1
König's Lemma

1.1 Two ways of looking at mathematics

It seems that in mathematics there are sometimes two or more ways of proving the same result. This is often mysterious, and seems to go against the grain, for we often have a deep-down feeling that if we choose the 'right' ideas or definitions, there must be only one 'correct' proof. This feeling that there should be just one way of looking at something is rather similar to Paul Erdős's idea of 'The Book' [1], a vast tome held by God, the SF, in which all the best, most revealing and perfect proofs are written.

Sometimes this mystery can be resolved by analysing the apparently different proofs into their fundamental ideas. It often turns out that, 'underneath the bonnet', there is actually just one key mathematical concept, and two seemingly different arguments are in some sense 'the same'. But sometimes there really are two different approaches to a problem. This should not be disturbing, but should instead be seen as a great opportunity. After all, two approaches to the same idea indicates that there are some new mathematics to be investigated and some new connections to be found and exploited, which hopefully will uncover a wealth of new results.

I shall give a rather simple example of just the sort of situation I have in mind that will be familiar to many readers – one which will be typical of the kind of theorem we will be considering throughout this book.

Consider a binary *tree*. A tree is a diagram (often called a *graph*) with a special *point* or *node* called the *root*, and *lines* or *edges* leaving this node downwards to other nodes. These again may have edges leading to further nodes. The thing that makes this a tree (rather than a more general kind of graph) is that the edges all go downwards from the root, and that means the tree cannot have any *loops* or *cycles*. The tree is a *binary* tree if every node is connected to *at most* two lower nodes. If every node is connected to *exactly*

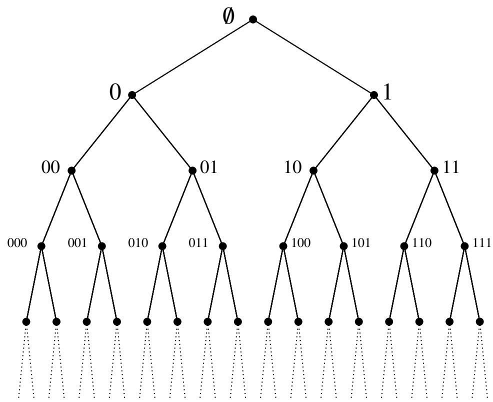

Figure 1.1 The full binary tree.

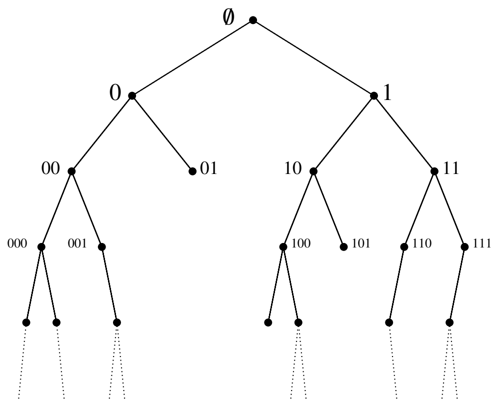

Figure 1.2 A binary tree.

two lower nodes, the tree is called the *full binary tree*. Note that in general, a node in a binary tree may be connected to 0, 1 or 2 lower nodes. We will label the nodes in our trees with sequences of integers. It is convenient to make labels for the nodes below the node that has label x by adding either the digit 0 or 1 to the end of x, giving $x0$ and $x1$. Figure 1.1 illustrates the full binary tree, whereas Figure 1.2 gives a typical (non-full) binary tree.

Trees are very important in mathematics, because many constructions follow trees in some way or other. Binary trees are especially interesting since a *walk* along a tree, following a path that starts at the root, has at most two choices of direction at every node. Binary trees arise quite naturally in many mathematical ideas and proofs and general theorems about them can be quite powerful and useful. One of the better known and more useful of these results is called König's Lemma.

To explain König's Lemma, consider what it means for a tree T to be *infinite*. There are two viewpoints, and two possible definitions.

Firstly, suppose you have somehow drawn the whole of the tree T on paper or on the blackboard and are inspecting it. You are in a fortunate position to be able to take in every one of its features, and to examine every one of its nodes and edges. You will quite naturally say that the tree is infinite if it has infinitely many nodes, or – amounting to the same thing – infinitely many edges. This is a sort of 'definition from perfect information' and is similar to what logicians call semantics, though we will not see the connection with semantics and the theory of 'meaning' for a while.

