ is the representative vector of any Γ-cycle, $\Gamma = \Gamma(C)$, and $\omega = \omega(Y)$ is the representative vector of any cocycle, $\Omega = \Omega(Y)$, then*

$$\gamma \cdot \omega = \sum_{i=1}^{n} \gamma_i \omega_i = 0,$$

i.e., γ and ω are orthogonal. (Here, $|E| = n$.)

Proof. Recall that $\Gamma = C^+ \cup C^-$, where C is a cycle in G, say

$$C = (u_0, e_1, u_1, \ldots, u_{k-1}, e_k, u_k), \quad \text{with} \quad u_k = u_0.$$

Then, by definition, we see that

$$\gamma \cdot \omega = |C^+ \cap \Omega^+(Y)| - |C^+ \cap \Omega^-(Y)| - |C^- \cap \Omega^+(Y)| + |C^- \cap \Omega^-(Y)|. \qquad (*)$$

As we traverse the cycle, C, when we traverse the edge e_i between u_{i-1} and u_i ($1 \leq i \leq k$), we note that

$$\begin{aligned} e_i \in (C^+ \cap \Omega^+(Y)) \cup (C^- \cap \Omega^-(Y)) \quad &\text{iff} \quad u_{i-1} \in Y,\, u_i \in V - Y \\ e_i \in (C^+ \cap \Omega^-(Y)) \cup (C^- \cap \Omega^+(Y)) \quad &\text{iff} \quad u_{i-1} \in V - Y,\, u_i \in Y. \end{aligned}$$

In other words, every time we traverse an edge coming out from Y, its contribution to $(*)$ is $+1$ and every time we traverse an edge coming into Y its contribution to $(*)$ is -1. After traversing the cycle C entirely, we must have come out from Y as many times as we came into Y, so these contributions must cancel out. $\square$.

Note that Proposition 5.7.9 implies that $|\Gamma \cap \Omega|$ is even.

Definition 5.7.10 Given any finite digraph, $G = (V, E, s, t)$, where $E = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$, the subspace, $\mathcal{F}(G)$, of $\mathbb{R}^n$ spanned by all vectors, $\gamma(\Gamma)$, where Γ is any Γ-cycle, is called the *cycle space of* G or *flow space of* G and the subspace, $\mathcal{T}(G)$, of $\mathbb{R}^n$ spanned by all vectors, $\omega(\Omega)$, where Ω is any cocycle, is called the *cocycle space of* G or *tension space of* G (or *cut space of* G).

When no confusion is possible, we write $\mathcal{F}$ for $\mathcal{F}(G)$ and $\mathcal{T}$ for $\mathcal{T}(G)$. Thus, $\mathcal{F}$ is the space consisting of all linear combinations $\sum_{i=1}^{k} \alpha_i \gamma_i$ of representative vectors of Γ-cycles, γ_i and $\mathcal{T}$ is the the space consisting of all linear combinations $\sum_{i=1}^{k} \alpha_i \omega_i$ of representative vectors of cocycles, ω_i, with $\alpha_i \in \mathbb{R}$. Proposition 5.7.9 says that the spaces $\mathcal{F}$ and $\mathcal{T}$ are mutually orthogonal.

Remark: The seemingly odd terminology “flow space” and “tension space” will be explained later.

Our next goal will be to determine the dimensions of $\mathcal{F}$ and $\mathcal{T}$ in terms of the number of edges, the number of nodes and the number of connected components of G and to give a convenient method for finding bases of $\mathcal{F}$ and $\mathcal{T}$. For this, we will use spanning trees and their dual, cotrees. But first, we will need a crucial theorem that also plays an important role in the theory of flows in networks.

Theorem 5.7.11 *(Arc Coloring Lemma; Minty [1960]) Let* $G = (V, E, s, t)$*, be a finite directed graph and assume that the edges of* G *are colored either in black, red or green. Pick any edge,* e*, and color it black. Then, exactly one of two possibilities may occur:*

(1) There is an elementary cycle containing e *whose edges are only red or black with all the black edges oriented in the same direction;*

(2) There is an elementary cocycle containing e *whose edges are only green or black with all the black edges oriented in the same direction.*

Proof. Let $a = s(e)$ and $b = t(e)$. Apply the following procedure for making nodes:

```
Intitially, only $b$ is marked.
while there is some marked node $x$ and some unmarked node $y$ with
   either a black edge, $e'$, with $(x, y) = (s(e'), t(e'))$ or
   a red edge, $e'$, with $(x, y) = \{s(e'), t(e')\}$
   then mark $y$; $\mathrm{arc}(y) = e'$
endwhile
```

When the marking algorithm stops, exactly one of the following two cases occurs:

(i) Node a has been marked. Let $e' = \mathrm{arc}(a)$ be the edge that caused a to be marked and let x be the other endpoint of e'. If $x = b$, we found an elementary cycle satisfying (i). If not, let $e'' = \mathrm{arc}(x)$ and let y be the other endpoint of e'' and continue in the same manner. This procedure will stop with b and yields the chain, C, from b to a along which nodes have been marked. This chain must be elementary because every edge in it was used once to mark some node (check that the set of edges used for the marking is a tree). If we add the edge, e, to the chain, C, we obtain an elementary cycle, Γ whose edges are colored black or red and with all edges colored black oriented in the same direction due to the marking scheme. It is impossible to have a cocycle whose edges are colored black or green containing e because it would have been impossible to conduct the marking through this cocycle and a would not have been marked.

(ii) Node a has not been marked. Let Y be the set of unmarked nodes. The set $\Omega(Y)$ is a cocycle whose edges are colored green or black containing e with all black edges in $\Omega^+(Y)$. This cocycle is the disjoint of elementary cocycles (by Proposition 5.7.8) and one of these elementary cocycles contains e. If a cycle with black or red edges containing e with all black edges oriented in the same direction existed, then a would have been marked, a contradiction. □

Corollary 5.7.12 *Every edge of a finite directed graph, G, belongs either to an elementary circuit or to an elementary cocircuit but not both.*

Proof. Color all edges black and apply Theorem 5.7.11. □

Although Minty's Theorem looks more like an amusing fact than a deep result, it is actually a rather powerful theorem. For example, we will see in Section 5.10 that Minty's Theorem can be used to prove the "hard part" of the Max-flow Min-cut Theorem (Theorem 5.10.7), an important theorem that has many applications. Here are a few more applications of Theorem 5.7.11.

Proposition 5.7.13 *Let G be a finite connected directed graph with at lest one edge. Then, the following conditions are equivalent: G is strongly connected iff*

(i) G is strongly connected.

(ii) Every edge belongs to some circuit.

(iii) G has no cocircuit.

Proof. $(i) \Longrightarrow (ii)$. If x and y are the endpoints of any edge, e, in G, as G is strongly connected, there is an elementary path from y to x and thus, an elementary circuit through e.

$(ii) \Longrightarrow (iii)$. This follows from Corollary 5.7.12.

$(iii) \Longrightarrow (i)$. Assume that G is not strongly connected and let Y' and Y'' be two strongly connected components linked by some edge, e, and let $a = s(e)$ and $b = t(e)$, with $a \in Y'$ and $b \in Y''$. The edge e does not belong to any circuit since otherwise, a and b would belong to the same strongly connected component. Thus, by Corollary 5.7.12, the edge e should belong to some cocircuit, a contradiction. □

In order to determine the dimension of the cycle space, $\mathcal{T}$, we will use spanning trees. Let us assume that G is connected since otherwise the same reasoning applies to the connected components of G. If T is any spanning tree of G, we know from Theorem 5.5.6, part (4), that adding any edge, $e \in E - T$, (called a *chord* of T) will create a (unique) cycle. We will see shortly that the vectors associated with these cycles form a basis of the Cycle space. We can find a basis of the cocycle space by considering sets of edges of the form $E - T$, where T is a spanning tree. Such sets of edges are called *cotrees*.

Definition 5.7.14 Let G be a finite directed connected graph, $G = (V, E, s, t)$. A spanning subgraph, (V, K, s, t), is a *cotree* iff $(V, E - K, s, t)$ is a spanning tree.

Cotrees are characterized in the following proposition:

Proposition 5.7.15 *Let G be a finite directed connected graph, $G = (V, E, s, t)$. If E is partitioned into two subsets, T and K, (i.e., $T \cup K = E$; $T \cap K = \emptyset$; $T, K \neq \emptyset$), then the following conditions ar equivalent:*

(1) (V, T, s, t) is tree.

(2) (V, K, s, t) is a cotree.

(3) (V, K, s, t) contains no elementary coycles of G and upon addition of any edge, $e \in T$, it does contain an elementary cocycle of G.

Proof. By definition of a cotree, (1) and (2) are equivalent, so we will prove the equivalence of (1) and (3).

$(1) \Longrightarrow (3)$. We claim that (V, K, s, t) contains no elementary coycles of G. Otherwise, K would contain some elementary coycle, $\Gamma(A)$, of G and then no chain in the tree (V, T, s, t) would connect A and $V - E$, a contradiction.

Next, for any edge, $e \in T$, observe that $(V, T - \{e\}, s, t)$ has two connected components, say A and B and then, $\Omega(A)$ is an elementary cocycle contained in $(V, K \cup \{e\}, s, t)$ (in fact, it is easy to see that it is the only one). Therefore, (3) holds

(3) $\Longrightarrow$ (1). We need to prove that (V, T, s, t) is tree. First, we show that (V, T, s, t) has no cycles. Let $e \in T$ be any edge; color e black; color all edges in $T - \{e\}$ red; color all edges in $K = E - T$ green. By (3), by adding e to K, we find an elementary cocycle of black or green edges that contains e. Thus, there is no cycle of red or black edges containing e. As e is arbitrary, there are no cycles in T.

Finally, we prove that (V, T, s, t) is connected. Pick any edge, $e \in K$, and color it black; color edges in T red; color edges in $K - \{e\}$ green. Since G has no cocycle of black and green edges containing e, there is a cycle of black or red edges containing e. Therefore, $T \cup \{e\}$ has a cycle, which means that there is a path from any two nodes in T. □

We are now ready for the main theorem of this section.

Theorem 5.7.16 *Let G be a finite directed graph, $G = (V, E, s, t)$ and assume that $|E| = n$, $|V| = m$ and that G has p connected components. Then, the cycle space, $\mathcal{F}$, and the cocycle space, $\mathcal{T}$, are subspaces of $\mathbb{R}^n$ of dimensions $\dim \mathcal{F} = n - m + p$ and $\dim \mathcal{T} = m - p$ and $\mathcal{T} = \mathcal{F}^{\perp}$ is the orthogonal complement of $\mathcal{F}$. Furthermore, if $C_1, \ldots, C_p$ are the connected components of G, bases of $\mathcal{F}$ and $\mathcal{T}$ can be found as follows:*

(1) Let $T_1, \ldots, T_p$, be any spanning trees in $C_1, \ldots, C_p$. For each spanning tree, T_i, form all the elementary cycles, $\Gamma_{i,e}$, obtained by adding any chord, $e \in C_i - T_i$, to T_i. Then, the vectors $\gamma_{i,e} = \gamma(\Gamma_{i,e})$ form a basis of $\mathcal{F}$.

(2) For any spanning tree, T_i, as above, let $K_i = C_i - T_i$ be the corresponding cotree. For every edge, $e \in T_i$ (called a twig), there is a unique elementary cocycle, $\Omega_{i,e}$, contained in $K_i \cup \{e\}$. Then, the vectors $\omega_{i,e} = \omega(\Omega_{i,e})$ form a basis of $\mathcal{T}$.

Proof. We know from Proposition 5.7.9 that $\mathcal{F}$ and $\mathcal{T}$ are orthogonal. Thus,

$$\dim \mathcal{F} + \dim \mathcal{T} \leq n.$$

Let us follow the procedure specified in (1). Let $C_i = (E_i, V_i)$, be the i-th connected component of G and let $n_i = |E_i|$ and $|V_i| = m_i$, so that $n_1 + \cdots + n_p = n$ and $m_1 + \cdots + m_p = m$. For any spanning tree, T_i, for C_i, recall that T_i has $m_i - 1$ edges and so, $|E_i - T_i| = n_i - m_i + 1$. If $e_{i,1}, \ldots, e_{i,n_i - m_i + 1}$ are the edges in $E_i - T_i$, then the vectors

$$\gamma_{i,e_{i,1}}, \ldots, \gamma_{i,e_{i,m_i}}$$

must be linearly independent, because $\gamma_{i,e_{i,j}} = \gamma(\Gamma_{i,e_{i,j}})$ and the elementary cycle, $\Gamma_{i,e_{i,j}}$, contains the edge, $e_{i,j}$, that none of the other $\Gamma_{i,e_{i,k}}$ contain for $k \neq j$. So, we get

$$(n_1 - m_1 + 1) + \cdots + (n_p - m_p + 1) = n - m + p \leq \dim \mathcal{F}.$$

Let us now follow the procedure specified in (2). For every spanning tree, T_i, let $e_{i,1}, \ldots, e_{i,m_i-1}$ be the edges in T_i. We know from proposition 5.7.15 that adding any edge, $e_{i,j}$ to $C_i - T_i$ determines a unique elementary cocycle, $\Omega_{i,e_{i,j}}$, containing $e_{i,j}$ and the vectors

$$\omega_{i,e_{i,1}}, \ldots, \omega_{i,e_{i,m_i-1}}$$

must be linearly independent since the elementary cocycle, $\Omega_{i,e_{i,j}}$, contains the edge, $e_{i,j}$, that none of the other $\Omega_{i,e_{i,k}}$ contain for $k \neq j$. So, we get

$$(m_1 - 1) + \cdots + (m_p - 1) = m - p \leq \dim \mathcal{T}.$$

But then, $n \leq \dim \mathcal{F} + \dim \mathcal{T}$, and since we also have $\dim \mathcal{F} + \dim \mathcal{T} \leq n$, we get

$$\dim \mathcal{F} = n - m + p \quad \text{and} \quad \dim \mathcal{T} = m - p.$$

Since the vectors produced in (1) and (2) are linearly independent and in each case, their number is equal to the dimension of the space to which they belong, they are bases of these spaces. □

Since $\dim \mathcal{F} = n - m + p$ and $\dim \mathcal{T} = m - p$ do not depend on the orientation of G, we conclude that the spaces $\mathcal{F}$ and $\mathcal{T}$ are uniquely determined by G, independently of the orientation of G, up to isomorphism. The number $n - m + p$ is called the *cyclomatic number of* G and $m - p$ is called the *cocyclomatic number of* G.

Remarks:

1. Some authors, including Harary [29] and Diestel [14], define the vector spaces $\mathcal{F}$ and $\mathcal{T}$ over the two-element field, $\mathbb{F}_2 = \{0, 1\}$. The same dimensions are obtained for $\mathcal{F}$ and $\mathcal{T}$ and $\mathcal{F}$ and $\mathcal{T}$ still orthogonal. On the other hand, because $1 + 1 = 0$, some interesting phenomena happen. For example, orientation is irrelevant, the sum of two cycles (or cocycles) is their symmetric difference and the space $\mathcal{F} \cap \mathcal{T}$ is **not** necessarily reduced to the trivial space, (0). The space $\mathcal{F} \cap \mathcal{T}$ is called the *bicycle space*. The bicycle space induces a partition of the edges of a graph called the *principal tripartition*. For more on this, Godsil and Royle [24], Sections 14.15 an 14.16 (and Chapter 14).

2. For those who know homology, of course, $p = \dim H_0$, the dimension of the zero-th homology group and $n - m + p = \dim H_1$, the dimension of the first homology group of G viewed as a topological space. Usually, the notation used is $b_0 = \dim H_0$ and $b_1 = \dim H_1$ (the first two *Betti numbers*). Then, the above equation can be rewritten as
$$m - n = b_0 - b_1,$$
which is just the formula for the *Euler-Poincaré characteristic*.

Figure 5.18, shows an unoriented graph (a cube) and a cocycle, Ω, which is also a cycle, Γ, shown in thick lines (i.e., a bicycle, over the field $\mathbb{F}_2$). However, as we saw in the example

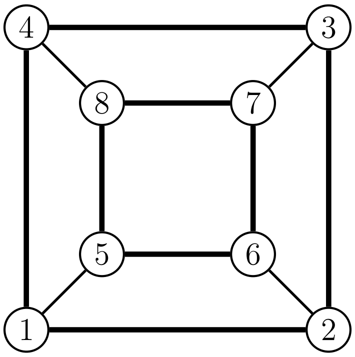

Figure 5.18: A bicycle in a graph (a cube)

from Figure 5.17, for any orientation of the cube, the vectors, γ and ω, corresponding to Γ and Ω are different (and orthogonal).

Let us illustrate the procedures for constructing bases of $\mathcal{F}$ and $\mathcal{T}$ on the graph G_8. Figure 5.19 shows a spanning tree, T and a cotree, K for G_8.

We have $n = 7; m = 5;\ p = 1$, and so, $\dim \mathcal{F} = 7 - 5 + 1 = 3$ and $\dim \mathcal{T} = 5 - 1 = 4$. If we add successively the edges e_2, e_6, and e_7 to the spanning tree, T, we get the three elementary cycles shown in Figure 5.20 with thicker lines.

If we add successively the edges e_1, e_3, e_4 and e_5 to the cotree, K, we get the four elementary cocycles shown in Figures 5.21 and 5.22 with thicker lines.

Given any node, $v \in V$, in a graph, G, for simplicity of notation let us denote the cocycle $\Omega(\{v\})$ by $\Omega(v)$. Similarly, we will write $\Omega^+(v)$ for $\Omega^+(\{v\})$; $\Omega^-(v)$ for $\Omega^-(\{v\})$, and similarly for the the vectors, $\omega(\{v\})$, etc. It turns our that vectors of the form $\omega(v)$ generate the cocycle space and this has important consequences.

Proposition 5.7.17 *Given any finite directed graph, $G = (V, E, s, t)$, for every cocycle, $\Omega = \Omega(Y)$, we have*

$$\omega(Y) = \sum_{v \in Y} \omega(v).$$

Consequently, the vectors of the form $\omega(v)$, with $v \in V$, generate the cocycle space, $\mathcal{T}$.

Proof. For any edge, $e \in E$, if $a = s(e)$ and $b = t(e)$, observe that

$$\omega(v)_e = \begin{cases} +1 & \text{if } v = a \\ -1 & \text{if } v = b \\ 0 & \text{if } v \neq a, b. \end{cases}$$

As a consequently, if we evaluate $\sum_{v \in Y} \omega(v)$, we find that

$$\left(\sum_{v \in Y} \omega(v) \right)_e = \begin{cases} +1 & \text{if } a \in Y \text{ and } b \in V - Y \\ -1 & \text{if } a \in V - Y \text{ and } b \in Y \\ 0 & \text{if } a, b \in Y \text{ or } a, b \in V - Y, \end{cases}$$

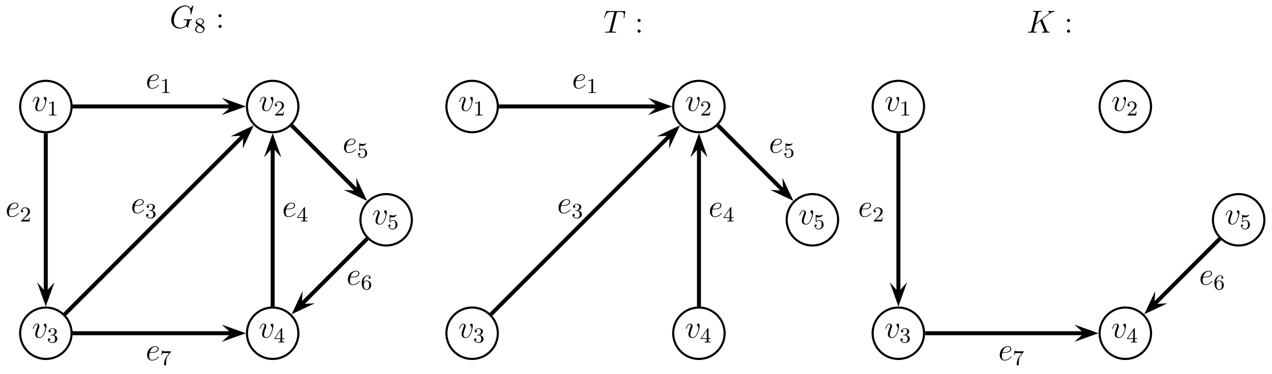

Figure 5.19: Graph G_8; A Spanning Tree, T; A Cotree, K

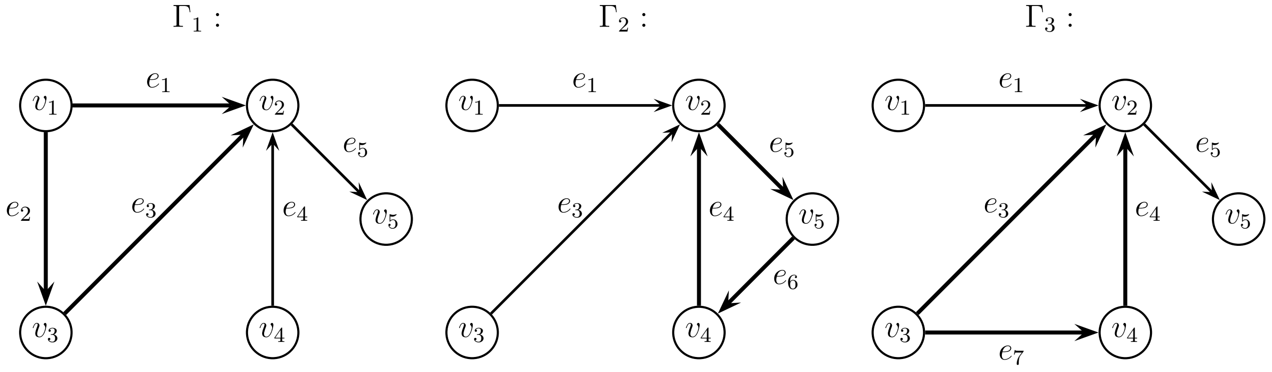

Figure 5.20: A Cycle Basis for G_8

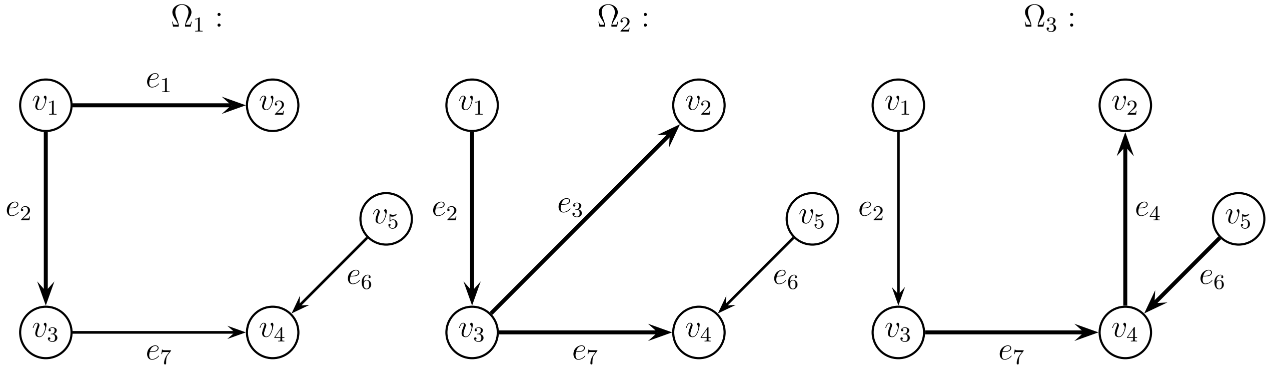

Figure 5.21: A Cocycle Basis for G_8

Ω_4 :

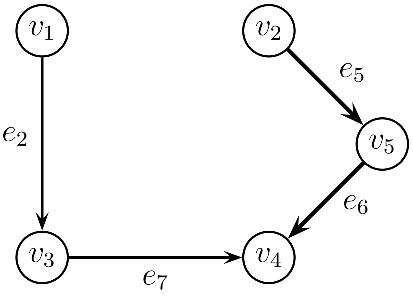

Figure 5.22: A Cocycle Basis for G_8 (continued)

which is exactly $\omega(Y)_v$. □

Proposition 5.7.17 allows us to characterize flows (the vectors in $\mathcal{F}$) in an interesting way which also reveals the reason behind the terminology.

Theorem 5.7.18 *Given any finite directed graph, $G = (V, E, s, t)$, a vector, $f \in \mathbb{R}^n$, is a flow in $\mathcal{F}$ iff*

$$\sum_{e \in \Omega^+(v)} [f(e)] - \sum_{e \in \Omega^-(v)} [f(e)] = 0, \quad \textit{for all} \quad v \in V. \tag{$\dagger$}$$

Proof. By Theorem 5.7.16, we know that $\mathcal{F}$ is the orthogonal complement of $\mathcal{T}$. Thus, for any $f \in \mathbb{R}^n$, we have $f \in \mathcal{F}$ iff $f \cdot \omega = 0$ for all $\omega \in \mathcal{T}$. Moreover, Proposition 5.7.17 says that $\mathcal{T}$ is generated by the vectors of the form $\omega(v)$, where $v \in V$, so $f \in \mathcal{F}$ iff $f \cdot \omega(v) = 0$ for all $v \in V$. But, (†) is exactly the assertion that $f \cdot \omega(v) = 0$ and the theorem is proved. □

Equation (†) justifies the terminology of "flow" for the elements of the space $\mathcal{F}$. Indeed, a *flow*, f, in a (directed) graph, $G = (V, E, s, t)$, is defined as a function, $f \colon E \to \mathbb{R}$, and we say that a flow is *conservative* (Kirchhoff's first law) iff for every node, $v \in V$, the total flow, $\sum_{e \in \Omega^+(v)} [f(e)]$, coming into the vertex, v, is equal to the total flow, $\sum_{e \in \Omega^-(v)} [f(e)]$, coming out of that vertex. This is exactly what equation (†) says.

We can also characterize tensions as follows:

Theorem 5.7.19 *Given any finite simple directed graph, $G = (V, E, s, t)$, for any, $t \in \mathbb{R}^n$, we have:*

(1) The vector, t, is a tension in $\mathcal{T}$ iff for every elementary cycle, $\Gamma = \Gamma^+ \cup \Gamma^-$, we have

$$\sum_{e \in \Gamma^+} [t(e)] - \sum_{e \in \Gamma^-} [t(e)] = 0. \tag{$*$}$$

(2) If G has no parallel edges (and no loops), then $t \in \mathbb{R}^n$ is a tension in $\mathcal{T}$ iff the following condition holds: There is a function, $\pi \colon V \to \mathbb{R}$, called a "potential function", such that

$$t(e) = \pi(t(e)) - \pi(s(e)), \tag{$**$}$$

for every $e \in E$.

Proof. (1) The equation, $(*)$, asserts that $\gamma(\Gamma) \cdot t = 0$ for every elementary cycle, Γ. Since every cycle is the disjoint union of elementary cycles, the vectors of the form $\gamma(\Gamma)$ generate the flow space, $\mathcal{F}$, and by Theorem 5.7.16, the tension space $\mathcal{T}$ is the orthogonal complement of $\mathcal{F}$, so t is a tension iff $(*)$ holds.

(2) Assume a potential function, $\pi \colon V \to \mathbb{R}$, exists, let $\Gamma = (v_0, e_1, v_1, \ldots, v_{k-1}, e_k, v_k)$, with $v_k = v_0$, be an elementary cycle and and let $\gamma = \gamma(\Gamma)$. We have

$$\begin{array}{rcl}
\gamma_1 t(e_1) & = & \pi(v_1) - \pi(v_0) \\
\gamma_2 t(e_2) & = & \pi(v_2) - \pi(v_1) \\
 & \vdots & \\
\gamma_{k-1} t(e_{k-1}) & = & \pi(v_{k-1}) - \pi(v_{k-2}) \\
\gamma_k t(e_k) & = & \pi(v_0) - \pi(v_{k-1})
\end{array}$$

and we see that when we add up both sides of these equations that we get $(*)$:

$$\sum_{e \in \Gamma^+} [t(e)] - \sum_{e \in \Gamma^-} [t(e)] = 0.$$

Let us now assume that $(*)$ holds for every elementary cycle and let $t \in \mathcal{T}$ be any tension. Consider the following procedure for assigning a value, $\pi(v)$, to every vertex, $v \in V$, so that $(**)$ is satisfied. Pick any vertex, v_0, and assign it the value, $\pi(v_0) = 0$.

Now, for every vertex, $v \in V$, that has not yet been assigned a value, do the following:

1. If there is an edge, $e = (u, v)$, with $\pi(u)$ already determined, set

$$t(v) = t(u) + t(e);$$

2. If there is an edge, $e = (v, u)$, with $\pi(u)$ already determined, set

$$t(v) = t(u) - t(e).$$

At the end of this process, all the nodes in the connected component of v_0 will have received a value and we repeat this process for all the other connected components. However, we have to check that each nodes receives a unique value (given the choice of v_0). If some node,

v, is assigned two different values, $\pi_1(v)$ and $\pi_2(v)$, then there exist two chains, σ_1 and σ_2, from v_0 to v, and if C is the cycle $\sigma_1\sigma_2^R$, we have

$$\gamma(C)\cdot t \neq 0.$$

However, any cycle is the disjoint union of elementary cycles, so there would be some elementary cycle, Γ, with

$$\gamma(\Gamma)\cdot t \neq 0,$$

contradicting $(*)$. Therefore, the function π is indeed well-defined and, by construction, satisfies $(**)$. □

Some of these results can be improved in various ways. For example, flows have what is called a "conformal decomposition".

Definition 5.7.20 Given any finite directed graph, $G = (V, S, s, t)$, we say that a flow, $f \in \mathcal{F}$, has a *conformal decomposition*, iff there are some cycles, $\Gamma_1, \ldots, \Gamma_k$, such that if $\gamma_i = \gamma(\Gamma_i)$, then

$$f = \alpha_1\gamma_1 + \cdots + \alpha_k\gamma_k,$$

with

1. $\alpha_i \geq 0$, for $i = 1, \ldots, k$;
2. For any edge, $e \in E$, if $f(e) > 0$ (resp. $f(e) < 0$) and $e \in \Gamma_j$, then $e \in \Gamma_j^+$ (resp. $e \in \Gamma_j^-$).

Proposition 5.7.21 *Given any finite directed graph, $G = (V, S, s, t)$, every flow, $f \in \mathcal{F}$, has some conformal decomposition. In particular, if $f(e) \geq 0$ for all $e \in E$, then all the Γ_j's are circuits.*

Proof. We proceed by induction on the number on nonzero components of f. First, note that $f = 0$ has a trivial conformal decomposition. Next, let $f \in \mathcal{F}$ be a flow and assume that every flow, f', having at least one more zero component than f has some conformal decomposition. Let $\overline{G}$ be the graph obtained by reversing the orientation of all edges, e, for which $f(e) < 0$ and deleting all the edges for wich $f(e) = 0$. Observe that $\overline{G}$ has no cocircuit, as the inner product of any elementary cocircuit with any nonzero flow cannot be zero. Hence, by the corollary to the Coloring Lemma, $\overline{G}$ has some circuit, C, and let Γ be a cycle of G corresponding to C. Let

$$\alpha = \min\{\min_{e\in\Gamma^+} f(e), \min_{e\in\Gamma^-} -f(e)\} \geq 0.$$

Then, the flow

$$f' = f - \alpha\gamma(\Gamma)$$

has at least one more zero component than f. Thus, f' has some conformal decomposition and, by construction, $f = f' + \alpha\gamma(\Gamma)$ is a conformal decomposition of f. □

We now take a quick look at various matrices associated with a graph.

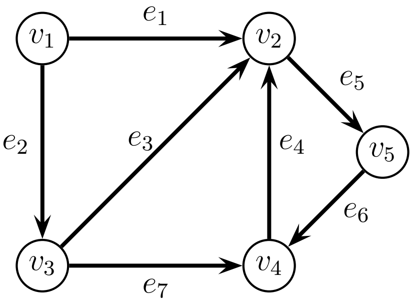

Figure 5.23: Graph G_8

5.8 Incidence and Adjacency Matrices of a Graph

In this section, we are assuming that our graphs are finite, directed, without loops and without parallel edges.

Definition 5.8.1 Let $G = (V, E)$ be a graph with $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$ and $E = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$. The *incidence matrix, $D(G)$, of G*, is the $m \times n$-matrix whose entries, $d_{i\,j}$, are

$$d_{i\,j} = \begin{cases} +1 & \text{if } \mathbf{v}_i = s(\mathbf{e}_j) \\ -1 & \text{if } \mathbf{v}_i = t(\mathbf{e}_j) \\ 0 & \text{otherwise.} \end{cases}$$

Remark: The incidence matrix actually makes sense for a graph, G, with parallel edges but without loops.

For simplicity of notation and when no confusion is possible, we write D instead of $D(G)$.

Since we assumed that G has no loops, observe that every column of D contains exactly two nonzero entries, $+1$ and -1. Also, the ith row of D is the vector, $\omega(\mathbf{v}_i)$, representing the cocycle, $\Omega(\mathbf{v}_i)$. For example, here is the incidence matrix of the graph G_8 shown again in Figure 5.23.

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

The incidence matrix, D, of a graph, G, represents a linear map from $\mathbb{R}^n$ to $\mathbb{R}^m$ called the *incidence map* (or *boundary map* and denoted by D (or ∂). For every $e \in E$, we have

$$D(\mathbf{e_j}) = s(\mathbf{e_j}) - t(\mathbf{e_j}).$$

Remark: Sometimes, it is convenient to consider the vector space, $C_1(G) = \mathbb{R}^E$, of all functions, $f\colon E \to \mathbb{R}$, called the *edge space of* G and the vector space, $C_0(G) = \mathbb{R}^V$, of all functions, $g\colon V \to \mathbb{R}$, called the *vertex space of* G. Obviously, $C_1(G)$ is isomorphic to $\mathbb{R}^n$ and $C_0(G)$ is isomorphic to $\mathbb{R}^m$. The transpose, $D^\top$, of D, is a linear map from $C_0(G)$ to $C_1(G)$ also called the *coboundary map* and often denoted by δ. Observe that $\delta(Y) = \Omega(Y)$ (viewing the subset, $Y \subseteq V$, as a vector in $C_0(G)$).

The spaces of flows and tensions can be recovered from the incidence matrix.

Theorem 5.8.2 *Given any finite graph, G, if D is the incidence matrix of G and $\mathcal{F}$ and $\mathcal{T}$ are the spaces of flows and tensions on G, then*

(1) $\mathcal{F} = \mathrm{Ker}\ D$;

(2) $\mathcal{T} = \mathrm{Im}\ D^\top$.

Futhermore, if G has p connected components and m nodes, then

$$\mathrm{rank}\, D = m - p.$$

Proof. We already observed that the ith row of D is the vector $\omega(\mathbf{v}_i)$ and we know from Theorem 5.7.18 that $\mathcal{F}$ is exactly the set of vectors orthogonal to all vectors of the form $\omega(\mathbf{v}_i)$. Now, for any $f \in \mathbb{R}^n$,

$$\mathbf{A}f = \begin{pmatrix} \omega(\mathbf{v}_1)\cdot f \\ \vdots \\ \omega(\mathbf{v}_m)\cdot f, \end{pmatrix}$$

and so, $\mathcal{F} = \mathrm{Ker}\ D$. Since the vectors $\omega(\mathbf{v}_i)$ generate $\mathcal{T}$, the rows of D generate $\mathcal{T}$, i.e., $\mathcal{T} = \mathrm{Im}\ D^\top$.

From Theorem 5.7.16, we know that

$$\dim \mathcal{T} = m - p$$

and since we just proved that $\mathcal{T} = \mathrm{Im}\ D^\top$, we get

$$\mathrm{rank}\, D = \mathrm{rank}\, D^\top = m - p,$$

which proves the last part of our theorem. □

Corollary 5.8.3 *For any graph, $G = (V, E, s, t)$, if $|V| = m$, $|E| = n$ and G has p connected components, then the incidence matrix, D, of G has rank n (i.e., the columns of D are linearly independent) iff $\mathcal{F} = (0)$ iff $n = m - p$.*

Proof. By Theorem 5.8.3, we have $\mathrm{rank}\, D = m - p$. So, $\mathrm{rank}\, D = n$ iff $n = m - p$ iff $n - m + p = 0$ iff $\mathcal{F} = (0)$ (since $\dim \mathcal{F} = n - m + p$). □

The incidence matrix of a graph has another interesting property observed by Poincaré. First, let us define a variant of triangular matrices.

Definition 5.8.4 An $n \times n$ (real or complex) matrix, $A = (a_{i\,j})$, is said to be *pseudo-triangular and non-singular* iff either

(i) $n = 1$ and $a_{1\,1} \neq 0$, or

(ii) $n \geq 2$ and A has some row, say k, with a unique nonzero entry, $a_{h,\,k}$, such that the submatrix, B, obtained by deleting the h-th row and the k-th column from A is also pseudo-triangular and non-singular.

It is easy to see a matrix defined as in Definition 5.8.4 can be transformed into a usual triangular matrix by permutation of its columns.

Proposition 5.8.5 *(Poincaré, 1901) If D is the incidence matrix of a graph, then every square $k \times k$ nonsingular submatrix*[2]*, B, of D is pseudo-triangular. Consequently,* $\det(B) = +1, -1$*, or* 0*, for any square $k \times k$ submatrix, B, of D.*

Proof. We proceed by induction on k. The result is obvious for $k = 1$.

Next, let B be a square $k \times k$-submatrix of D which is nonsingular, not pseudo-triangular and yet, every nonsingular $h \times h$-submatrix of B is pseudo-triangular if $h < k$. We know that every column of B has at most two nonzero entries (since every column of D contains two nonzero entries: $+1$ and -1). Also, as B is not pseudo-triangular (but nonsingular) every row of B contains at least two nonzero elements. But then, no row of B may contain three of more elements, because the number of nonzero slots in all columns is at most $2k$ and by the pigeonhole principle, we could fit $2k + 1$ objects in $2k$ slots, which is impossible. Therefore, every row of B contains exactly two nonzero entries. Again, the pigeonhole principle implies that every column also contains exactly two nonzero entries. But now, the nonzero entries in each column are $+1$ and -1, so if we add all the rows of B, we get the zero vector, which shows that B is singular, a contradiction. Therefore, B is pseudo-triangular.

Since the entries in D are $+1, -1, 0$, the above immediately implies that
$\det(B) = +1, -1$, or 0, for any square $k \times k$ submatrix, B, of D. □

A square matrix such, A, such that $\det(B) = +1, -1$, or 0, for any square $k \times k$ submatrix, B, of A is said to be *totally unimodular*. This is a very strong property of incidence matrices that has far reaching implications in the study of optimization problems for networks.

Another important matrix associated to a graph is its adjacency matrix.

Definition 5.8.6 Let $G = (V, E)$ be a graph with $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}$. The *ajacency matrix, $A(G)$, of G*, is the $m \times m$-matrix whose entries, $a_{i\,j}$, are

$$a_{i\,j} = \begin{cases} 1 & \text{if } (\exists e \in E)(\{s(e), t(e)\} = \{\mathbf{v}_i, \mathbf{v}_j\}) \\ 0 & \text{otherwise.} \end{cases}$$

[2] Given any $m \times n$ matrix, $A = (a_{i\,j})$, if $1 \leq h \leq m$ and $1 \leq k \leq n$, then a $h \times k$-submatrix, B, of A is obtained by picking any k columns of A and then any h rows of this new matrix.

When no confusion is possible, we write A for $A(G)$. Note that the matrix A is symmetric and $a_{i\,i} = 0$. Here is the adjacency matrix of the graph G_8 shown in Figure 5.23:

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

We have the following useful relationship between the incidence matrix and the adjacency matrix of a graph:

Proposition 5.8.7 *Given any graph, G, if D is the incidence matrix of G, A is the adjacency matrix of G and Δ is the diagonal matrix such that $\Delta_{i\,i} = d(\mathbf{v}_i)$, the degree of node $\mathbf{v}_i$, then*

$$DD^\top = \Delta - A.$$

Consequently, $DD^\top$ is independent of the orientation of G and $\Delta - A$ is symmetric positive, semi-definite, i.e., the eigenvalues of $\Delta - A$ are real and non-negative.

Proof. It is well-known that $DD^\top_{i\,j}$ is the inner product of the ith row, d_i and the jth row, d_j, of D. If $i = j$, then as

$$d_{i\,k} = \begin{cases} +1 & \text{if } s(\mathbf{e}_k) = \mathbf{v}_i \\ -1 & \text{if } t(\mathbf{e}_k) = \mathbf{v}_i \\ 0 & \text{otherwise,} \end{cases}$$

we see that $d_i \cdot d_i = d(\mathbf{v}_i)$. If $i \neq j$, then $d_i \cdot d_j \neq 0$ iff there is some edge, $\mathbf{e}_k$ with $s(\mathbf{e}_k) = \mathbf{v}_i$ and $t(\mathbf{e}_k) = \mathbf{v}_i$, in which case, $d_i \cdot d_j = -1$. Therefore,

$$DD^\top = \Delta - A,$$

as claimed. Now, $DD^\top$ is obviously symmetric and it is well kown that its eigenvalues are non-negative (for example, see Gallier [20], Chapter 12). □

Remarks:

1. The matrix, $L = DD^\top = \Delta - A$, is known as the *Laplacian (matrix)* of the graph, G. Another common notation for the matrix $DD^\top$ is Q. Since the colums of D contain exactly the two nonzero entries, $+1$ and -1, we see that the vector, $\mathbf{1}$, defined such that $\mathbf{1}_i = 1$, is an eigenvector for the eigenvalue 0.

2. If G is connected, then D has rank $m - 1$, so the rank of $DD^\top$ is also $m - 1$ and the other eigenvalues of $DD^\top$ besides 0 are strictly positive. The smallest positive eigenvalue of $L = DD^\top$ has some remarkable properties. There is an area of graph theory overlapping (linear) algebra, called *spectral graph theory* that investigates the

properties of graphs in terms of the eigenvalues of its Laplacian matrix but this is beyond the scope of these notes. Some good references for algebraic graph theory include Biggs [5], Godsil and Royle [24] and Chung [9], for spectral graph theory.

One of the classical and surprising results in algebraic graph theory is a formula that gives the number of spanning trees, $\tau(G)$, of a connected graph, G, in terms of its Laplacian, $L = DD^{\top}$. If J denotes the square matrix whose entries are all 1's and if $\operatorname{adj} L$ denotes the adjoint matrix of L (the transpose of the matrix of cofactors of L), i.e. the matrix given by

$$(\operatorname{adj} L)_{i\,j} = (-1)^{i+j} \det L(j, i),$$

where $L(j, i)$ is the matrix obtained by deleting the jth row and the i-column of L, then we have

$$\operatorname{adj} L = \tau(G)J.$$

We also have

$$\tau(G) = m^{-2} \det(J + L),$$

where m is the number of nodes of G.

3. As we already observed, the incidence matrix also makes sense for graphs with parallel edges and no loops. But now, in order for the equation $DD^{\top} = \Delta - A$ to hold, we need to define A differently. We still have the same definition as before for the adjacency matrix but we can define the new matrix, $\mathcal{A}$, such that

$$\mathcal{A}_{i\,j} = |\{e \in E \mid s(e) = \mathbf{v}_i,\, t(e) = \mathbf{v}_j\}|,$$

i.e., $\mathcal{A}_{i\,j}$ is the number of parallel edges between $\mathbf{v}_i$ and $\mathbf{v}_j$. Then, we can check that

$$DD^{\top} = \Delta - \mathcal{A}.$$

4. There are also versions of the adjacency matrix and of the incidence matrix for undirected graphs. In this case, D is no longer totally unimodular.