Now consider you are an ant walking on the binary tree T, which is again drawn in its entirety on paper. You start at the root node, and you follow the edges, like ant tracks, which you hope will take you to something interesting. Unlike the mathematician viewing the tree in its entirety, you can only see the node you are at and the edges leaving it. If you take a walk down the tree, you may have choices of turning left or right at any given node and continuing your path. But it is possible that you have no choice at all, because either there is only one edge out of the node other than the one you entered it by, or possibly there is no such edge at all, in which case your walk has come to an end. To the ant, which cannot perceive the whole of the tree, but just follows paths, there is a quite different idea of what it means for the tree to be infinite: the ant would say that T is infinite if it can find somehow (by guessing the right combination of 'left' and 'right' choices) an infinite path through the tree. The ant's definition of 'infinite' might be thought of as a 'definition from imperfect information' and is similar to the logician's idea of *proof*. If you like, you can think of an infinite path chosen by the ant as a *proof* that the tree is infinite. Like all proofs, it supports the claim made, without giving much extra information – such as what the tree looks like off this path.

König's Lemma is the statement that, for binary trees, these two ideas of a tree being infinite are the same. It is in fact a rather useful statement with many interesting applications. The key feature of this statement is that it relates two definitions, one mathematical definition working from perfect or total information, and one working from the point of view of much more limited information, and shows that they actually say the same thing.

As with all 'if and only if' theorems, there are two directions that must be proved. The first, that if there is an infinite path through the tree then the tree is infinite, is immediate. This easier direction is called a *Soundness Theorem* since it says the ant's perception based on partial information is *sound*, or in other words will not result in erroneous conclusions. The other direction is the non-trivial one, and its mathematical strength lies in the way it states that a rather general mathematical situation (that the tree is infinite) can always be detected in a special way from partial information. The reason why it is called *Completeness* will be discussed later in relation to some other examples.

This has been a long preliminary discussion, but I hope it has proved illuminating. We shall now turn to the more formal mathematical details and define *tree*, *path*, etc., and then state and prove König's Lemma properly.

Definition 1.1 The set of *natural numbers*, $\mathbb{N}$, will be taken in this book to be $\{0, 1, 2, \ldots\}$.

For those readers who expect the natural numbers to start with 1, I can only say that I appreciate that there are occasions when it is convenient to forget about zero, but for me zero is very natural, probably the most logically natural number of all, so is included here in the set of natural numbers.

Definition 1.2 A *sequence* is a function s whose domain is either the set $\mathbb{N}$ of all natural numbers or a subset of it of the form $\{x \in \mathbb{N} : x < n\}$ for some $n \in \mathbb{N}$. Normally the values of the sequence will be numbers, 0 or 1 say, but the definition above (with $n = 0$) allows the empty sequence with no values at all. We write a sequence by listing its values in order, for example as 00110101001 or 0101010101. The *length* of a sequence is the number of elements in the domain of the function. This will always be a natural number or infinity.

Definition 1.3 If s is a sequence of length l and $n \in \mathbb{N}$ is at most l, then $s \restriction n$ denotes the initial part of s of length n.

For example, if $s = 00100011$ then $s \restriction 4 = 0010$.

Definition 1.4 If s is a sequence of length l and x is 0 or 1 then sx is the sequence of length $l + 1$ whose last element is x and all other elements agree with those of s.

Our definition of a tree is of a set of sequences that is closed under the restriction operation $\restriction$.

Definition 1.5 A *tree* is a set of sequences T such that for any $s \in T$ of length n and for any $l < n$ then $s \restriction l \in T$.

Think of a sequence $s \in T$ as a finite path starting from the root and arriving at some node. The individual digits in the sequence determine which choice of edge is made at each node. The set of nodes of the whole tree is then the set of sequences in the set T and two nodes $s, t \in T$ are connected by a single edge if one can be got from the other by adding a single number to the sequence. In other words, s and t are *connected* if $s \restriction (n-1) = t$ when s is the longer of the two and has length n, or the other way round if t is longer. Then the condition in the definition says, not unreasonably, that each node that this path passes through must also be in the tree. The root of the tree is the empty sequence of length 0.