5.9 Eulerian and Hamiltonian Cycles

In this short section, we discuss two classical problems that go back to the very beginning of graph theory. These problems have to do with the existence of certain kinds of cycles in graphs. These problems come in two flavors depending whether the graphs are directed or not but there are only minor differences between the two versions and traditionally the focus is on undirected graphs.

The first problems goes back to Euler and is usually known as the *Königsberg bridge problem*. In 1736, the town of Königsberg had seven bridges joining four areas of land.

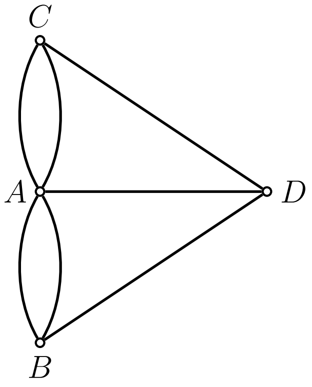

Figure 5.24: A graph modeling the Königsberg bridge problem

Euler was asked whether it is possible to find a cycle that crosses every bridge exactly once (and returns to the starting point).

The graph shown in Figure 5.24 models the Königsberg bridge problem. The nodes A, B, C, D correspond to four areas of land in Königsberg and the edges to the seven bridges joining these areas of land.

In fact, the problem is unsolvable, as shown by Euler, because some nodes do not have an even degree. We will now define the problem precisely and give a complete solution.

Definition 5.9.1 Given a finite undirected graph, $G = (V, E)$, (resp., a directed graph, $G = (V, E, s, t)$), an *Euler cycle* (or *Euler tour*) (resp. an *Euler circuit*) is a cycle in G that passes through every node and every edge (exactly once) (resp. a circuit in G that passes through every node and every edge (exactly once)). The *Eulerian cycle (resp. circuit) Problem* is the problem: Given a graph G, is there an Eulerian cycle (resp. circuit) in G?

Theorem 5.9.2 *(1) An undirected graph, $G = (V, E)$, has an Eulerian cycle iff the following properties hold:*

(a1) The graph G is connected.

(b1) Every node has even degree.

(2) A directed graph, $G = (V, E, s, t)$, has an Eulerian circuit iff the following properties hold:

(a2) The graph G is strongly connected.

(b2) Every node has the same number of incoming and outgoing edges, i.e., $d^+(v) = d^-(v)$, for all $v \in V$.

Proof. We prove (1) leaving (2) as an easy exercise (the proof of (2) is very similar to the proof of (1)). Clearly, if a Euler cycle exists, G is connected and since every edge is traversed exactly once, every node is entered as many times as it is exited so the degree of every node is even.

For the converse, observe that G must contain a cycle as otherwise, being connected, G would be a tree but we proved earlier that every tree has some node of degree 1. (If G is directed and strongly connected, then we know that every edge belongs to a circuit.) Let Γ be any cycle in G. We proceed by induction on the number of edges in G. If G has a single edge, clearly $\Gamma = G$ and we are done. If G has no loops and G has two edges, again $\Gamma = G$ and we are done. If G has no loops and no parallel edges and if G has three edges, then again, $\Gamma = G$. Now, consider the induction step. Assume $\Gamma \neq G$ and consider the graph $G' = (V, E - \Gamma)$. Let $G_1, \ldots, G_p$ be the connected components of G'. Pick any connected component, G_i, of G'. Now, all nodes in G_i have even degree, G_i is connected and G_i has strictly fewer edges than G so, by the induction hypothesis, G_i contains an Euler cycle, Γ_i. But then, Γ and each Γ_i share some vertex and we can combine Γ and the Γ_i's to form an Euler cycle in G. $\square$

There are iterative algorithms that will find an Euler cycle if one exists. It should also be noted that testing whether or not a graph has an Euler cycle is computationally quite an easy problem. This is not so for the Hamiltonian cycle problem described next.

A game invented by Sir William Hamilton in 1859 uses a regular solid dodecahedron whose twenty vertices are labeled with the names of famous cities. The player is challenged to "travel around the world" by finding a circuit along the edges of the dodecahedron which passes through every city exactly once.

In graphical terms, assuming an orientation of the edges between cities, the graph D shown in Figure 5.25 is a plane projection of a regular dodecahedron and we want to know if there is a Hamiltonian cycle in this directed graph (this is a directed version of the problem).

Finding a Hamiltonian cycle in this graph does not appear to be so easy! A solution is shown in Figure 5.26 below:

Definition 5.9.3 Given any undirected graph, G, (resp. directed graph, G) a *Hamiltonian cycle* in G (resp. *Hamiltonian circuit* in G) is a cycle that passes though every vertex of G exactly once (resp. circuit that passes though every vertex of G exactly once). The *Hamiltonian cycle (resp. circuit) problem* is to decide whether a graph, G has a Hamiltonian cycle (resp. Hamiltonian circuit).

Unfortunately, no theorem analogous to Theorem 5.9.2 is known for Hamiltonian cycles. In fact, the Hamiltonian cycle problem is known to be NP-complete and so far, appears to be a computationally hard problem (of exponential time complexity). Here is a proposition that may be used to prove that certain graphs are not Hamiltonian. However, there are graphs satisfying the condition of that proposition that are not Hamiltonian!

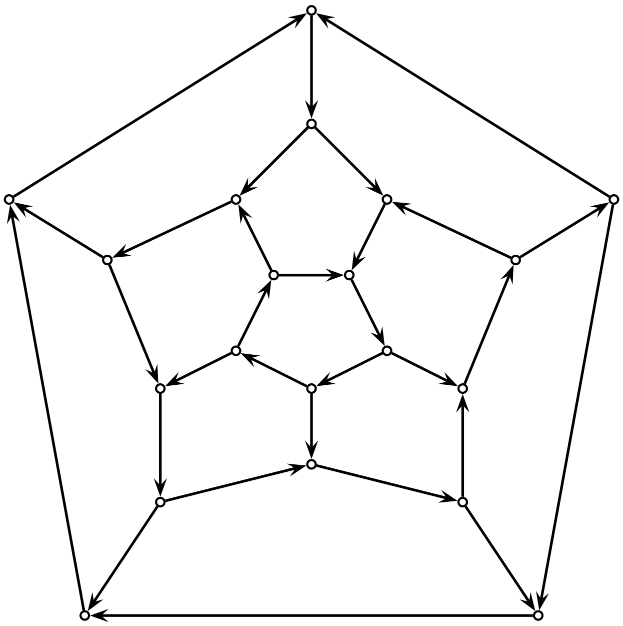

Figure 5.25: A tour "around the world."

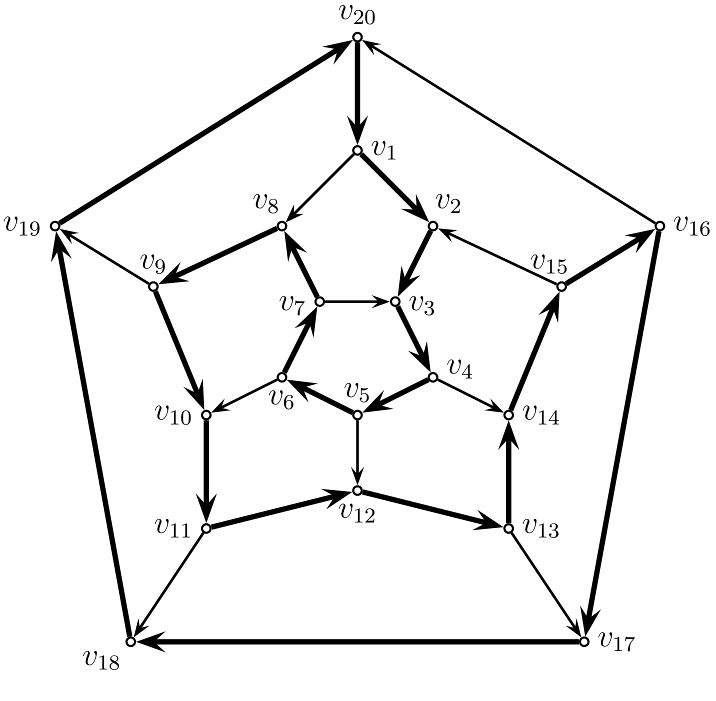

Figure 5.26: A Hamiltonian cycle in D

Proposition 5.9.4 *If a graph, $G = (V, E)$, possesses a Hamiltonian cycle then, for every nonempty set, S, of nodes, if $G\langle V - S\rangle$ is the induced subgraph of G generated by $V - S$ and if $c(G\langle V - S\rangle)$ is the number of connected components of $G\langle V - S\rangle$, then*

$$c(G\langle V - S\rangle) \leq |S|.$$

Proof. Let Γ be a Hamiltonian cycle in G and let $\widetilde{G}$ be the graph $\widetilde{G} = (V, \Gamma)$. If we delete k vertices we can't cut a cycle into more than k pieces and so

$$c(\widetilde{G}\langle V - S\rangle) \leq |S|.$$

However, we also have

$$c(\widetilde{G}\langle V - S\rangle) \leq c(G\langle V - S\rangle),$$

which proves the proposition. □

5.10 Network Flow Problems; The Max-Flow Min-Cut Theorem

The network flow problem is a perfect example of a problem which is important practically but also theoretically because in both cases it has unexpected applications. In this section, we solve the network flow problem using some of the notions from Section 5.7. First, let us describe the kinds of graphs that we are dealing with, usually called networks (or transportation networks or flow networks).

Definition 5.10.1 A *network* (or *flow network*) is a quadruple, $N = (G, c, v_s, s_t)$, where G is a finite diagraph, $G = (V, E, s, t)$, without loops, $c\colon E \to \mathbb{R}_+$, is a function called *capacity function* assigning a *capacity*, $c(e) > 0$, (or *cost* or *weight*), to every edge, $e \in E$, and $v_s, v_t \in V$ are two (distinct) distinguished nodes.[3] Moreover, we assume that there are no edges incoming into v_s ($d_G^-(v_s) = 0$), which is called the *source* and that there are no edges outgoing from v_t ($d_G^+(v_t) = 0$), which is called the *terminal* (or *sink*).

An example of a network is showed in Figure 5.27 with the capacity of each edge showed within parentheses.

Intuitively, we can think of the edges of a network as conduits for fluid, or wires for electricity, or highways for vehicle, *etc.*, and the capacity of each edge is the maximum amount of "flow" that can pass through that edge. The purpose of a network is to carry "flow", defined as follows:

Definition 5.10.2 Given a network, $N = (G, c, v_s, v_t)$, a *flow in N* is a function, $f\colon E \to \mathbb{R}$, such that the following conditions hold:

[3] Most books use the notation s and t for v_s and v_t. Sorry, s and t are already used in the definition of a digraph!

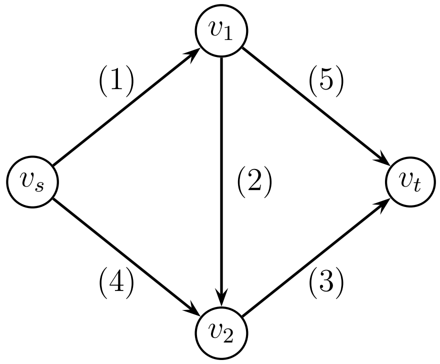

Figure 5.27: A network, N

(1) (Conservation of flow)

$$\sum_{t(e)=v} f(e) = \sum_{s(e)=v} f(e), \quad \text{for all } v \in V - \{v_s, v_t\}$$

(2) (Admissibility of flow)

$$0 \leq f(e) \leq c(e), \quad \text{for all } e \in E$$

Given any two sets of nodes, $S, T \subseteq V$, let

$$f(S,T) = \sum_{\substack{e \in E \\ s(e) \in S,\, t(e) \in T}} f(e) \quad \text{and} \quad c(S,T) = \sum_{\substack{e \in E \\ s(e) \in S,\, t(e) \in T}} c(e).$$

When $S = \{u\}$ or $T = \{v\}$, we write $f(u,T)$ for $f(\{u\},T)$ and $f(S,v)$ for $f(S,\{v\})$ (similarly, we write $c(u,T)$ for $c(\{u\},T)$ and $c(S,v)$ for $c(S,\{v\})$). The *net flow out of* S is defined as $f(S,\overline{S}) - f(\overline{S},S)$ (where $\overline{S} = V - S$). The *value,* $|f|$ *(or* $v(f)$*) of the flow* f is the quantity

$$|f| = f(v_s, V - \{v_s\}).$$

We can now state the

Network Flow Problem: Find a flow, f, in N, for which the value, $|f|$, is maximum (we call such a flow a *maximum flow*).

Figure 5.28 shows a flow in the network N, with value, $|f| = 3$. This is not a maximum flow, as the reader should check (the maximum flow value is 4).

Remarks:

1. For any set of edges, $\mathcal{E} \subseteq E$, let

$$\begin{aligned} f(\mathcal{E}) &= \sum_{e \in \mathcal{S}} f(e) \\ c(\mathcal{E}) &= \sum_{e \in \mathcal{S}} c(e). \end{aligned}$$

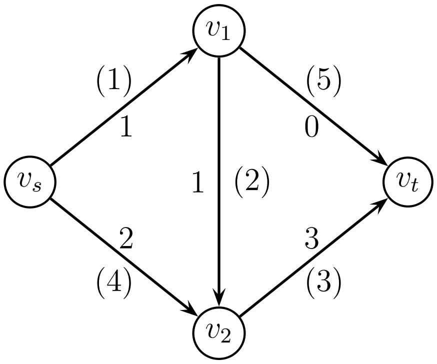

Figure 5.28: A flow in the network, N

Then, note that the net flow out of S can also be expressed as

$$f(\Omega^+(S)) - f(\Omega^-(S)) = f(S, \overline{S}) - f(\overline{S}, S).$$

Now, recall that $\Omega(S) = \Omega^+(S) \cup \Omega^-(S))$ is a cocycle (see Definition 5.7.5). So if we define the value, $f(\Omega(S))$, of the cocycle, $\Omega(S)$, to be

$$f(\Omega(S)) = f(\Omega^+(S)) - f(\Omega^-(S)),$$

the net flow through S is the value of the cocycle, $\Omega(S)$.

2. By definition, $c(S, \overline{S}) = c(\Omega^+(S))$.

3. Since G has no loops, there are no edges from u to itself, so

$$f(u, V - \{u\}) = f(u, V)$$

and similarly,

$$f(V - \{v\}, v) = f(V, v).$$

4. Some authors (for example, Wilf [45]) do not require the distinguished node, v_s, to be a source and the distinguished node, v_t, to be a sink. This makes essentially no difference but if so, the value of the flow f must be defined as

$$|f| = f(v_s, V - \{v_s\}) - f(V - \{v_s\}, v_s) = f(v_s, V) - f(V, v_s).$$

Intuitively, because flow conservation holds for every node except v_s and v_t, the net flow, $f(V, v_t)$, into the sink should be equal to the net flow, $f(v_s, V)$ our of the source, v_s. This is indeed true and follows from the following proposition:

Proposition 5.10.3 *Given a network, $N = (G, c, v_s, v_t)$, for any flow, f, in N and for any subset, $S \subseteq V$, if $v_s \in S$ and $v_t \notin S$, then the net flow through S has the same value, namely $|f|$, that is*

$$|f| = f(\Omega(S)) = f(S, \overline{S}) - f(\overline{S}, S) \leq c(S, \overline{S}) = c(\Omega^+(S)).$$

In particular,

$$|f| = f(v_s, V) = f(V, v_t).$$

Proof. Recall that $|f| = f(v_s, V)$. Now, for any node, $v \in S - \{v_s\}$, since $v \neq v_t$, the equation

$$\sum_{t(e)=v} f(e) = \sum_{s(e)=v} f(e)$$

holds and we see that

$$|f| = f(v_s, V) = \sum_{v \in S}\Big(\sum_{s(e)=v} f(e) - \sum_{t(e)=v} f(e)\Big) = \sum_{v \in S}(f(v, V) - f(V, v)) = f(S, V) - f(V, S).$$

However, $V = S \cup \overline{S}$, so

$$\begin{aligned} |f| &= f(S, V) - f(V, S) \\ &= f(S, S \cup \overline{S}) - f(S \cup \overline{S}, S) \\ &= f(S, S) + f(S, \overline{S}) - f(S, \overline{S}) - f(S, S) \\ &= f(S, \overline{S}) - f(S, \overline{S}), \end{aligned}$$

as claimed. Since the capacity of every edge is non-negative, it is obvious that

$$|f| = f(S, \overline{S}) - f(S, \overline{S}) \leq f(S, \overline{S}) \leq c(S, \overline{S}) = c(\Omega^+(S)),$$

since a flow is admissible. Finally, if we set $S = V - \{v_t\}$, we get

$$f(S, \overline{S}) - f(S, \overline{S}) = f(V, v_t)$$

and so, $|f| = f(v_s, V) = f(V, v_t)$. $\square$

Proposition 5.10.3 shows that the sets of edges, $\Omega^+(S)$, with $v_s \in S$ and $v_t \notin S$, play a very special role. Indeed, as a corollary of Proposition 5.10.3, we see that the value any flow in N is bounded by the capacity, $c(\Omega^+(S))$, of the set $\Omega^+(S)$, for any S with $v_s \in S$ and $v_t \notin S$. This suggests the following definition:

Definition 5.10.4 Given a network, $N = (G, c, v_s, v_t)$, a *cut separating v_s and v_t, for short a v_s-v_t-cut*, is any subset of edges, $\mathcal{C} = \Omega^+(W)$, where W is a subset of V with $v_s \in W$ and $v_t \notin W$. The *capacity of a v_s-v_t-cut, $\mathcal{C}$,* is

$$c(\mathcal{C}) = c(\Omega^+(W)) = \sum_{e \in \Omega^+(W)} c(e).$$

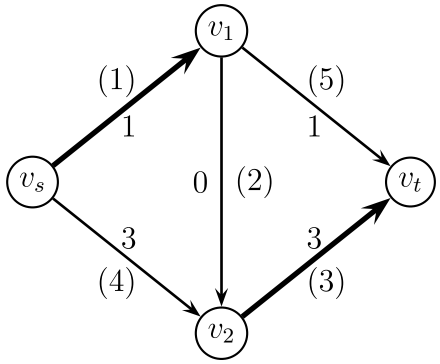

Figure 5.29: A maximum flow and a minimum cut in the network, N

Remark: Some authors, including Papadimitriou and Steiglitz [35] and Wilf [45], define a v_s-v_t-cut as a pair $(W, \overline{W})$, where W is a subset of V with with $v_s \in W$ and $v_t \notin W$. This definition is clearly equivalent to our definition above, which is due to Sakarovitch [39]. We have a slight prerefence for Definition 5.10.4 because it places the emphasis on edges as opposed to nodes. Indeed, the intuition behind v_s-v_t-cuts is that any flow from v_s to v_t must pass through some edge of any v_s-v_t-cut. Thus, it is not surprising that the capacity of v_s-v_t-cuts places a restriction on how much flow can be sent from v_s to v_t.

We can rephrase Proposition 5.10.3 as follows:

Proposition 5.10.5 *The maximum value of any flow, f, in N is bounded by the minimum capacity, $c(\mathcal{C})$, of any v_s-v_t-cut, $\mathcal{C}$, in N, i.e.,*

$$\max |f| \leq \min c(\mathcal{C}).$$

Proposition 5.10.5 is half of the so-called *Max-flow Min-cut Theorem*. The other half of this theorem says that the above inequality is indeed an equality. That is, there is actually some v_s-v_t-cut, $\mathcal{C}$, whose capacity, $c(\mathcal{C})$, is the maximum value of the flow in N.

A v_s-v_t-cut of minimum capacity is called a *minimum v_s-v_t-cut*, for short, a *minimum cut*.

An example of a minimum cut is shown in Figure 5.29, where

$$\mathcal{C} = \Omega^+(\{v_s, v_2\}) = \{(v_s v_1), (v_2 v_t)\},$$

these two edges being shown as thicker lines. The capacity of this cut is 4 and a maximum flow is also shown in Figure 5.29.

What we intend to do next is to prove the celebrated "Max-flow, Min-cut Theorem" (due to Ford and Fulkerson, 1957) and then to give an algorithm (also due to Ford and Fulkerson) for finding a maximum flow, provided some reasonable assumptions on the capacity function.

In preparation for this, we present a handy trick (found both in Berge [3] and Sakarovitch [39]), the *return edge*.

Recall that one of the consequences of Proposition 5.10.3 is that the net flow out from v_s is equal to the net flow into v_t. Thus, if we add a new edge, e_r, called the *return edge*, to G, obtaining the graph $\widetilde{G}$ (and the network $\widetilde{N}$), we see that any flow, f, in N satisfying condition (1) of Definition 5.10.2 yields a genuine flow, $\widetilde{f}$, in $\widetilde{N}$ (a flow according to Definition 5.7.10, by Theorem 5.7.18), such that $f(e) = \widetilde{f}(e)$ for every edge of G and $\widetilde{f}(e_r) = |f|$. Consequently, the Network flow problem is equivalent to find a (genuine) flow in $\widetilde{N}$ such that $\widetilde{f}(e_r)$ is maximum. Another advantage of this formulation is that all the results on flows from Section 5.7 can be applied directly to $\widetilde{N}$. To simplify the notation, as $\widetilde{f}$ extends f, let us also use thw notation f for $\widetilde{f}$. Now, if D is the indicence matrix of $\widetilde{G}$ (again, we use the simpler notation, D, instead of $\widetilde{D}$), we know that f is a flow iff

$$Df = 0.$$

Therefore, the network flow problem can be stated as a *linear programing problem* as follows:

$$\text{Maximize } z = f(e_r)$$

subject to the linear constraints

$$\begin{aligned} Df &= 0 \\ 0 &\leq f \\ f &\leq c, \end{aligned}$$

where we view f as a vector in $\mathbb{R}^{n+1}$, with $n = |E(G)|$.

Consequently, we obtain the existence of maximal flows, a fact which is not immediately obvious.

Proposition 5.10.6 *Given any network, $N = (G, c, v_s, v_t)$, there is some flow, f, of maximum value.*

Proof. If we go back to the formulation of the Max-flow problem as a linear program, we see that the set

$$C = \{x \in \mathbb{R}^{n+1} \mid 0 \leq x \leq c\} \cap \mathrm{Ker}\, D$$

is compact, as the intersection of a compact subset and a closed subset of $\mathbb{R}^{n+1}$ (in fact, C is also convex) and nonempty, as 0 (the zero vector) is a flow. But then, the projection, $\pi\colon x \mapsto x(e_r)$, is a continuous function, $\pi\colon C \to \mathbb{R}$, on a nonempty compact, so it achieves its maximum value for some $f \in C$. Such an f is a flow on $\widetilde{N}$ with maximal value. □

Now that we know that maximum flows exist, it remains to prove that a maximal flow is realized by some minimal cut to complete the Max-flow, Min-cut Theorem of Ford and

Fulkerson. This can be done in various ways usually using some version of an algorithm due to Ford and Fulkerson. Such proofs can be found in Papadimitriou and Steiglitz [35], Wilf [45], Cameron [8] and Sakarovitch [39].

Sakarovitch makes the interesting observation (given as an exercise) that the Arc Coloring Lemma due to Minty (Theorem 5.7.11) yields a simple proof of the part of the Max-flow, Min-cut Theorem that we seek to establish (See [39], Chapter 4, Exercise 1, page 105). Therefore, we choose to present such a proof since it is rather original and quite elegant.

Theorem 5.10.7 *(Max-Flow, Min-Cut Theorem (Ford and Fulkerson)) For any network, $N = (G, c, v_s, v_t)$, the maximum value, $|f|$, of any flow, f, in N is equal to the minimum capacity, $c(\mathcal{C})$, of any v_s-v_t-cut, $\mathcal{C}$, in N.*

Proof. By Proposition 5.10.5, we already have half of our theorem. By Proposition 5.10.6, we know that some maximum flow, say f, exists. It remains to show that there is some v_s-v_t-cut, $\mathcal{C}$, such that $|f| = c(\mathcal{C})$.

We proceed as follows:

Form the graph, $\widetilde{G} = (V, E \cup \{e_r\}, s, t)$ from $G = (V, E, s, t)$, with $s(e_r) = v_t$ and $t(e_r) = v_s$. Then, form the graph, $\widehat{G} = (V, \widehat{E}, \widehat{s}, \widehat{t})$, whose edges are defined as follows:

(a) $e_r \in \widehat{E}$; $\widehat{s}(e_r) = s(e_r)$, $\widehat{t}(e_r) = t(e_r)$;

(b) If $e \in E$ and $0 < f(e) < c(e)$, then $e \in \widehat{E}$; $\widehat{s}(e) = s(e)$, $\widehat{t}(e) = t(e)$;

(c) If $e \in E$ and $f(e) = 0$, then $e \in \widehat{E}$; $\widehat{s}(e) = s(e)$, $\widehat{t}(e) = t(e)$;

(d) If $e \in E$ and $f(e) = c(e)$, then $e \in \widehat{E}$, with $\widehat{s}(e) = t(e)$ and $\widehat{t}(e) = s(e)$.

In order to apply Minty's Theorem, we color all edges constructed in (a), (c) and (d) in black and all edges constructed in (b) in red and we pick e_r as the distinguished edge. Now, apply Minty's Lemma. We have two possibilities:

1. There is an elementary cycle, Γ, in $\widehat{G}$, with all black edges oriented the same way. Since e_r is incoming into v_s, the direction of the cycle is from from v_s to v_t, so $e_r \in \Gamma^+$. This implies that all edges of type (d), $e \in \widehat{E}$, have an orientation consistent with the direction of the cycle. Now, Γ is also a cycle in $\widetilde{G}$ and, in $\widetilde{G}$, each edge, $e \in E$, with $f(e) = c(e)$ is oriented in the inverse direction of the cycle, i.e, $e \in \Gamma^{-1}$ in $\widetilde{G}$. Also, all edges of type (c), $e \in \widehat{E}$, with $f(e) = 0$, are oriented in the direction of the cycle, i.e., $e \in \Gamma^+$ in $\widetilde{G}$. We also have $e_r \in \Gamma^+$ in $\widetilde{G}$.

   We show that the value of the flow, $|f|$, can be increased. Since $0 < f(e) < c(e)$ for every red edge, $f(e) = 0$ for every edge of type (c) in Γ^+, $f(e) = c(e)$ for every edge of type (d) in Γ^-, and since all capacities are strictly positive, if we let

$$\begin{aligned} \delta_1 &= \min_{e \in \Gamma^+} \{c(e) - f(e)\} \\ \delta_2 &= \min_{e \in \Gamma^-} \{f(e)\} \end{aligned}$$

and

$$\delta = \min\{\delta_1, \delta_2\},$$

then $\delta > 0$. We can increase the flow, f, in $\widetilde{N}$, by adding δ to $f(e)$ for every edge $e \in \Gamma^+$ (including edges of type (c) for which $f(e) = 0$) and subtracting δ from $f(e)$ for every edge $e \in \Gamma^-$ (including edges of type (d) for which $f(e) = c(e)$) obtaining a flow, f' such that

$$|f'| = f(e_r) + \delta = |f| + \delta > |f|,$$

as $e_r \in \Gamma^+$, contradicting the maximality of f. Therefore, we conclude that alternative (1) is impossible and we must have the second alternative:

2. There is an elementary cocycle, $\Omega_{\widehat{G}}(W)$, in $\widehat{G}$ with all edges black and oriented in the same direction (there are no green edges). Since $e_r \in \Omega_{\widehat{G}}(W)$, either $v_s \in W$ or $v_t \in W$ (but not both). In the second case ($v_t \in W$), we have $e_r \in \Omega^+_{\widehat{G}}(W)$ and $v_s \in \overline{W}$. Then, consider $\Omega^+_{\widehat{G}}(\overline{W}) = \Omega^-_{\widehat{G}}(W)$, with $v_s \in \overline{W}$. Thus, we are reduced to the case where $v_s \in W$.

   If $v_s \in W$, then $e_r \in \Omega^-_{\widehat{G}}(W)$ and since all edges are black, $\Omega_{\widehat{G}}(W) = \Omega^-_{\widehat{G}}(W)$, in $\widehat{G}$. However, as every edge, $e \in \widehat{E}$, of type (d) corresponds to an inverse edge, $e \in E$, we see that $\Omega_{\widehat{G}}(W)$ defines a cocycle, $\Omega_{\widetilde{G}}(W) = \Omega^+_{\widetilde{G}}(W) \cup \Omega^-_{\widetilde{G}}(W)$, with

$$\begin{aligned} \Omega^+_{\widetilde{G}}(W) &= \{e \in E \mid s(e) \in W\} \\ \Omega^-_{\widetilde{G}}(W) &= \{e \in E \mid t(e) \in W\}. \end{aligned}$$

   Moreover, by construction, $f(e) = c(e)$ for all $e \in \Omega^+_{\widetilde{G}}(W)$, $f(e) = 0$ for all
   $e \in \Omega^-_{\widetilde{G}}(W) - \{e_r\}$, and $f(e_r) = |f|$. We say that the edges of the cocycle $\Omega_{\widetilde{G}}(W)$ are *saturated*. Consequently, $\mathcal{C} = \Omega^+_{\widetilde{G}}(W)$ is a v_s-v_t-cut in N with

$$c(\mathcal{C}) = f(e_r) = |f|,$$

   establishing our theorem. □

It is interesting that the proof in part (1) of Theorem 5.10.7 contains the main idea behind the algorithm of Ford and Fulkerson that we now describe.

The main idea is to look for an (elementary) chain from v_s to v_t so that together with the return edge, e_r, we obtain a cycle, Γ, such that the edges in Γ satisfy the following properties:

(1) $\delta_1 = \min_{e \in \Gamma^+}\{c(e) - f(e)\} > 0$;

(2) $\delta_2 = \min_{e \in \Gamma^-}\{f(e)\} > 0$.

Such a chain is called a *flow augmenting chain*. Then, if we let $\delta = \min\{\delta_1, \delta_2\}$, we can increase the value of the flow by adding δ to $f(e)$ for every edge $e \in \Gamma^+$ (including the edge, e_r, which belongs to Γ^+) and subtracting δ from $f(e)$ for all edges $e \in \Gamma^-$. This way, we get a new flow, f', whose value is $|f'| = |f| + \delta$. Indeed, $f' = f + \delta\gamma(\Gamma)$, where $\gamma(\Gamma)$ is the flow associated with the cycle, Γ. The algorithms goes through rounds each consisting of two phases: During phase 1, a flow augmenting chain is found by the procedure *findchain*; During phase 2, the flow along the edges of the augmenting chain is increased using the function *changeflow*.

During phase 1, the nodes of the augmenting chain are saved in the (set) variable, Y, and the edges of this chain are saved in the (set) variable, $\mathcal{E}$. We assign the special capacity value ∞ to e_r, with the convention that $\infty \pm \alpha = \alpha$ and that $\alpha < \infty$ for all $\alpha \in \mathbb{R}$.

```
procedure findchain($N$: network; $e_r$: edge; $Y$: node set; $\mathcal{E}$: edge set; $\delta$; real; $f$; flow)
  begin
    $\delta := \delta(v_s) := \infty$; $Y := \{v_s\}$;
    while $(v_t \notin Y) \wedge (\delta > 0)$ do
      if there is an edge $e$ with $s(e) \in Y$, $t(e) \notin Y$ and $f(e) < c(e)$ then
        $Y := Y \cup \{t(e)\}$; $\mathcal{E}(t(e)) := e$; $\delta(t(e)) := \min\{\delta(s(e)), c(e) - f(e)\}$
      else
        if there is an edge $e$ with $t(e) \in Y$, $s(e) \notin Y$ and $f(e) > 0$ then
          $Y := Y \cup \{s(e)\}$; $\mathcal{E}(s(e)) := e$; $\delta(s(e)) := \min\{\delta(t(e)), f(e)\}$
        else $\delta := 0$ (no new arc can be traversed)
        endif
      endif
    endwhile;
    if $v_t \in Y$ then $\delta := \delta(v_t)$ endif
  end
```

Here is now the procedure to update the flow:

```
procedure changeflow($N$: network; $e_r$: edge; $\mathcal{E}$: edge set; $\delta$: real; $f$; flow)
  begin
    $u := v_t$; $f(e_r) := f(e_r) + \delta$;
    while $u \neq v_s$ do $e := \mathcal{E}(u)$;
      if $u = t(e)$ then $f(e) := f(e) + \delta$; $u := s(e)$;
      else $f(e) := f(e) - \delta$; $u = t(e)$
      endif
    endwhile
  end
```

Finally, the algorithm *maxflow* is given below:

```
procedure maxflow($N$: network; $e_r$: edge; $Y$: set of nodes; $\mathcal{E}$: set of edges; $f$; flow)
  begin
    for each $e \in E$ do $f(e) := 0$ enfdor;
    repeat until $\delta = 0$
      findchain($N, e_r, Y, \mathcal{E}, \delta, f$);
      if $\delta > 0$ then
        changeflow($N, e_r, \mathcal{E}, \delta, f$)
      endif
    endrepeat
  end
```

The reader should run the algorithm *maxflow* on the network of Figure 5.27 to verify that the maximum flow shown in Figure 5.29 is indeed found, with $Y = \{v_s, v_2\}$ when the algorithm stops.

The correctness of the algorithm *maxflow* is easy to prove.

Theorem 5.10.8 *If the algorithm, maxflow, terminates and during the last round through findchain the node v_t is not marked, then the flow, f, returned by the algorithm is a maximum flow.*

Proof. Observe that if Y is the set of nodes returned when *maxflow* halts, then $v_s \in Y$, $v_t \notin Y$ and

1. If $e \in \Omega^+(Y)$, then $f(e) = c(e)$, as otherwise, procedure *findchain* would have added $t(e)$ to Y;

2. If $e \in \Omega^-(Y)$, then $f(e) = 0$, as otherwise, procedure *findchain* would have added $s(e)$ to Y.

But then, as in the end of the proof of Theorem 5.10.7, we see that the edges of the coycle $\Omega(Y)$ are saturated and we know that $\Omega^+(Y)$ is a minimal cut and that $|f| = c(\Omega^+(Y))$ is maximal. □

We still have to show that the algorithm terminates but there is a catch. Indeed, the version of the Ford and Fulkerson algorithm that we just presented may not terminate if the capacities are irrational! Moreover, in the limit, the flow found by the algorithm may not be maximum! An example of this bad behavior due to Ford and Fulkerson is reproduced in Wilf [45] (Chapter 3, Section 5). However, we can prove the following termination result which, for all practical purposes, is good enough, since only rational numbers can be stored by a computer.

Theorem 5.10.9 *Given a network, N, if all the capacities are multiple of some number, λ, then the algorithm, maxflow, always terminates. In particular, the algorithm maxflow always terminates if the capacites are rational (or integral).*

Proof. The number δ will always be a multiple of λ, so $f(e_r)$ will increase by at least λ during each iteration. Thus, eventually, the value of a minimal cut, which is a multiple of λ, will be reached. $\square$

If all the capacities are integers, an easy induction yields the following useful and nontrivial proposition:

Proposition 5.10.10 *Given a network, N, if all the capacities are integers, then the algorithm maxflow outputs a maximum flow, $f\colon E \to \mathbb{N}$, such that the flow in every edge is an integer.*

Remark: Proposition 5.10.10 only asserts that some maximum flow is of the form $f\colon E \to \mathbb{N}$. In general, there is more than one maximum flow and other maximum flows may not have integer values on all edges.

Theorem 5.10.9 is good news but it is also bad news from the point of view of complexity. Indeed, the present version of the Ford and Fulkerson algorithm has a running time that depends on capacities and so, it can be very bad.

There are various ways of getting around this difficulty to find algorithms that do not depend on capacities and quite a few researchers have studied this problem. An excellent discussion of the progress in network flow algorithms can be found in Wilf (Chapter 3).

A fairly simple modification of the Ford and Fulkerson algorithm consists in looking for flow augmenting chains of shortest length. To explain this algorithm we need the concept of *residual network*, which is a useful tool in any case. Given a network, $N = (G, c, s, t)$ and given any flow, f, the *residual network*, $N_f = (G_f, c_f, v_f, v_t)$ is defined as follows:

1. $V_f = V$;
2. For every edge, $e \in E$, if $f(e) < c(e)$, then $e^+ \in E_f$, $s_f(e^+) = s(e)$, $t_f(e^+) = t(e)$ and $c_f(e^+) = c(e) - f(e)$; the edge e^+ is called a *forward edge*;
3. For every edge, $e \in E$, if $f(e) > 0$, then $e^- \in E_f$, $s_f(e^-) = t(e)$, $t_f(e^-) = s(e)$ and $c_f(e^-) = f(e)$; the edge e^- is called a *backward edge* because it has the inverse orientation of the original edge, $e \in E$;

The capacity, $c_f(e^\epsilon)$, of an edge $e^\epsilon \in E_f$ (with $\epsilon = \pm$) is usually called the *residual capacity* of e^ϵ. Observe that the same edge, e, in G, will give rise to two edges e^+ and e^- (with the same set of endpoints but with opposite orientations) in G_f if $0 < f(e) < c(e)$. Thus, G_f has at most twice as many edges as G. Also, note that every edge, $e \in E$, which is *saturated*, i.e., for which $f(e) = c(e)$, does not survive in G_f.

Observe that there is a one-to-one correspondence between (elementary) flow augmenting chains in the original graph, G, and (elementary) flow augmenting paths in G_f. Furthermore,

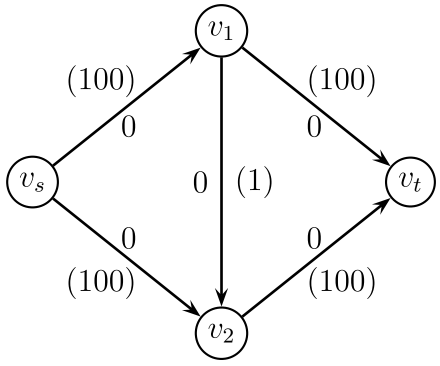

Figure 5.30: A poor choice of augmenting paths yields a slow method

in order to check that an elementary path, π, from v_s to v_t in G_f is a flow augmenting path, all we have to do is to compute

$$c_f(\pi) = \min_{e^\epsilon \in \pi}\{c_f(e^\epsilon)\},$$

the *bottleneck* of the path, π. Then, as before, we can update the flow, f in N, to get the new flow, f', by setting

$$\begin{aligned}
f'(e) &= f(e) + c_f(\pi), && \text{if} \quad e^+ \in \pi \\
f'(e) &= f(e) - c_f(\pi) && \text{if} \quad e^- \in \pi, \\
f'(e) &= f(e) && \text{if} \quad e \in E \quad \text{and} \quad e^\epsilon \notin \pi,
\end{aligned}$$

for every edge $e \in E$. Note that the function, $f_\pi\colon E \to \mathbb{R}$, defined by

$$\begin{aligned}
f_\pi(e) &= c_f(\pi), && \text{if} \quad e^+ \in \pi \\
f_\pi(e) &= -c_f(\pi) && \text{if} \quad e^- \in \pi, \\
f_\pi(e) &= 0 && \text{if} \quad e \in E \quad \text{and} \quad e^\epsilon \notin \pi,
\end{aligned}$$

is a flow in N with $|f_\pi| = c_f(\pi)$ and $f' = f + f_{N,\pi}$ is a flow in N, with $|f'| = |f| + c_f(\pi)$ (same reasoning as before). Now, we can repeat this process: Compute the new residual graph, $N_{f'}$ from N and f', update the flow f' to get the new flow f'' in N, etc.

The same reasoning as before shows that if we obtain a residual graph with no flow augmenting path from v_s to v_t, then a maximum flow has been found.

It should be noted that a poor choice of augmenting paths may cause the algorithm to perform a lot more steps than necessary. For example, if we consider the network shown in Figure 5.30, and if we pick the flow augmenting paths in the residual graphs to be alternatively (v_s, v_1, v_2, v_t) and (v_s, v_2, v_1, v_t), at each step, we only increase the flow by 1, so it will take 200 steps to find a maximum flow!

One of the main advantages of using residual graphs is that they make is convenient to look for better strategies for picking flow augmenting paths. For example, we can choose an

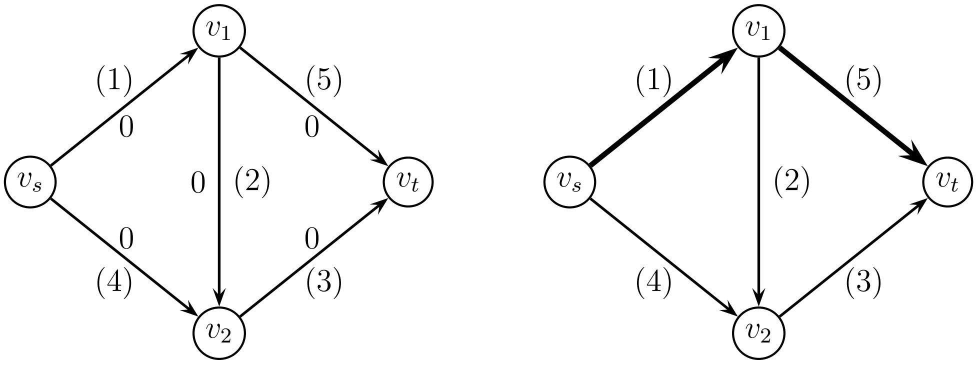

Figure 5.31: Construction of the residual graph, N_f, from N, round 1

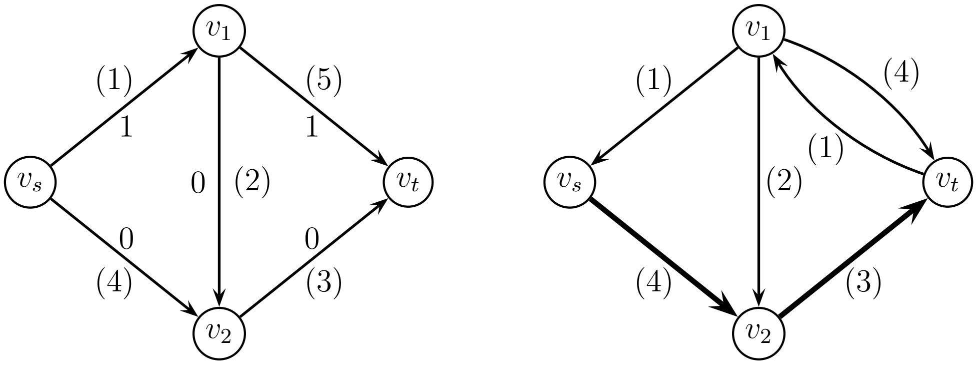

Figure 5.32: Construction of the residual graph, N_f, from N, round 2

elementary flow augmenting path shortest length (for example, using breadth-first search). Then, it can be shown that this revised algorithm terminates in $O(|V|\cdot|E|)$ steps (see Cormen, Leiserson, Rivest and Stein [11], Section 26.2, and Sakarovitch [39], Chapter 4, Exercise 5). Edmonds and Karp designed an algorithm running in time $O(|E|\cdot|V|^2)$ based on this idea (1972), see [11], Section 26.2. Another way of selecting "good" augmenting paths, the *scaling Max-Flow algorithm*, is described in Kleinberg and Tardos [33] (see Section 7.3).

Here is an illustration of this faster algorithm, starting with the network, N, shown in Figure 5.27. The sequence of residual network construction and flow augmentation steps is shown in Figures 5.31, 5.32 and 5.33. During the first two rounds, the augmented path chosen is shown in thicker lines. In the third and final round, there is no path from v_s to v_t in the residual graph, indicating that a maximum flow has been found.

Another idea originally due to Dinic (1970) is to use *layered networks*, see Wilf [45] (Sections 3.6-3.7) and Papadimitriou and Steiglitz [35] (Chapter 9). An algorithm using layered networks running in time $O(V^3)$ is given in the two references above. There are yet other faster algorithms, for instance "preflow-push algorithms" also called "preflow-push relabel algorithms", originally due to Goldberg. A *preflow* is a function, $f\colon E \to \mathbb{R}$, that satisfies condition (2) of Definition 5.10.2 but which, instead of satisfying condition (1),

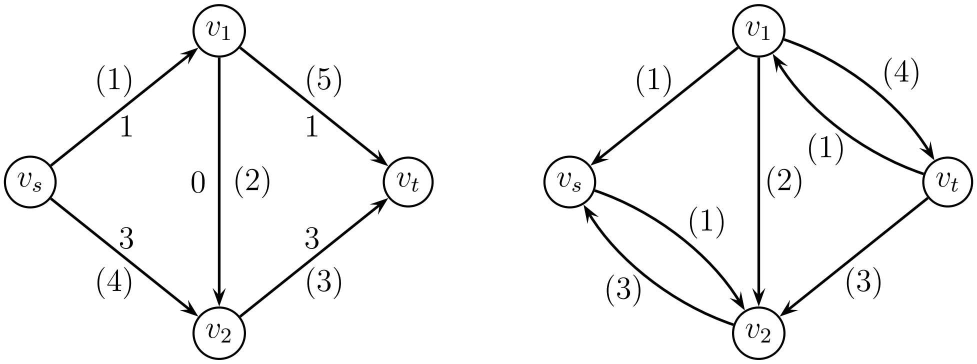

Figure 5.33: Construction of the residual graph, N_f, from N, round 3

satisfies the inequality

(1′) (Non-negativity of net flow)

$$\sum_{s(e)=v} f(e) \geq \sum_{t(e)=v} f(e) \quad \text{for all } v \in V - \{v_s, v_t\},$$

that is, the net flow out of v is non-negative. Now, the principle of all methods using preflows is to augment a preflow until it becomes a maximum flow. In order to do this, a labeling algorithm assigning a *height*. Algorithms of this type are discussed in Cormen, Leiserson, Rivest and Stein [11], Sections 26.4 and 26.5 and in Kleinberg and Tardos [33], Section 7.4.

The Max-flow, Min-cut Theorem (Theorem 5.10.7) is a surprisingly powerful theorem in the sense that it can be used to prove a number of other results whose original proof is sometimes quite hard. Among these results, let us mention the *maximum matching problem* in a bipartite graph, discussed in Wilf [45] (Sections 3.8), Cormen, Leiserson, Rivest and Stein [11] (Section 26.3) Kleinberg and Tardos [33] (Section 7.5) and Cameron [8] (Chapter 11, Section 10), finding the *edge connectivity* of a graph, discussed in Wilf [45] (Sections 3.8), and a beautiful *theorem of Menger* on edge-disjoint paths and *Hall's Marriage Theorem*, both discussed in Cameron [8] (Chapter 11, Section 10). More problems that can be solved effectively using flow algorithms, including image segmentation, are discussed in Sections 7.6–7.13 of Kleinberg and Tardos [33]. We only mention one of Menger's theorems, as it is particularly elegant.

Theorem 5.10.11 *(Menger) Given any finite digraph, G, for any two nodes, v_s and v_t, the maximum number of pairwise edge-disjoint paths from v_s to v_t is equal to the the minimum number of edges in a v_s-v_t-separating set. (A a v_s-v_t-separating set in G is a set of edges, C, such every path from v_s to v_t uses some edge in C.)*

It is also possible to generalize the basic flow problem in which our flows, f, have the property that $0 \leq f(e) \leq c(e)$ for every edge, $e \in E$, to *channeled flows*. This generalization consists in adding another capacity function, $b\colon E \to \mathbb{R}$, relaxing the condition that $c(e) > 0$ for all $e \in E$, and in allowing flows such that condition (2) of Definition 5.10.2 is replaced by

(2′) (Admissibility of flow)

$$b(e) \leq f(e) \leq c(e), \quad \text{for all } e \in E$$

Now, the "flow" $f = 0$ is no longer necessarily admissible and the channeled flow problem does not always have a solution. However, it is possible to characterize when it has a solution.

Theorem 5.10.12 *(Hoffman) A network, $N = (G, b, c, v_s, v_t)$, has a channeled flow iff for every cocycle, $\Omega(Y)$, of G, we have*

$$\sum_{e\in\Omega^-(Y)} b(e) \leq \sum_{e\in\Omega^+(Y)} c(e). \tag{$\dagger$}$$

Observe that the necessity of the condition of Theorem 5.10.12 is an immediate consequence of Proposition 5.7.9. That it is sufficient can be proved by modifying the algorithm *maxflow* or its version using residual networks. The principle of this method is to start with a flow, f, in N that does not necessarily satisfy condition (2′) and to gradually convert it to an admissible flow in N (if one exists) by applying the method for finding a maximum flow to a modified version, $\widetilde{N}$, of N in which the capacities have been adjusted to that f is an admissible flow in $\widetilde{N}$. Now, if a flow, f, in N does not satisfy condition (2′), then there are some *offending edges*, e, for which either $f(e) < b(e)$ or $f(e) > c(e)$. The new method makes sure that at the end of every (successful) round through the basic *maxflow* algorithm applied to the modified network, $\widetilde{N}$, some offending edge of N is no longer offending.

Let f be a flow in N and assume that $\widetilde{e}$ is an offending edge (i.e. either $f(e) < b(e)$ or $f(e) > c(e)$). Then, we construct the network, $\widetilde{N}(f, \widetilde{e})$, as follows: The capacity functions, $\widetilde{b}$ and $\widetilde{c}$ are given by

$$\widetilde{b}(e) = \begin{cases} b(e) & \text{if } b(e) \leq f(e) \\ f(e) & \text{if } f(e) < b(e) \end{cases}$$

and

$$\widetilde{c}(e) = \begin{cases} c(e) & \text{if } f(e) \leq c(e) \\ f(e) & \text{if } f(e) > c(e). \end{cases}$$

We also add one new edge, $\widetilde{e}_r$, to N whose endpoints and capacities are determined by:

1. If $f(\widetilde{e}) > c(\widetilde{e})$, then $s(\widetilde{e}_r) = t(\widetilde{e})$, $t(\widetilde{e}_r) = s(\widetilde{e})$, $\widetilde{b}(\widetilde{e}_r) = 0$ and $\widetilde{c}(\widetilde{e}_r) = f(\widetilde{e}) - c(\widetilde{e})$.

2. If $f(\widetilde{e}) < b(\widetilde{e})$, then $s(\widetilde{e}_r) = s(\widetilde{e})$, $t(\widetilde{e}_r) = t(\widetilde{e})$, $\widetilde{b}(\widetilde{e}_r) = 0$ and $\widetilde{c}(\widetilde{e}_r) = b(\widetilde{e}) - f(\widetilde{e})$.

Now, observe that the original flow, f, in N extended so that $f(\widetilde{e}_r) = 0$ is a channeled flow in $\widetilde{N}(f, \widetilde{e})$ (i.e., conditions (1) and (2′) are satisfied). Starting from the new network, $\widetilde{N}(f, \widetilde{e})$, apply the Max-flow algorithm, say using residual graphs, with the following small change in 2:

1. For every edge, $e \in \widetilde{E}$, if $f(e) < \widetilde{c}(e)$, then $e^+ \in \widetilde{E}_f$, $s_f(e^+) = s(e)$, $t_f(e^+) = t(e)$ and $c_f(e^+) = \widetilde{c}(e) - f(e)$; the edge e^+ is called a *forward edge*;

2. For every edge, $e \in \widetilde{E}$, if $f(e) > \widetilde{b}(e)$, then $e^- \in \widetilde{E}_f$, $s_f(e^-) = t(e)$, $t_f(e^-) = s(e)$ and $c_f(e^-) = f(e) - \widetilde{b}(e)$; the edge e^- is called a *backward edge*.

Now, we consider augmenting paths from $t(\widetilde{e}_r)$ to $s(\widetilde{e}_r)$. For any such elementary path, π, in $\widetilde{N}(f, \widetilde{e})_f$, as before we compute

$$c_f(\pi) = \min_{e^\epsilon \in \pi} \{c_f(e^\epsilon)\},$$

the *bottleneck* of the path, π, and we say that π is a flow augmenting path iff $c_f(\pi) > 0$. Then, we can update the flow, f in $\widetilde{N}(f, \widetilde{e})$, to get the new flow, f', by setting

$$\begin{aligned}
f'(e) &= f(e) + c_f(\pi) && \text{if} \quad e^- \in \pi, \\
f'(e) &= f(e) - c_f(\pi) && \text{if} \quad e^- \in \pi, \\
f'(e) &= f(e) && \text{if} \quad e \in \widetilde{E} \quad \text{and} \quad e^\epsilon \notin \pi,
\end{aligned}$$

for every edge $e \in \widetilde{E}$.

We run the flow augmenting path procedure on $\widetilde{N}(f, \widetilde{e})$ and f until it terminates with a maximum flow, $\widetilde{f}$. If we recall that the offending edge is $\widetilde{e}$, then, there are four cases:

1. $f(\widetilde{e}) > c(\widetilde{e})$.

   (a) When the Max-flow algorithm terminates, $\widetilde{f}(\widetilde{e}_r) = \widetilde{c}(\widetilde{e}_r) = f(\widetilde{e}) - c(\widetilde{e})$. If so, define $\widehat{f}$ as follows:
   $$\widehat{f}(e) = \begin{cases} \widetilde{f}(\widetilde{e}) - \widetilde{f}(\widetilde{e}_r) & \text{if } e = \widetilde{e} \\ \widetilde{f}(e) & \text{if } e \neq \widetilde{e}. \end{cases} \tag{$*$}$$
   It is clear that $\widehat{f}$ is a flow in N and $\widehat{f}(\widetilde{e}) = c(\widetilde{e})$ (there are no elementary paths from $t(\widetilde{e})$ to $s(\widetilde{e})$). But then, $\widetilde{e}$ is not an offending edge for $\widehat{f}$, so we repeat the procedure of constructing the modified network, etc.

   (b) When the Max-flow algorithm terminates, $\widetilde{f}(\widetilde{e}_r) < \widetilde{c}(\widetilde{e}_r)$. The flow, $\widehat{f}$, defined in ($*$) above is still a flow but the Max-flow algorithm must have terminated with a residual graph with no flow augmenting path from $s(\widetilde{e})$ to $t(\widetilde{e})$. Then, there is a set of nodes, Y with $s(\widetilde{e}) \in Y$ and $t(\widetilde{e}) \notin Y$. Moreover, the way the Max-flow algorithm is designed implies that

$$\begin{aligned} &\widehat{f}(\widetilde{e}) > c(\widetilde{e}) \\ &\widehat{f}(e) = \widetilde{c}(e) \geq c(e) \qquad \text{if} \quad e \in \Omega^+(Y) - \{\widetilde{e}\} \\ &\widehat{f}(e) = \widetilde{b}(e) \leq b(e) \qquad \text{if} \quad e \in \Omega^-(Y). \end{aligned}$$

As $\widehat{f}$ also satisfies $(*)$ above, we conclude that the cocycle condition $(\dagger)$ of Theorem 5.10.12 fails for $\Omega(Y)$.

2. $f(\widetilde{e}) < b(\widetilde{e})$.

   (a) When the Max-flow algorithm terminates, $\widetilde{f}(\widetilde{e}_r) = \widetilde{c}(\widetilde{e}_r) = b(\widetilde{e}) - f(\widetilde{e})$. If so, define $\widehat{f}$ as follows:

   $$\widehat{f}(e) = \begin{cases} \widetilde{f}(\widetilde{e}) + \widetilde{f}(\widetilde{e}_r) & \text{if } e = \widetilde{e} \\ \widetilde{f}(e) & \text{if } e \neq \widetilde{e}. \end{cases} \tag{$**$}$$

   It is clear that $\widehat{f}$ is a flow in N and $\widehat{f}(\widetilde{e}) = b(\widetilde{e})$ (there are no elementary paths from $s(\widetilde{e})$ to $t(\widetilde{e})$). But then, $\widetilde{e}$ is not an offending edge for $\widehat{f}$, so we repeat the procedure of constructing the modified network, etc.

   (b) When the Max-flow algorithm terminates, $\widetilde{f}(\widetilde{e}_r) < \widetilde{c}(\widetilde{e}_r)$. The flow, $\widehat{f}$, defined in $(**)$ above is still a flow but the Max-flow algorithm must have terminated with a residual graph with no flow augmenting path from $t(\widetilde{e})$ to $s(\widetilde{e})$. Then, as in the case where $f(\widetilde{e}) > c(\widetilde{e})$, there is a set of nodes, Y with $s(\widetilde{e}) \in Y$ and $t(\widetilde{e}) \notin Y$ and it is easy to show that the cocycle condition $(\dagger)$ of Theorem 5.10.12 fails for $\Omega(Y)$.

Therefore, if the algorithm does not fail during every round through the Max-flow algorithm applied to the modified network, $\widetilde{N}$, which, as we observed, is the case if condition $(\dagger)$ holds, then a channeled flow, $\widehat{f}$, will be produced and this flow will be a maximum flow. This proves the converse of Theorem 5.10.12.

The Max-flow, Min-cut Theorem can also be generalized to channeled flows as follows:

Theorem 5.10.13 *For any network, $N = (G, b, c, v_s, v_t)$, if a flow exists in N, then the maximum value, $|f|$, of any flow, f, in N is equal to the minimum capacity, $c(\Omega(Y)) = c(\Omega^+(Y)) - b(\Omega^-(Y))$, of any v_s-v_t-cocycle in N (this means that $v_s \in Y$ and $v_r \notin Y$).*

If the capacity functions b and c have the property that $b(e) < 0$ and $c(e) > 0$ for all $e \in E$, then the condition of Theorem 5.10.12 is trivially satisfied. Furthermore, in this case, the flow $f = 0$ is admissible, Proposition 5.10.6 holds and we can apply directly the construction of the residual network, N_f, described above.

A variation of our last problem appears in Cormen, Leiserson, Rivest and Stein [11] (Chapter 26): In this version, the underlying graph, G, of the network, N, is assumed to have no parallel edges (and no loops), so that every edge, e, can be identified with the pair, (u, v), of its endpoints (so, $E \subseteq V \times V$). A flow, f, in N is a function, $f\colon V \times V \to \mathbb{R}$, where is not necessarily the case that $f(u, v) \geq 0$ for all (u, v), but there is a capacity function, $c\colon V \times V \to \mathbb{R}$, such that $c(u, v) \geq 0$, for all $(u, v) \in V \times V$ and it is required that

$$\begin{aligned} f(v, u) &= -f(u, v) \quad \text{and} \\ f(u, v) &\leq c(u, v), \end{aligned}$$

for all $(u, v) \in V \times V$. Moreover, in view of the skew symmetry condition ($f(v, u) = -f(u, v)$), the equations of conservation of flow are written as

$$\sum_{(u,v) \in E} f(u, v) = 0,$$

for all $u \neq v_s, v_t$.

We can reduce this last version of the flow problem to our previous setting by noticing that in view of skew symmetry, the capacity conditions are equivalent to having capacity functions, b', and c', defined such that

$$\begin{aligned} b'(u, v) &= -c(v, u) \\ c'(u, v) &= c(u, v), \end{aligned}$$

for every $(u, v) \in E$ and f must satisfy

$$b'(u, v) \leq f(u, v) \leq c'(u, v)$$

for all $(u, v) \in E$. However, we must also have $f(v, u) = -f(u, v)$, which is an additional constraint in case G has both edges (u, v) and (v, u). This point may be a little confusing since in our previous setting, $f(u, v)$ and $f(v, u)$ are independent values. However, this new problem is solved essentially as the previous one. The construction of the residual graph is identical to the previous case and so is the flow augmentation procedure along an elementary path, *except that* we force $f_\pi(v, u) = f_\pi(u, v)$ to hold during this step. For details, the reader is referred to Cormen, Leiserson, Rivest and Stein [11], Chapter 26.

More could be said about flow problems but we believe that we have covered the basics satisfactorily and we refer the reader to the various references mentioned in this section for more on this topic.

5.11 Matchings, Coverings, Bipartite Graphs

In this section, we will be dealing with finite unoriented graphs. Consider the following problem: We have a set of m machines, $M_1, \ldots, M_m$, and n tasks, $T_1, \ldots, T_n$. Furthermore,

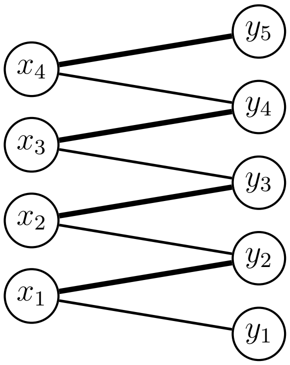

Figure 5.34: A bipartite graph, G, and a maximum matching in G

.

each machine, M_i, is capable of performing a subset of tasks, $S_i \subseteq \{T_1, \ldots, T_n\}$. Then, the problem is to find a set of assignments, $\{(M_{i_1}, T_{i_1}), \ldots, (M_{j_p}, T_{j_p})\}$, with $\{i_1, \ldots, i_p\} \subseteq \{1, \ldots, m\}$ and $\{j_1, \ldots, j_p\} \subseteq \{1, \ldots, n\}$, such that

(1) $T_{j_k} \in S_{i_k}, \quad 1 \leq k \leq p$;

(2) p is maximum.

The problem we just described is called a *maximum matching problem*. A convenient way to describe this problem is to build a graph, G (undirected), with $m + n$ nodes partitioned into two subsets X and Y, with $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$, and with an edge between x_i and y_j iff $T_j \in S_i$, that is, if machine M_i can perform task T_j. Such a graph, G, is called a *bipartite graph*. An example of a bipartite graph is shown in Figure 5.34. Now, our matching problem is to find an edge set of maximum size, M, such that no two edges share a common endpoint or, equivalently, such that every node belongs to at most one edge of M. Such a set of edges is called a *maximum matching* in G. A maximum matching whose edges are shown as thicker lines is shown in Figure 5.34.

Definition 5.11.1 A graph, $G = (V, E, st)$, is a *bipartite graph* iff its set of edges, V, can be partitioned into two nonempty disjoint sets, V_1, V_2, so that for every edge, $e \in E$, $|st(e) \cap V_1| = |st(e) \cap V_2| = 1$, i.e., one endpoint of e belongs to V_1 while the other belongs to V_2.

Note that in a bipartite graph, there are no edges linking nodes in V_1 (or nodes in V_2). Thus, there are no loops.

Remark: The *complete bipartite graph* for which $|V_1| = m$ and $|V_2| = n$ is the bipartite graph that has all edges (i, j), with $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$. This graph is denoted

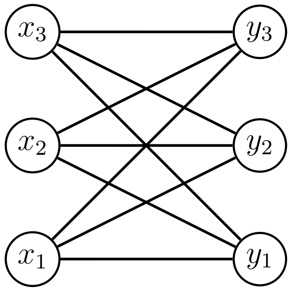

Figure 5.35: The bipartite graph $K_{3,3}$

.

$K_{m,n}$. The complete bipartite graph $K_{3,3}$ plays a special role, namely, it is not a planar graph, which means that it is impossible to draw it on a plane without avoiding that two edges (drawn as continuous simple curves) intersect. A picture of $K_{3,3}$ is shown in Figure 5.35.

The maximum matching problem in a bipartite graph can be nicely solved using the methods of Section 5.10 for finding Max-flows. Indeed, our matching problem is equivalent to finding a maximum flow in the network, N, constructed from the bipartite graph G as follows:

1. Add a new source, v_s and a new sink, v_t;
2. Add an oriented edge, (v_s, u), for every $u \in V_1$;
3. Add an oriented edge, (v, v_t), for every $v \in V_2$;
4. Orient every edge, $e \in E$, from V_1 to V_2;
5. Define the capacity function, c, so that $c(e) = 1$, for every edge of this new graph.

The network corresponding to the bipartite graph of Figure 5.34 is shown inFigure 5.36.

Now, it is very easy to check that there is a matching, M, containing p edges iff there is a flow of value p. Thus, there is a one-to-one correspondence between maximum matchings and maximum integral flows. As we know that the algorithm *maxflow* (actually, its various versions) produces an integral solution when ran on the zero flow, this solution yields a maximum matching.

The notion of graph coloring is also important and has bearing on the notion of bipartite graph.

Definition 5.11.2 Given a graph, $G = (V, E, st)$, a *k-coloring* of G is a partition of V into k pairwise disjoint nonempty subsets, $V_1, \ldots, V_k$, so that no two vertices in any subset V_i are adjacent (i.e., the endpoints of every edge, $e \in E$, must belong to V_i and V_j, for some

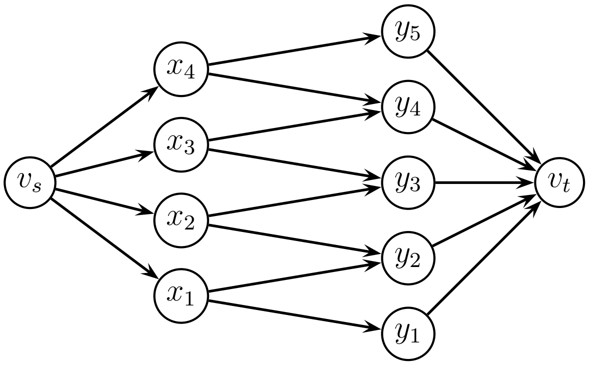

Figure 5.36: The network associated with a bipartite graph

.

$i \neq j$). If a graph, G, admits a k-coloring, we say that that G *is k-colorable*. The *chromatic number*, $\chi(G)$, of a graph, G, is the minimum k for which G is k-colorable.

Remark: Although the notation, $\chi(G)$, for the chromatic number of a graph is often used in the graph theory literature, it is an unfortunate choice because it can be confused with the Euler characteristic of a graph (see Theorem 5.12.8). Other notations for the chromatic number include $\gamma(G)$, $\nu(G)$ and $\mathrm{chr}(G)$.

The following theorem gives some useful characterizations of bipartite graphs. First, we must define the incidence matrix of an unoriented graph, G. Assume that G has edges $\mathbf{e}_1, \ldots, \mathbf{e}_n$ and vertices $\mathbf{v}_1, \ldots, \mathbf{v}_m$. The *incidence matrix, A, of* G, is the $m \times n$ matrix whose entries are given by

$$a_{i\,j} = \begin{cases} 1 & \text{if } \mathbf{v}_i \in st(\mathbf{e}_j) \\ 0 & \text{otherwise.} \end{cases}$$

Note that, unlike the incidence matrix of a directed graph, the incidence matrix of an undirected graph only has non-negative entries. As a consequence, these matrices are not necessarily totally unimodular. For example, the reader should check that for any elementary cycle, C, of odd length, the incidence matrix, A, of C has a determinant whose value is ± 2. However, the next theorem will show that the incidence matrix of a bipartite graph is totally unimodular and in fact, this property characterizes bipartite graphs.

In order to prove part of the next theorem we need the notion of distance in a graph, an important concept in any case. If G is a connected graph, for any two nodes u and v of G, the length of a chain, π, from u to v is the number of edges in π and the *distance*, $d(u, v)$, from u to v is the minimum length of all path from u to v. Of course, $u = v$ iff $d(u, v) = 0$.

Theorem 5.11.3 *Given any graph, $G = (V, E, st)$, the following properties are equivalent:*

(1) G is bipartite.

(2) $\gamma(G) = 2$.

(3) G has no elementary cycle of odd length.

(4) G has no cycle of odd length.

(5) The incidence matrix of G is totally unimodular.

Proof. The equivalence (1) $\Longleftrightarrow$ (2) is clear by definition of the chromatic number.

(3) $\Longleftrightarrow$ (4) holds because every cycle is the concatenation of elementary cycles. So, a cycle of odd length must contain some elementary cycle of odd length.

(1) $\Longrightarrow$ (4). This is because the vertices of a cycle belong alternatively to V_1 and V_2. So, there must be an even number of them.

(4) $\Longrightarrow$ (2). Clearly, a graph is k-colorable iff all its connected components are k-colorable, so we may assume that G is connected. Pick any node, v_0, in G and let V_1 be the subset of nodes whose distance from v_0 is even and V_2 be the subset of nodes whose distance from v_0 is odd. We claim that any two nodes, u and v, in V_1 (resp. V_2) are not adjacent. Otherwise, by going up the chains from u and v back to v_0 and by adding the edge from u to v, we would obtain a cycle of odd length, a contradiction. Therefore, G, is 2-colorable.

(1) $\Longrightarrow$ (5). Orient the edges of G so that for every $e \in E$, $s(e) \in V_1$ and $t(e) \in V_2$. Then, we know from Proposition 5.8.5 that the incidence matrix, D, of the oriented graph G is totally unimodular. However, because G is bipartite, D is obtained from A by multiplying all the rows corresponding to nodes in V_2 by -1 and so, A is also totally unimodular.

(5) $\Longrightarrow$ (3). Let us prove the contrapositive. If G has an elementary cycle, C, of odd length, then we observed that the submatrix of A corresponding to C has determinant ± 2. □

We now define the general notion of a matching.

Definition 5.11.4 Given a graph, $G = (V, E, st)$, a *matching, M, in G* is a subset of edges so that any two distinct edges in M have no common endpoint (are not adjacent) or equivalently, so that every vertex, $v \in E$, is incident to at most one edge in M. A vertex, $v \in V$ is *matched* iff it is incident some some edge in M and otherwise it is said to be *unmatched*. A matching, M, is a *perfect matching* iff every node is matched.

An example of a perfect matching, $M = \{(ab), (cd), (ef)\}$, is shown in Figure 5.37, with the edges of the matching indicated in thicker lines. The pair $\{(bc), (ed)\}$ is also a matching, in fact, a maximal matching (no edge can be added to this matching and still have a matching).

It is possible to characterize maximum matchings in terms of certain types of chains called *alternating chains* defined below:

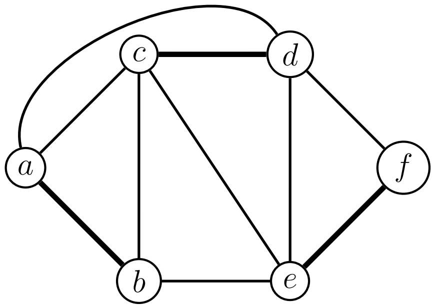

Figure 5.37: A perfect matching in a graph

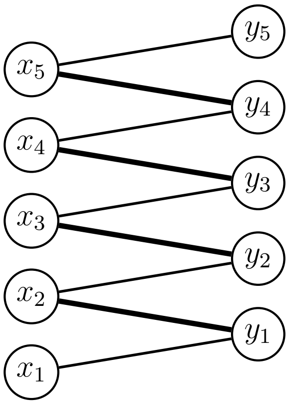

Figure 5.38: An alternating chain in G

Definition 5.11.5 Given a graph, $G = (V, E, st)$, and a matching, M, in G, an elementary chain is an *alternating chain w.r.t* M iff the edges in this chain belong alternately to M and $E - M$.

Theorem 5.11.6 *(Berge) Given any graph, $G = (V, E, st)$, a matching, M, in G is a maximum matching iff there are no alternating chains w.r.t. M whose endpoints are unmatched.*

Proof. First, assume that M is a maximum matching and that C is an alternating chain w.r.t. M whose enpoints, u and v are unmatched. An an example, consider the alternating chain shown in Figure 5.38, where the edges in $C \cap M$ are indicated in thicker lines.

We can form the set of edges

$$M' = (M - (C \cap M)) \cup (C \cap (E - M)),$$

which consists in deleting the edges in M from C and adding the edges from C not in M. It is immediately verified that M' is still a matching but $|M'| = |M| + 1$ (see Figure 5.38), contradicting the fact that M is a maximum matching. Therefore, there are no alternating chains w.r.t. M whose endpoints are unmatched.

Conversely, assume that G has no alternating chains w.r.t. M whose endpoints are unmatched and let M' be another matching with $|M'| > |M|$ (i.e., M is not a maximum matching). Consider the spanning subgraph, H, of G, whose set of edges is

$$(M - M') \cup (M' - M).$$

As M and M' are matchings, the connected components of H are either isolated vertices, or elementary cycles of even length, or elementary chains, and in these last two cases, the edges in these cycles or chains belong alternately to M and M'; this is because $d_H(u) \leq 2$ for every vertex $u \in V$ and if $d_H(u) = 2$, then u is adjacent to one edge in M and one edge in M'.

Now, H must possess a connected component that is a chain, C, whose enpoints are in M', as otherwise we would have $|M'| \leq |M|$, contradicting the assumption $|M'| > |M|$. However, C is an alternating chain w.r.t. M whose endpoints are unmatched, a contradiction. □

A notion closely related to the concept of a matching but, in some sense, dual, is the notion of a *line cover*.

Definition 5.11.7 Given any graph, $G = (V, E, st)$, without loops or isolated vertices, a *line cover* (or *line covering*) of G is a set of edges, $\mathcal{C} \subseteq E$, so that every vertex $u \in V$ is incident to some edge in $\mathcal{C}$. A *minimum line cover*, $\mathcal{C}$, is a line cover of minimum size.

The maximum matching, M, in the graph of Figure 5.37 is also a minimum line cover. The set $\{(ab), (bc), (de), (ef)\}$ is also a line cover but it is not minimum, although minimal. The relationship between maximum matchings and minimum covers is given by the following theorem:

Theorem 5.11.8 *Given any graph, $G = (V, E, st)$, without loops or isolated vertices, with $|V| = n$, let M be a maximum matching and let $\mathcal{C}$ be a minimum line cover. Then, the following properties hold:*

(1) If we associate to every unmatched vertex of V some edge incident to this vertex and add all such edges to M, then we obtain a minimum line cover, $\mathcal{C}_M$.

(2) Every maximum matching, M', of the spanning subgraph, $(V, \mathcal{C})$, is a maximum matching of G.

(3) $|M| + |\mathcal{C}| = n$.

Proof. It is clear that $\mathcal{C}_M$ is a line cover. As the number of vertices umatched by M is $n - 2|M|$ (as each edge in M matches exactly two vertices), we have

$$|\mathcal{C}_M| = |M| + n - 2|M| = n - |M|. \qquad (*)$$

Furthermore, as $\mathcal{C}$ is a minimum line cover, the spanning subgraph, $(V, \mathcal{C})$, does not contain any cycle or chain of length greater than or equal to 2. Consequently, each edge $e \in \mathcal{C} - M'$ corresponds to a single vertex unmatched by M'. Thus,

$$|\mathcal{C}| - |M'| = n - 2|M'|,$$

that is

$$|\mathcal{C}| = n - |M'|. \tag{$**$}$$

As M is a maximum matching of G,

$$|M'| \leq |M|$$

and so, using $(*)$ and $(**)$, we get

$$|\mathcal{C}_M| = n - |M| \leq n - |M'| = |\mathcal{C}|,$$

that is, $|\mathcal{C}_M| \leq |\mathcal{C}|$. However, $\mathcal{C}$ is a minimum matching, so $|\mathcal{C}| \leq |\mathcal{C}_M|$, which proves that

$$|\mathcal{C}| = |\mathcal{C}_M|.$$

The last equation proves the remaining claims. □

There are also notions analogous to matchings and line covers but applying to vertices instead of edges.

Definition 5.11.9 Let $G = (V, E, st)$ be any graph. A set, $U \subseteq V$, of nodes is *independent* (or *stable*) iff no two nodes in U are adjacent (there is no edge having these nodes as endpoints). A *maximum independent set* is an independent set of maximum size. A set, $\mathcal{U} \subseteq V$, of nodes is a *point cover* (or *vertex cover* or *transversal*) iff every edge of E is incident to some node in $\mathcal{U}$. A *minimum point cover* is a point cover of minimum size.

For example, $\{a, b, c, d, f\}$ is point cover of the graph of Figure 5.37. The following simple proposition holds:

Proposition 5.11.10 *Let $G = (V, E, st)$ be any graph, U be any independent set, $\mathcal{C}$ be any line cover, $\mathcal{U}$ be any point cover and M be any matching. Then, we have the following inequalities:*

(1) $|U| \leq |\mathcal{C}|$;

(2) $|M| \leq |\mathcal{U}|$

(3) U is an independent set of nodes iff $V - U$ is a point cover.

```
procedure marking($G, M, mark$)
  begin
    for each $u \in V_1 \cup V_2$ do $mark(u) := 0$ endfor;
    while $\exists u \in V_1 \cup V_2$ with $mark(u) = 0$ and $u$ not matched by $M$ do
      $mark(u) := +$;
      while $\exists v \in V_1 \cup V_2$ with $mark(v) = 0$ and $v$ adjacent to $w$ with $mark(w) = +$ do
        $mark(v) := -$;
        if $v$ is not matched by $M$ then exit $(\alpha)$
        $(*$ an alternating chain has been found $*)$
        else find $w \in V_1 \cup V_2$ so that $(vw) \in M$; $mark(w) := +$
        endif
      endwhile
    endwhile;
    for each $u \in V_1$ with $mark(u) = 0$ do $mark(u) := +$ endfor;
    for each $u \in V_2$ with $mark(u) = 0$ do $mark(u) := -$ endfor $(\beta)$
  end
```

Figure 5.39: Procedure *marking*

Proof. (1) Since U is an independent set of nodes, every edge in $\mathcal{C}$ is incident with at most one vertex in U, so $|U| \leq |\mathcal{C}|$.

(2) Since M is a matching, every vertex in $\mathcal{U}$ is incident to at most one edge in M, so $|M| \leq |\mathcal{U}|$.

(3) Clear from the definitions. □

It should be noted that the inequalities of Proposition 5.11.10 can be strict. For example, if G is an elementary cycle with $2k+1$ edges, the reader should check that both inequalities are strict.

We now go back to bipartite graphs and give an algorithm which, given a bipartite graph, $G = (V_1 \cup V_2, E)$, will decide whether a matching, M, is a maximum matching in G. This algorithm, shown in Figure 5.39, will mark the nodes with the one of the three tags, $+$, $-$, or 0.

The following theorem tells us what is the behavior of the procedure *marking*.

Theorem 5.11.11 *The procedure marking always terminates in one of the following two (mutually exclusive) situations:*

(a) The algorithm finds an alternating chain w.r.t. M whose endpoints are unmatched.

(b) The algorithm finds a point cover, $\mathcal{U}$, with $|\mathcal{U}| = |M|$, which shows that M is a maximum matching.

Proof. Since nodes keep being marked, the algorithm obviously terminates. There are no pairs of adjacent nodes bothy marked $+$ since, as soon as a node is marked $+$, all of its adjacent nodes are labeled $-$. Consequently, if the algorithm ends in (β), those nodes marked $-$ form a point cover.

We also claim that the endpoints, u and v, of any edge in the matching can't both be marked $-$. Otherwise, by following backward the chains that allowed the marking of u and v, we would find an odd cycle, which is impossible in a bipartite graph. Thus, if we end in (β), each node marked $-$ is incident to exactly one edge in M. This shows that the set, $\mathcal{U}$, of nodes marked $-$ is a point cover with $|\mathcal{U}| = |M|$. By Proposition 5.11.10, we see that $\mathcal{U}$ is a minimum point cover and that M is a maximum matching.

If the algorithm ends in (α), by tracing the chain starting from the unmatched node, u, marked $-$ back to the node marked $+$ causing u to marked, and so on, we find an alternating chain w.r.t. M whose endpoints are not matched. $\square$

The following important corollaries follow immediately from Theorem 5.11.11:

Corollary 5.11.12 *In a bipartite graph, the size of a minimum point cover is equal to the size of maximum matching.*

Corollary 5.11.13 *In a bipartite graph, the size of a maximum independent set is equal to the size of a minimum line cover.*

Proof. We know from Proposition 5.11.10 that the complement of a point cover is an independent set. Consequently, by Corollary 5.11.12, the size of a maximum independent set is $n - |M|$, where M is a maximum matching and n is the number of vertices in G. Now, from Theorem 5.11.8 (3), for any maximum matching, M, and any minimal line cover, $\mathcal{C}$, we have $|M| + |\mathcal{C}| = n$ and so, the size of a maximum independent set is equal to the size of a minimal line cover. $\square$

We can derive more classical theorems from the above results.

Given any graph, $G = (V, E, st)$, for any subset of nodes, $U \subseteq V$, let

$$N_G(U) = \{v \in V - U \mid (\exists u \in U)(\exists e \in E)(st(e) = \{u, v\})\},$$

be the set of *neighbours* of U, i.e., the set of vertices *not* in U and adjacent to vertices in U.

Theorem 5.11.14 *(König (1931)) For any bipartite graph, $G = (V_1 \cup V_2, E, st)$, the maximum size of a matching is given by*

$$\min_{U \subseteq V_1}(|V_1 - U| + |N_G(U)|).$$

Proof. This theorem will follow from Corollary 5.11.12 if we can show that every minimum point cover is of the form $(V_1 - U) \cup N_G(U)$, for some subset, U, of V_1. However, a moment of reflexion shows that this is indeed the case. $\square$

Theorem 5.11.14 implies another classical result:

Theorem 5.11.15 *(König-Hall) For any bipartite graph, $G = (V_1 \cup V_2, E, st)$, there is a matching, M, such that all nodes in V_1 are matched iff*

$$|N_G(U)| \geq |U| \quad \textit{for all} \quad U \subseteq V_1.$$

Proof. By Theorem 5.11.14, there is a matching, M in G with $|M| = |V_1|$ iff

$$|V_1| = \min_{U \subseteq V_1}(|V_1 - U| + |N_G(U)|) = \min_{U \subseteq V_1}(|V_1| + |N_G(U)| - |U|),$$

that is, iff $|N_G(U)| - |U| \geq 0$ for all $U \subseteq V_1$. □

Now, it is clear that a bipartite graph has a perfect matching (i.e., a matching such that every vertex is matched), M, iff $|V_1| = |V_2|$ and M matches all nodes in V_1. So, as a corollary of Theorem 5.11.15, we see that a bipartite graph has a perfect matching iff $|V_1| = |V_2|$ and if

$$|N_G(U)| \geq |U| \quad \text{for all} \quad U \subseteq V_1.$$

As an exercise, the reader should show the

Marriage Theorem (Hall, 1935) *Every k-regular bipartite graph, with $k \geq 1$, has a perfect matching* (a graph is k-regular iff every node has degree k).

For more on bipartite graphs, matchings, covers, *etc.*, the reader should consult Diestel [14] (Chapter 2), Berge [3] (Chapter 7) and also Harary [29] and Bollobas [7].

5.12 Planar Graphs

Suppose we have a graph, G, and that we want to draw it "nicely" on a piece of paper, which means that we draw the vertices as points and the edges as line segments joining some of these points, in such a way that *no two edges cross each other*, except possibly at common endpoints. We will have more flexibility and still have a nice picture if we allow each abstract edge to be represented by a continuous simple curve (a curve that has no self-intersection), that is, a subset of the plane homeomorphic to the closed interval $[0, 1]$ (in the case of a loop, a subset homeomorphic to the circle, S^1). If a graph can be drawn in such a fashion, it is called a *planar graph*. For example, consider the graph depicted in Figure 5.40.

If we look at Figure 5.40, we may believe that the graph G is not planar, but this is no so. In fact, by moving the vertices in the plane and by continuously deforming some of the edges, we can obtain a planar drawing of the same graph, as shown in Figure 5.41.

However, we should not be overly optimistic. Indeed, if we add an edge from node 5 to node 4, obtaining the graph known as K_5 shown in Figure 5.42, it can be proved that there is no way to move the nodes around and deform the edge continuously to obtain a planar graph (we will prove this a little later using the Euler formula). Another graph that is non-planar is the bipartite grapk $K_{3,3}$. The two graphs, K_5 and $K_{3,3}$ play a special role with respect to

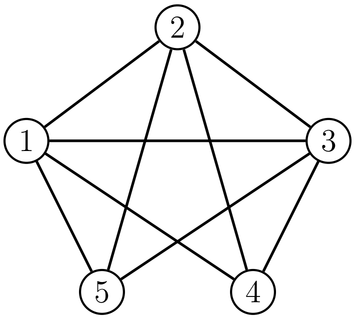

Figure 5.40: A Graph, G, drawn with intersecting edges

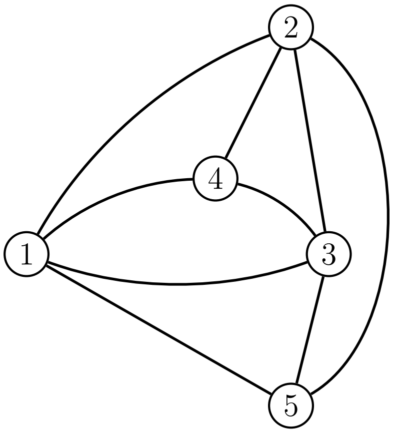

Figure 5.41: The Graph, G, drawn as a plane graph

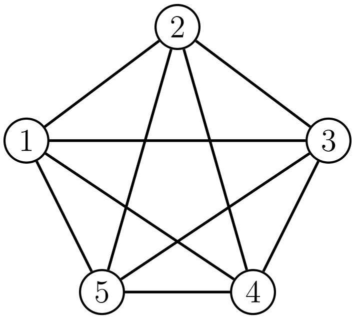

Figure 5.42: The complete graph K_5, a non-planar graph

planarity. Indeed, a famous theorem of Kuratowski says that a graph is planar if and only if it does not contain K_5 or $K_{3,3}$ as a minor (we will explain later what a minor is).

Remark: Given n vertices, say $\{1,\ldots,n\}$, the graph whose edges are all subsets $\{i,j\}$, with $i,j \in \{1,\ldots,n\}$ and $i \neq j$, is the *complete graph on n vertices* and is denoted by K_n (but Diestel uses the notation K^n).

In order to give a precise definition of a planar graph, let us review quickly some basic notions about curves. A *simple curve* (or *Jordan curve*) is any injective continuous function, $\gamma\colon [0,1] \to \mathbb{R}^2$. Since $[0,1]$ is compact and γ is continuous, it is well-known that the inverse, $f^{-1}\colon \gamma([0,1]) \to [0,1]$, of f is also continuous. So, γ is a homeomorphism between $[0,1]$ and its image, $\gamma([0,1])$. With a slight abuse of language we will also call the image, $\gamma([0,1])$, of γ, a simple curve. This image is a connected and compact subset of $\mathbb{R}^2$. The points $a = \gamma(0)$ and $b = \gamma(1)$ are called the *boundaries* or *endpoints* of γ (and $\gamma([0,1])$). The open subset $\gamma([0,1]) - \{\gamma(0), \gamma(1)\}$ is called the *interior* of $\gamma([0,1])$ and is denoted $\overset{\circ}{\gamma}$. A continuous function , $\gamma\colon [0,1] \to \mathbb{R}^2$, such that $\gamma(0) = \gamma(1)$ and γ is injective on $[0,1)$ is called a *simple closed curve* or *simple loop* or *closed Jordan curve*. Again, by abuse of language, we call the image, $\gamma([0,1])$, of γ, a simple closed curve *etc*. Equivalently, if $S^1 = \{(x,y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\}$ is the unit circle in $\mathbb{R}^2$, a simple closed curve is any subset of $\mathbb{R}^2$ homeomorphic to S^1. In this case, we call $\gamma(0) = \gamma(1)$ the *boundary* or *base point* of γ. The open subset $\gamma([0,1]) - \{\gamma(0)\}$ is called the *interior* of $\gamma([0,1])$ and is also denoted $\overset{\circ}{\gamma}$.

Remark: The notions of simple curve and simple closed curve also make sense if we replace $\mathbb{R}^2$ by any topological space, X, in particular, a surface (In this case, a simple (closed) curve is a continuous injective function $\gamma\colon [0,1] \to X$ *etc*.).

We can now define plane graphs as follows:

Definition 5.12.1 A *plane graph* is a pair, $\mathcal{G} = (V, E)$, where V is a finite set of points in $\mathbb{R}^2$, E is a finite set of simple curves and closed simple curves in $\mathbb{R}^2$ called *edges* and *loops*, respectively, and satisfying the following properties:

(i) The endpoints of every edge in E are vertices in V and the base point of every loop is a vertex in V.

(ii) The interior of every edge contains no vertex and the interiors of any two distinct edges are disjoint. Equivalently, every edge contains no vertex except for its boundaries (base point in the case of a loop) and any two distinct edges intersect only at common boundary points.

We say that G is a *simple plane graph* if it has no loops and if different edges have different sets of endpoints

Obviously, a plane graph, $\mathcal{G} = (V, E)$, defines an "abstract graph", $G = (V, E, st)$, such that

(a) For every simple curve, γ,
$$st(\gamma) = \{\gamma(0), \gamma(1)\}$$

(b) For every simple closed curve, γ,
$$st(\gamma) = \{\gamma(0)\}.$$

For simplicity of notation, we will usually write $\mathcal{G}$ for both the plane graph and the abstract graph associated with $\mathcal{G}$.

Definition 5.12.2 Given an abstract graph, G, we say that *G is a planar graph* iff there is some plane graph, $\mathcal{G}$, and an isomorphism, $\varphi\colon G \to \mathcal{G}$, between G and the abstract graph associated with $\mathcal{G}$. We call φ an *embedding of G in the plane* or a *planar embedding of G*.

Remarks:

1. If G is a *simple* planar graph, then by a theorem of Fary, G can be drawn as a plane graph in such a way that the edges are straight line segments (see Gross and Tucker [26], Section 1.6).

2. In view of the remark just before Definition 5.12.1, given any topological space, X, for instance, a surface, we can define a graph on X as a pair, (V, E), where V is a finite set of points in X and E is a finite sets of simple (closed) curves on X satisfying the conditions of Definition 5.12.1.

3. Recall the *stereographic projection (from the north pole)*, $\sigma_N\colon (S^2 - \{N\}) \to \mathbb{R}^2$, from the sphere, $S^2 = \{(x, y, x) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 = 1\}$ onto the equatorial plane, $z = 0$, with $N = (0, 0, 1)$ (the north pole), given by
$$\sigma_N(x, y, z) = \left(\frac{x}{1-z}, \frac{y}{1-z}\right).$$
We know that σ_N is a homeomorphism, so if φ is a planar embedding of a graph G into the plane, then $\sigma_N^{-1} \circ \varphi$ is an embedding of G into the sphere. Conversely, if ψ is an embedding of G into the sphere, then $\sigma_N \circ \psi$ is a planar embedding of G. Therefore, a graph can be embedded in the plane iff it can be embedded in the sphere. One of the nice features of embedding in the sphere is that the sphere is compact (closed and bounded), so the faces (see below) of a graph embedded in the sphere are all bounded.

4. The ability to embed a graph in a surface other that the sphere broadens the class of graphs that can be drawn without pairs of intersecting edges (except at endpoints). For example, it is possible to embed K_5 and $K_{3,3}$ (which are known *not* to be planar) into a torus (try it!). It can be shown that for every (finite) graph, G, there is some surface, X, such that G can be embedded in X. Intuitively, whenever two edges cross on a sphere, by lifting one of the two edges a little bit and adding a “handle” on which the lifted edge lies we can avoid the crossing. An excellent reference on the topic of graphs on surfaces is Gross and Tucker [26].

One of the new ingredients of plane graphs is that the notion of a face makes sense. Given any nonempty open subset, Ω, of the plane $\mathbb{R}^2$, we say that two points, $a, b \in \Omega$ are *(arcwise) connected*[4] iff there is a simple curve, γ, such that $\gamma(0) = a$ and $\gamma(1) = b$. Being connected is an equivalence relation and the equivalence classes of Ω w.r.t. connectivity are called the *connected components* (or *regions*) of Ω. Each region is maximally connected and open. If R is any region of Ω and if we denote the closure of R (i.e., the smallest closed set containing R) by $\overline{R}$, then the set $\partial R = \overline{R} - R$ is also a closed set called the *boundary* (or *frontier*) of R.

Now, given a plane graph, $\mathcal{G}$, if we let $|\mathcal{G}|$ be the the subset of $\mathbb{R}^2$ consisting of the union of all the vertices and edges of $\mathcal{G}$, then this is a closed set and it complement, $\Omega = \mathbb{R}^2 - |\mathcal{G}|$, is an open subset of $\mathbb{R}^2$.

Definition 5.12.3 Given any plane graph, $\mathcal{G}$, the regions of $\Omega = \mathbb{R}^2 - |\mathcal{G}|$ are called the *faces* of $\mathcal{G}$.

As expected, for every face, F, of $\mathcal{G}$, the boundary, ∂F, of F is the subset, $|\mathcal{H}|$, associated with some subgraph, $\mathcal{H}$, of $\mathcal{G}$. However, one should observe that the boundary of a face may be disconnected and may have several "holes". The reader should draw lots of planar graphs to understand this phenomenon. Also, since we are considering finite graphs, the set $|\mathcal{G}|$ is bounded and thus, every plane graph has exactly one unbounded face. Figure 5.43 shows a planar graph and its faces. Observe that there are five faces, where A is bounded by the entire graph, B is bounded by the triangle $(4, 5, 6)$ the outside face, C, is bounded by the two edges from 8 to 2, the loop around node 2, the two edges from 2 to 7 and the outer edge from 7 to 8, D is bounded by the two edges between 7 and 8, and E is bounded by the loop around node 2.

Remarks:

1. Using (inverse) stereographic projection, we see that all the faces of a graph embedded in the sphere are bounded.

2. If a graph, G, is embedded in a surface, S, then the notion of face still makes sense. Indeed, the faces of G are the regions of the open set $\Omega = S - |G|$.

Actually, one should be careful (as usual) not to rely too much on intuition when dealing with planar graphs. Although certain facts seem obvious, they may turn out to be false after closer scrutiny and when they are true, they may be quite hard to prove. One of the best examples of an "obvious" statement whose proof is much less trivial than one might expect is the Jordan curve theorem which is actually needed to justify certain "obvious" facts about faces of plane graphs.

[4] In topology, a space is connected iff it cannot be expressed as the union of two nonempty disjoint open subsets. For *open* subsets of $\mathbb{R}^n$, connectedness is equivalent to arc connectedness. So it is legitimate to use the term connected.

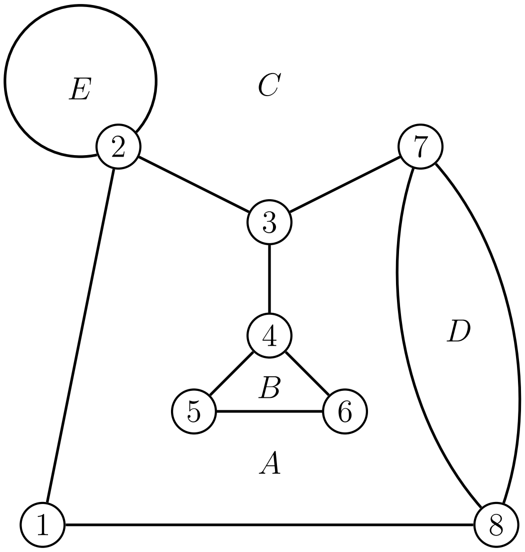

Figure 5.43: A planar graph and its faces

Theorem 5.12.4 *(Jordan Curve Theorem) Given any closed simple curve, γ, in $\mathbb{R}$, the complement, $\mathbb{R}^2 - \gamma([0,1])$, of $\gamma([0,1])$, consist of exactly two regions both having $\gamma([0,1])$ as boundary.*

Proof. There are several proofs all using machinery (such as homology or differential topology) beyond the scope of these notes. A proof using the notion of winding number is given in Guillemin and Pollack [27] (Chapter 2, Section 5) and another proof using homology can be found in Munkres [34] (Chapter 4, Section 36). □

Using Theorem 5.12.4, the following properties can be proved:

Proposition 5.12.5 *Let $\mathcal{G} = (V, E)$ be any plane graph and let $e \in E$ be any edge of $\mathcal{G}$. Then the following properties hold:*

(1) For any face, F of $\mathcal{G}$, either $e \subseteq \partial f$ or $\partial F \cap \overset{\circ}{e} = \emptyset$.

(2) If e lies on a cycle, C, of $\mathcal{G}$, then e lies on the boundary of exactly two faces of G and these are contained in distinct faces of C.

(3) If e lies on no cycle, then e lies on the boundary of exactly one face of $\mathcal{G}$.

Proof. See Diestel [14], Section 4.2.

As a corollaries, we also have

Proposition 5.12.6 *Let $\mathcal{G} = (V, E)$ be any plane graph and let F be any face of $\mathcal{G}$. Then, the boundary, ∂F, of F is a subgraph of $\mathcal{G}$ (more accurately, $\partial = |\mathcal{H}|$, for some subgraph, $\mathcal{H}$, of $\mathcal{G}$).*

Proposition 5.12.7 *Every plane forest has a single face.*

One of the main theorems about plane graphs is the so-called *Euler formula*.

Theorem 5.12.8 *Let G be any connected planar graph with n_0 vertices, n_1 edges and n_2 faces. Then, we have*

$$n_0 - n_1 + n_2 = 2.$$

Proof. We proceed by induction on n_1. If $n_1 = 0$, the formula is trivially true, as $n_0 = n_2 = 1$. Assume the theorem holds for any $n_1 < n$ and let G be a connected planar graph with n edges. If G has no cycle, then as it is connected, it is a tree, $n_0 = n + 1$ and $n_2 = 1$, so $n_0 - n_1 + n_2 = n + 1 - n + 1 = 2$, as desired. Otherwise, let e be some edge of G belonging to a cycle. Consider the graph $G' = (V, E - \{e\})$, it is still a connected planar graph. Therefore, by the induction hypothesis,

$$n_0 - (n_1 - 1) + n_2' = 2.$$

However, by Proposition 5.12.5, as e lies on exactly two faces of G, we deduce that $n_2 = n_2' + 1$. Consequently

$$2 = n_0 - (n_1 - 1) + n_2' = n_0 - n_1 + 1 + n_2 - 1 = n_0 - n_1 + n_2,$$

establishing the induction hypothesis. □

Remarks:

1. Euler's formula was already known to Descartes in 1640 but the first proof by given by Euler in 1752. Poincaré generalized it to higher-dimensional polytopes.

2. The numbers n_0, n_1, and n_2 are often denoted by n_v, n_e and n_f (v for *vertex*, e for *edge* and f for *face*).

3. The quantity $n_0 - n_1 + n_2$ is called the *Euler characteristic* of the graph G and it is usually denoted by χ_G.

4. If a connected graph, G, is embedded in a surface (orientable), S, then we still have an Euler formula of the form

   $$n_0 - n_1 + n_2 = \chi(X) = 2 - 2g,$$

   where $\chi(S)$ is a number depending only on the surface, S, called the *Euler characteristic* of the surface and g is called the *genus* of the surface. It turns out that $g \geq 0$ is the number of "handles" that need to be glued to the surface of a sphere to get a homeomorphic copy of the surface S. For on this fascinating subject, see Gross and Tucker [26].

It is really remarkable that the quantity $n_0 - n_1 + n_2$ is independent of the way a planar graph is drawn on a sphere (or in the plane). A neat application of Euler's formula is the proof that there are only five regular convex polyhedra (the so-called *platonic solids*). Such a proof can be found in many places, for instance Berger [4] and Cromwell [12]. It is easy to generalize Euler's formula to planar graphs that are not necessarily connected.

Theorem 5.12.9 *Let G be any planar graph with n_0 vertices, n_1 edges, n_2 faces and c connected components. Then, we have*

$$n_0 - n_1 + n_2 = c + 1.$$

Proof. Reduce the proof of Theorem 5.12.9 to the proof of Theorem 5.12.8 by adding vertices and edges between connected components to make G connected. Details are left as an exercise. □

Using the Euler formula we can now prove rigorously that K_5 and $K_{3,3}$ are not planar graphs. For this, we will need the following fact:

Proposition 5.12.10 *If G is any simple, connected, plane graph with $n_1 \geq 3$ edges and n_2 faces, then*

$$2n_1 \geq 3n_2.$$

Proof. Let $F(G)$ be the set of faces of G. Since G is connected, by Proposition 5.12.5 (2), every edge belongs to exactly two faces. Thus, if s_F is the number of sides of a face, F, of G, we have

$$\sum_{F \in F(G)} s_F = 2n_1.$$

Furthermore, as G has no loops, no parallel edges and $n_0 \geq 3$, every face has at least three sides, i.e., $s_F \geq 3$. It follows that

$$2n_1 = \sum_{F \in F(G)} s_F \geq 3n_2,$$

as claimed. □

The proof of Proposition 5.12.10 shows that the crucial constant on the right-hand the inequality is the the minimum length of all cycles in G. This number is called the *girth* of the graph G. The girth of a graph with a loop is 1 and the girth of a graph with parallel edges is 2. The girth of a tree is undefined (or infinite). Therefore, we actually proved:

Proposition 5.12.11 *If G is any connected, plane graph with n_1 edges and n_2 faces and G is not a tree, then*

$$2n_1 \geq \mathrm{girth}(G)\, n_2.$$

Corollary 5.12.12 *If G is any simple, connected, plane graph with $n \geq 3$ nodes then G has at most $3n-6$ edges and $2n-4$ faces.*

Proof. By Proposition 5.12.10, we have $2n_1 \geq 3n_2$, where n_1 is the number of edges and n_2 is the number of faces. So, $n_2 \leq \frac{2}{3}n_1$ and by Euler's formula

$$n - n_1 + n_2 = 2,$$

we get

$$n - n_1 + \frac{2}{3}n_1 \geq 2,$$

that is,

$$n - \frac{1}{3}n_1 \geq 2,$$

namely $n_1 \leq 3n-6$. Using $n_2 \leq \frac{2}{3}n_1$, we get $n_2 \leq 2n-4$. □

Corollary 5.12.13 *The graphs K_5 and $K_{3,3}$ are not planar.*

Proof. We proceed by contradiction. First, consider K_5. We have $n_0 = 5$ and K_5 has $n_1 = 10$ edges. On the other hand, by Corollary 5.12.12, K_5 should have at most $3 \times 5 - 6 = 15 - 6 = 9$ edges, which is absurd.

Next, consider $K_{3,3}$. We have $n_0 = 6$ and $K_{3,3}$ has $n_1 = 9$ edges. By the Euler formula, we should have

$$n_2 = 9 - 6 + 2 = 5.$$

Now, as $K_{3,3}$ is bipartite, it does not contain any cycle of odd length, and so each face has at least *four* sides, which implies that

$$2n_1 \geq 4n_2$$

(because the girth of $K_{3,3}$ is 4.) So, we should have

$$18 = 2 \cdot 9 \geq 4 \cdot 5 = 20,$$

which is absurd. □

Another important property of simple planar graph is the following:

Proposition 5.12.14 *If G is any simple, planar graph, then there is a vertex, u, such that $d_G(u) \leq 5$.*

Proof. If the property holds for any connected component of G, then it holds for G, so we may assume that G is connected. We already know from Proposition 5.12.10 that $2n_1 \geq 3n_2$. i.e.

$$n_2 \leq \frac{2}{3}n_1. \qquad (*)$$

If $d_G(u) \geq 6$ for every vertex, u, as $\sum x_{u \in V} d_G(u) = 2n_1$, then $6n_0 \leq 2n_1$, i.e., $n_0 \leq n_1/3$. By Euler's formula, we would have

$$n_2 = n_1 - n_0 + 2 \geq n_1 - \frac{1}{3}n_1 + 2 > \frac{2}{3}n_1,$$

contradicting $(*)$. □

Remarkably, Proposition 5.12.14 is the key ingredient in the proof that every planar graph is 5-colorable.

Theorem 5.12.15 *Every planar graph, G, is* 5*-colorable.*

Proof. Clearly, parallel edges and loop play no role in finding a coloring of the vertices of G, so we may assume that G is a simple graph. Also, the property is clear for graphs with less than 5 vertices. We will proceed by induction on the number of vertices, m. By Proposition 5.12.14, the graph G has some vertex, u_0, with $d_G(u) \leq 5$. By the induction hypothesis, we can color the subgraph, G', induced by $V - \{u_0\}$ with 5 colors. If $d(u_0) < 5$, we can color u_0 with one of the colors not used to color the nodes adjacent to u_0 (at most 4) and we are done. So, assume $d_G(u_0) = 5$ and let $v_1, \ldots, v_5$ be the nodes adjacent to u_0 and encountered in this order when we rotate counter-clockwise around u_0 (see Figure 5.44). If $v_1, \ldots, v_5$ are not colored with different colors, again, we are done.

Otherwise, by the induction hypothesis, let $\{X_1, \ldots, X_5\}$ be a coloring of G' and, by renaming the X_i's if necessary, assume that $v_i \in X_i$, for $i = 1, \ldots, 5$. There are two cases:

(1) There is no chain from v_1 to v_3 whose nodes belong alternately to X_1 and X_2. If so, v_1 and v_3 must belong to different connected components of the subgraph, H', of G' induced by $X_1 \cup X_2$. Then, we can permute the colors 1 and 3 in the connected component of H' that contains v_3 and color u_0 with color 3.

(2) There is a chain from v_1 to v_3 whose nodes belong alternately to X_1 and X_2. In this case, as G is a planar graph, there can't be any chain from v_2 to v_4 whose nodes belong alternately to X_2 and X_4. So, v_2 and v_4 do not belong to the same connected component of the subgraph, H'', of G' induced by $X_2 \cup X_4$. But then, we can permute the colors 2 and 4 in the connected component of H'' that contains v_4 and color u_0 with color 4. □

Theorem 5.12.15 raises a very famous problem known as the *four color problem*: Can every planar graph be colored with four colors?

This question was apparently first raised by Francis Guthrie in 1850, communicated to De Morgan by Guthrie's brother Frederick in 1852 and brought to the attention to a wider public by Cayley in 1878. In the next hunded years, several incorrect proofs were proposed and this problem became know as the *four color conjecture*. Finally, in 1977, Appel and Haken gave the first "proof" of the four color conjecture. However, this proof was somewhat

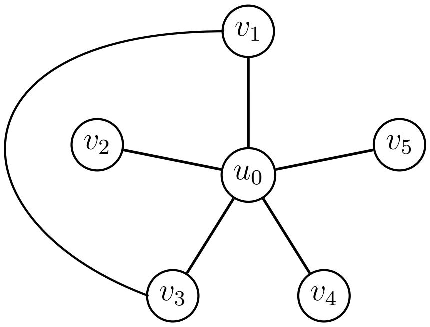

Figure 5.44: The 5 nodes adjacent to u_0

controversial for various reasons, one of the reasons being that it relies on a computer program for checking a large number of unavoidable configurations. Appel and Haken subsequently published a 741 page paper correcting a number of errors and addressing various criticisms. More recently (1997) a much shorter proof, still relying on a computer program, but a lot easier to check (including the computer part of it) has been given by Robertson, Sanders, Seymour and Thomas. For more on the four color problem, see Diestel [14], Chapter 5, and the references given there.

let us now go back to Kuratowski's criterion for non-planarity. For this, it is useful to introduce the notion of edge contraction in a graph.

Definition 5.12.16 Let $G = (V, E, st)$ be any graph and let e be any edge of G. The graph obtained by *contracting the edge e into a new vertex,* v_e, is the graph, $G/e = (V', E', st')$, with $V' = (V - st(e)) \cup \{v_e\}$ where v_e is a new node ($v_e \notin V$); $E' = E - \{e\}$; and with

$$st'(e') = \begin{cases} st(e') & \text{if } st(e') \cap st(e) = \emptyset \\ \{v_e\} & \text{if } st(e') = st(e) \\ \{u, v_e\} & \text{if } st(e') \cap st(e) = \{z\} \text{ and } st(e') = \{u, z\} \text{ with } u \neq z \\ \{v_e\} & \text{if } st(e') = \{x\} \text{ or } st(e') = \{y\} \text{ with } st(e) = \{x, y\}. \end{cases}$$

If G is a simple graph, then we need to eliminate parallel edges and loops. In, this case, $e = \{x, y\}$ and $G/e = (V', E', st)$ is defined so that $V' = (V - \{x, y\}) \cup \{v_e\}$ where v_e is a new node and

$$\begin{aligned} E' = {} & \{\{u, v\} \mid \{u, v\} \cap \{x, y\} = \emptyset\} \\ & \cup \{\{u, v_e\} \mid \{u, x\} \in E - \{e\} \quad \text{or} \quad \{u, y\} \in E - \{e\}\}. \end{aligned}$$

Figure 5.45 shows the result of contracting the upper edge $\{2, 4\}$ (shown as a thicker line) in the graph shown on the left, which is not a simple graph. Observe how the lower edge $\{2, 4\}$ becomes a loop around 7 and the two edges $\{5, 2\}$ and $\{5, 4\}$ become parallel edges between 5 and 7.

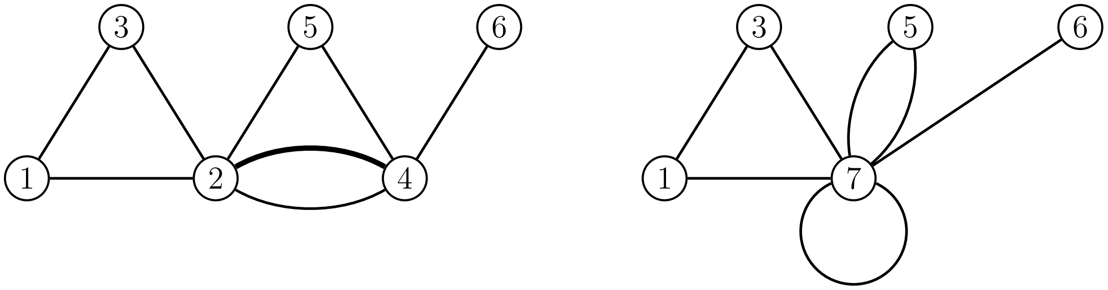

Figure 5.45: Edge Contraction in a graph

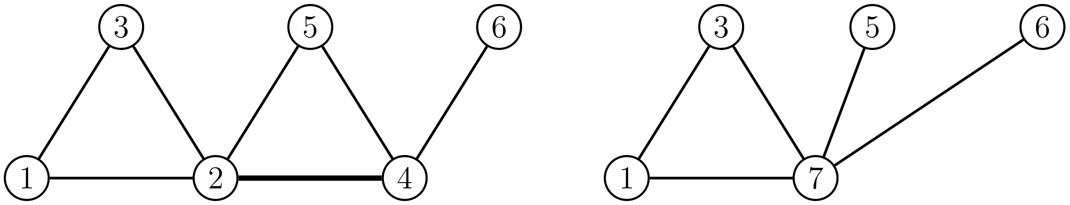

Figure 5.46: Edge Contraction in a simple graph

Figure 5.46 shows the result of contracting edge $\{2, 4\}$ (shown as a thicker line) in the simple graph shown on the left. This time, the two edges $\{5, 2\}$ and $\{5, 4\}$ become a single edge and there is no loop around 7 as the contracted edge is deleted.

Now, given a graph, G, we can repeatedly contract edges. We can also take a subgraph of a graph G and then perform some edge contractions. We obtain what is known as a minor of G.

Definition 5.12.17 Given any graph, G, a graph, H, is a *minor* of G is there is a sequence of graphs, $H_0, H_1, \ldots, H_n$ $(n \geq 1)$, such that

(1) $H_0 = G$; $H_n = H$;

(2) Either H_{i+1} is obtained from H_i by deleting some edge or some node of H_i and all the edges incident with this node, or

(3) H_{i+1} is obtained from H_i by edge contraction,

with $0 \leq i \leq n - 1$. If G is a simple graph, we require that edge contractions be of the second type described in Definition 5.12.16, so that H is a a simple graph.

It is easily shown that the minor relation is a partial order on graphs (and simple graphs). Now, the following remarkable theorem originally due to Kuratowski characterizes planarity in terms of the notion of minor:

Theorem 5.12.18 *(Kuratowski, 1930) For any graph, G, the following assertions are equivalent:*

(1) G is planar;

(2) G contains neither K_5 nor $K_{3,3}$ as a minor.

Proof. The proof is quite involved. The first step is to prove the theorem for 3-connected graphs. (A graph $G = (V, E)$ is k-connected iff $|V| > k$ and iff every graph obtained by deleting any set, $S \subseteq V$, of nodes with $|S| < k$ and the edges incident to these node is still connected. So, a 1-connected graph is just a connected graph.) We refer the reader to Diestel [14], Section 4.4, for a complete proof. $\square$

Another way to state Kuratowski's theorem involves edge subdivision, an operation of independent interest. Given a graph, $G = (V, E, st)$, possibly with loops and parallel edges, the result of subdividing an edge, e, consists in creating a new vertex, v_e, deleting the edge e, and adding two new edges from v_e to the old endpoints of e (possibly the same point). Formally, we have the following definition:

Definition 5.12.19 Given any graph, $G = (V, E, st)$, for any edge, $e \in E$, the result of *subdividing the edge* e is the graph, $G' = (V \cup \{v_e\}, (E - \{e\}) \cup \{e^1, e^2\}, st')$, where v_e is a new vertex and e^1, e^2 are new edges, $st'(e') = st(e')$ for all $e' \in E - \{e\}$ and if $st(e) = \{u, v\}$ ($u = v$ is possible), then $st'(e^1) = \{v_e, u\}$ and $st'(e^2) = \{v_e, v\}$. If a graph, G', is obtained from a graph, G, by a sequence of edge subdivisions, we say that G' is a *subdivision* of G.

Observe that by repeatedly subdividing edges, any graph can be transformed into a simple graph. Given two graphs, G and H, we say that G and H are *homeomorphic* iff they have respective subdivisions G' and H' that are isomorphic graphs. The idea is that homeomorphic graphs "look the same", viewed as topological spaces. Figure 5.47 shows an example of two homeomorphic graphs. A graph, H, that has a subdivision, H', which is a subgraph of some graph, G, is called a *topological minor* of G. Then, it is not hard to show (see Diestel [14], Chapter 4, or Gross and Tucker [26], Chapter 1) that Kuratowski's Theorem is equivalent to the statement

A graph, G, is planar iff it does not contain any subgraph homeomorphic to either K_5 or $K_{3,3}$ or, equivalently, if it has has neither K_5 nor $K_{3,3}$ as a topological minor.

Another somewhat surprising characterization of planarity involving the concept of cycle space over $\mathbb{F}_2$ (see Definition 5.7.10 and the Remarks after Theorem 5.7.16) and due to MacLane is the following:

Theorem 5.12.20 *(MacLane, 1937) A graph, G is planar iff its cycle space, $\mathcal{F}$, over $\mathbb{F}_2$ has a basis such that every edge of G belongs to at most two cycles of this basis.*

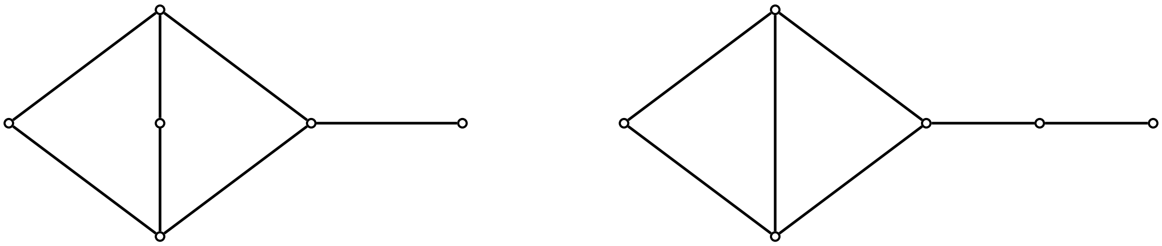

Figure 5.47: Two homeomorphic graphs

Proof. See Diestel [14], Section 4.4. □

We conclude this section on planarity with a brief discussion of the dual graph of a plane graph, a notion originally due to Poincaré. Duality can be generalized to simplicial complexes and relates Voronoi diagrams and Delaunay triangulations, two very important tools in computational geometry.

Given a plane graph, $G = (V, E)$, let $F(G)$ be the set of faces of G. The crucial point is that every edge of G is part of the boundary of at most two faces. A dual graph, $G^* = (V^*, E^*)$, of G is a graph whose nodes are in one-to-one correspondence with the faces of G, whose faces are in one-to-one correspondence with the nodes of G and whose edges are also are in one-to-one correspondence with the the egdes of G. For any edge, $e \in E$, a dual edge, e^*, links the two nodes v_{F_1} and v_{F_2} associated with the faces F_1 and F_2 adjacent to e or, e^* is a loop from v_F to itself if e is ajacent to a single face. Here is the precise definition:

Definition 5.12.21 Let $G = (V, E)$ be a plane graph and let $F(G)$ be its set of faces. A *dual graph* of G is a graph, $G^* = (V^*, E^*)$, where

(1) $V^* = \{v_F \mid F \in F(G)\}$, where v_F is a point chosen in the (open) face, F, of G;

(2) $E^* = \{e^* \mid e \in E\}$, where e^* is a simple curve from v_{F_1} to v_{F_2} crossing e, if e is part of the boundary of two faces F_1 and F_2 or else, a closed simple curve crossing e from v_F to itself, if e is part of the boundary of exactly one face, F.

(3) For each $e \in E$, we have $e^* \cap G = e \cap G^* = \overset{\circ}{e} \cap \overset{\circ}{e^*}$, a one point set.

An example of a dual graph is shown in Figure 5.48. The graph G has four faces, a, b, c, d and the dual graph, G^*, has nodes also denoted a, b, c, d enclosed in a small circle, with the edges of the dual graph shown with thicker lines.

Note how the edge $\{5, 6\}$ gives rise to the loop from d to itself and that there are parallel edges between d and a and between d and c. Thus, even if we start with a simple graph, a dual graph may have loops and parallel edges.

Actually, it is not entirely obvious that a dual of a plane graph is a plane graph but this is not difficult to prove. It is also important to note that a given plane graph, G, *does*

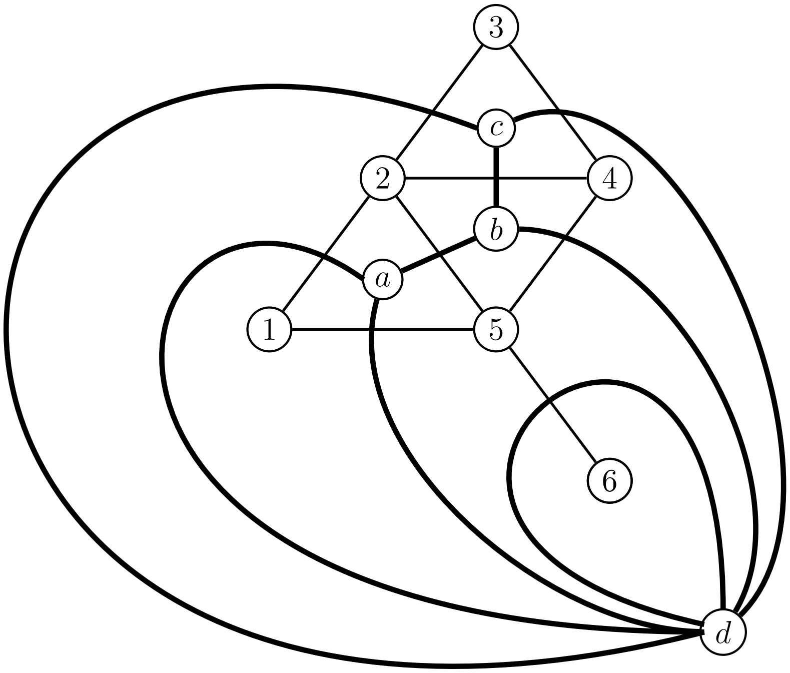

Figure 5.48: A graph and its dual graph

not have a unique dual since the vertices and the edges of a dual graph can be chosen in infinitely different ways in order to satisfy the conditions of Definition 5.12.21. However, given a plane graph, G, if H_1 and H_2 are two dual graphs of G, then it is easy to see that H_1 and H_2 are isomorphic. Therefore, with a slight abuse of language, we may refer to "the" dual graph of a plane graph. Also observe that even if G is not connected, its dual, G^*, is always connected.

The notion of dual graph applies to a *plane* graph and *not to a planar graph*. Indeed, the graphs G_1^* and G_2^* associated to two different embeddings, G_1 and G_2, of the same abstract planar graph, G, may **not** be isomorphic, even though G_1 and G_2 are isomorphic as abstact graphs. For example, the two plane graphs, G_1 and G_2, shown in Figure 5.49 are isomorphic but their dual graphs, G_1^* and G_2^*, are not, as the reader should check (one of these two graphs has a node of degree 7 but for the other graph all nodes have degree at most 6).

Remark: If a graph, G, is embedded in a surface, S, then the notion of dual graph also makes sense. More for on this, see Gross and Tucker [26].

In the following proposition, we summarize some useful properties of dual graphs.

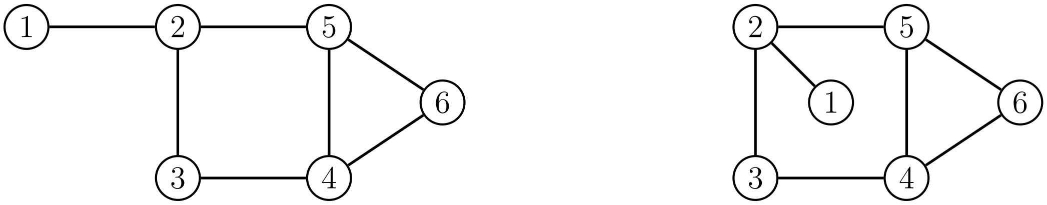

Figure 5.49: Two isomorphic plane graphs whose dual graphs are not isomorphic

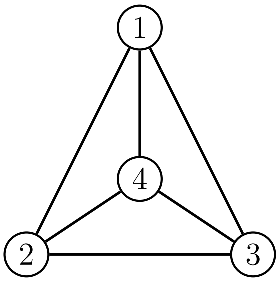

Figure 5.50: A self-dual graph

Proposition 5.12.22 *The dual, G^* of any plane graph is connected. Furthermore, if G is a connected plane graph, then G^{**} is isomorphic to G.*

Proof. Left as an exercise.

We a slight abuse of notation we often write $G^{**} = G$ (when G is connected). A plane graph, G, whose dual, G^*, is equal to G (i.e., isomorphic to G) is called *self-dual*. For example, the plane graph shown in Figure 5.50 (the projection of a tetrahedron on the plane) is self dual.

The duality of plane graphs is also reflected algebraically as a duality between their cycle spaces and their cut spaces (over $\mathbb{F}_2$).

Proposition 5.12.23 *If G is any connected plane graph, G, then the following properties hold:*

(1) A set of edges, $C \subseteq E$, is a cycle in G iff $C^* = \{e^* \in E^* \mid e \in C\}$ is a minimal cutset in G^*.

(2) If $\mathcal{F}(G)$ and $\mathcal{T}(G^*)$ denote the cycle space of G over $\mathbb{F}_2$ and the cut space of G^* over $\mathbb{F}_2$, respectively, then the dual, $\mathcal{F}^*(G)$, of $\mathcal{F}(G)$ (as a vector space) is equal to the cut space, $\mathcal{T}(G^*)$, of G^*, i.e.,

$$\mathcal{F}^*(G) = \mathcal{T}(G^*).$$

(3) If T is any spanning tree of G, then $(V^*, (E - E(T))^*)$ is a spanning tree of G^* (Here, $E(T)$ is the set of edges of the tree, T.)

Proof. See Diestel [14], Section 4.6. □

The interesting problem of finding an algorithmic test for planarity has received quite a bit of attention. Hopcroft and Tarjan have given an algorithm running in linear time in the number of vertices. More more on planarity, the reader should consult Diestel [14], Chapter 4, or Harary [29], Chapter 11.

Besides the four color "conjecture", the other most famous theorem of graph theory is the *graph minor theorem*, due to Roberston and Seymour and we can't resist stating this beautiful and amazing result. For this, we need to explain what is a *well-quasi order*, for short, a *w.q.o*.

Recall that a partial order on a set, X, is a binary relation, $\leq$, which is reflexive, symmetric and anti-symmetric. A *quasi-order* (or *preorder*) is a relation which is reflexive and transitive (but not necessarily anti-symmetric). A *well-quasi-order*, for short, a *w.q.o* , is a quasi-order with the following property:

For every infinite sequence, $(x_n)_{n\geq 1}$, of elements $x_i \in X$, there exist some indices, i, j, with $1 \leq i < j$, so that $x_i \leq x_j$.

Now, we know that being a minor of another graph is a partial order and thus, a quasi-order. Here is Robertson and Seymour's theorem:

Theorem 5.12.24 *(Graph Minor Theorem, Robertson and Seymour, 1985-2004) The minor relation on finite graphs is a well quasi-order.*

Remarkably, the proof of Theorem 5.12.24 is spread over 20 Journal papers (under the common title, *Graph Minors*) written over nearly 18 years and taking well over 500 pages! Many original techniques had to be invented to come up with this proof, one of which is a careful study of the conditions under which a graph can be embedded in a surface and a "Kuratowski-type" criterion based on a finite family of "forbidden graphs". The interested reader is urged to consult Chapter 12 of Diestel [14] and the references given there.

A precursor of the graph minor theorem is a theorem of Kruskal (1960) which applies to trees. Although much easier to prove that the graph minor theorem, the proof fo Kruskal's Theorem is very ingenious. It turns out that there are also some interesting connections between Kruskal's Theorem and proof theory, due to Harvey Friedman. A survey on this topic can be found in Gallier [16].

Bibliography

[1] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To truth Through Proof.* Academic Press, 1986.

[2] Claude Berge. *Principles of Combinatorics.* Academic Press, first edition, 1971.

[3] Claude Berge. *Graphs and Hypergraphs.* Elsevier North-Holland, first edition, 1973.

[4] Marcel Berger. *Géométrie 1.* Nathan, 1990. English edition: Geometry 1, Universitext, Springer Verlag.

[5] Norman Biggs. *Algebraic Graph Theory*, volume 67 of *Cambridge Tracts in Mathematics.* Cambridge University Press, first edition, 1974.

[6] Garrett Birkhoff. *Lattice Theory.* Colloquium Publications, Vol. XXV. AMS, third edition, 1973.

[7] Béla Bollobas. *Modern Graph Theory.* GTM No. 184. Springer Verlag, first edition, 1998.

[8] J. Cameron, Peter. *Combinatorics: Topics, Techniques, Algorithms.* Cambridge University Press, first edition, 1994.

[9] Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics.* AMS, first edition, 1997.

[10] John H. Conway and K. Guy, Richard. *The Book of Numbers.* Copernicus, Springer-Verlag, first edition, 1996.

[11] H. Cormen, Thomas, E. Leiserson, Charles, L. Rivest, Ronald, and Clifford Stein. *Introduction to Algorithms.* MIT Press, second edition, 2001.

[12] Peter Cromwell. *Polyhedra.* Cambridge University Press, first edition, 1994.

[13] H.B. Curry and R. Feys. *Combinatory Logic, Vol. I.* Studies in Logic. North-Holland, third edition, 1974.

[14] Reinhard Diestel. *Graph Theory.* GTM No. 173. Springer Verlag, third edition, 2005.

[15] Herbert B. Enderton. *Elements of Set Theory.* Academic Press, first edition, 1977.

[16] Jean Gallier. What's so Special about Kruskal's Theorem and the Ordinal Γ_0? *Annals of Pure and Applied Logic*, 53:199–260, 1991.

[17] Jean Gallier. Constructive Logics. Part I: A Tutorial on Proof Systems and Typed λ-Calculi. *Theoretical Computer Science*, 110(2):249–339, 1993.

[18] Jean Gallier. On the Correspondence Between Proofs and λ-Terms. In Philippe de Groote, editor, *Cahiers Du Centre de Logique, Vol. 8*, pages 55–138. Academia, Louvain-La-Neuve, 1995.

[19] Jean H. Gallier. *Logic for Computer Science.* Harper and Row, New York, 1986.

[20] Jean H. Gallier. *Geometric Methods and Applications, For Computer Science and Engineering.* TAM, Vol. 38. Springer, first edition, 2000.

[21] G. Gentzen. Investigations into logical deduction. In M.E. Szabo, editor, *The Collected Papers of Gerhard Gentzen.* North-Holland, 1969.

[22] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1989.

[23] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[24] Chris Godsil and Gordon Royle. *Algebraic Graph Theory.* GTM No. 207. Springer Verlag, first edition, 2001.

[25] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation For Computer Science.* Addsion Wesley, second edition, 1994.

[26] L. Gross, Jonathan and W. Tucker, Thomas. *Topological Graph Theory.* Dover, first edition, 2001.

[27] Victor Guillemin and Alan Pollack. *Differential Topology.* Prentice Hall, first edition, 1974.

[28] Paul R. Halmos. *Naive Set Theory.* Undergraduate Text in Mathematics. Springer Verlag, first edition, 1974.

[29] Frank Harary. *Graph Theory.* Addison Wesley, first edition, 1971.

[30] W. A. Howard. The formulae-as-types notion of construction. In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic Press, London, 1980. Reprint of manuscript first published in 1969.

[31] Michael Huth and Mark Ryan. *Logic in Computer Science. Modelling and reasonning about systems.* Cambridge University Press, Cambridge, United Kingdom, 2000.

[32] S. Kleene. *Introduction to Metamathematics.* North-Holland, seventh edition, 1952.

[33] Jon Kleinberg and Eva Tardos. *Algorithm Design.* Addison Wesley, first edition, 2006.

[34] James R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, first edition, 1984.

[35] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization. Algorithms and Complexity.* Dover, first edition, 1998.

[36] D. Prawitz. *Natural deduction, a proof-theoretical study.* Almquist & Wiksell, Stockholm, 1965.

[37] D. Prawitz. Ideas and results in proof theory. In J.E. Fenstad, editor, *Proc. 2nd Scand. Log. Symp.*, pages 235–307. North-Holland, 1971.

[38] Michel Sakarovitch. *Optimisation Combinatoire, Méthodes mathématiques et algorithmiques. Graphes et Programmation Linéaire.* Hermann, first edition, 1984.

[39] Michel Sakarovitch. *Optimisation Combinatoire, Méthodes mathématiques et algorithmiques. Programmation Discréte.* Hermann, first edition, 1984.

[40] Richard P. Stanley. *Enumerative Combinatorics, Vol. I.* Cambridge Studies in Advanced Mathematics, No. 49. Cambridge University Press, first edition, 1997.

[41] Patrick Suppes. *Axiomatic Set Theory.* Dover, first edition, 1972.

[42] G. Takeuti. *Proof Theory*, volume 81 of *Studies in Logic.* North-Holland, 1975.

[43] A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*, volume 43 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1996.

[44] D. van Dalen. *Logic and Structure.* Universitext. Springer Verlag, second edition, 1980.

[45] Herbert S. Wilf. *Algorithms and Complexity.* A K Peters, LTD, second edition, 2002.