Definition 1.6 A *subtree* of a tree T is a subset S of T that is a tree in its own right.

A subtree of a tree T might contain fewer nodes, and therefore fewer choices at certain nodes.

Definition 1.7 A *binary tree* is a tree T where all the sequences in it are functions from some $\{n \in \mathbb{N} : n < k\}$ to $\{0, 1\}$.

In other words, at any node, a path from the root of a binary tree has at most two options: to go left (0) or right (1). However, it may turn out that only one, or possibly neither, of these options is available at a particular node.

Definition 1.8 A tree T is *infinite* if it contains infinitely many sequences, or (equivalently) has infinitely many nodes.

A path is a subtree with no branching allowed. That means for any two nodes in the tree, one is a 'predecessor' of the other. More formally, we have the following definition.

Definition 1.9 A *path*, p, in a tree T is a subtree of T such that for any $s, t \in p$ with lengths n, k respectively and $n \leqslant k$, we have $s = t \restriction n$.

A tree T containing an infinite path p is obviously infinite. König's Lemma states that the converse is also true for binary trees.

Theorem 1.10 (König's Lemma) *Let T be an infinite binary tree. Then T contains an infinite path p.*

Proof Suppose T is an infinite binary tree. For a sequence s of length n let T_s be $\{r \in T : r \upharpoonright n = s\} \cup \{s \upharpoonright k : k < n\}$, which we will call the *subtree of T below s*. You will be able to check easily that T_s is a tree. In general it may or may not be infinite.

We are going to find a sequence $s(n)$ of elements of T such that

- $s(n)$ has length n,
- $s(n) = s(n+1) \upharpoonright n$,
- the tree $T_{s(n)}$ below $s(n)$ is infinite.

This construction is by induction, using the third property above as our induction hypothesis. When we have completed the proof the set $\{s(n) : n \in \mathbb{N}\}$ will be our infinite path p in T.

So suppose inductively that we have chosen $s = s(n)$ of length n and T_s is infinite. Then since the tree is binary, made from sequences of 0s and 1s, we have

$$T_s = \{r \in T : r \upharpoonright (n+1) = s0\} \cup \{r \in T : r \upharpoonright (n+1) = s1\} \cup \{s \upharpoonright k : k \leqslant n\}.$$

This is, by the induction hypothesis, infinite. Hence (as the third of these three sets is obviously finite) at least one of the first two sets, corresponding to '0' or '1' respectively, is infinite. If the first of these is infinite we set $s(n+1) = s0$ and in this case we have

$$T_{s(n+1)} = \{r \in T : r \upharpoonright (n+1) = s0\} \cup \{s0\} \cup \{s \upharpoonright k : k \leqslant n\}$$

which is infinite. If not we set $s(n+1) = s1$ which would then be infinite as before. Either way we have defined $s(n+1)$ and proved the induction hypothesis for $n+1$. □

1.2 Examples and exercises

The central theorem of this book, the Completeness Theorem for first-order logic, is not only of the same flavour as König's Lemma, but is in fact a powerful generalisation of it. To give you an idea of the power that this sort of theorem has, it is useful to see a selection of applications of König's Lemma here.

We start by exploring the limits of König's Lemma a little: it turns out that the important thing is not that there are at most two choices at each node but that the number of ways in which the branches divide is always finite.

Definition 1.11 If T is a tree and $s \in T$ is a node of T then the *valency* or *degree* of s is the number of nodes of T connected to s. Thus this is the number of x such that $sx \in T$ plus one (to cater for the edge back towards the root), or just the number of such x if s is the root node.

Exercise 1.12 Prove the following generalisation of König's Lemma: an infinite tree in which every vertex has finite valency has an infinite path. Assume that the tree has vertices or nodes which are sequences of *natural numbers* of finite length and that for each $s \in T$ there is a bound $B_s \in \mathbb{N}$ on the possible values x such that $sx \in T$.

There are two ways that you might have done the last exercise. You might have modified the proof given above, or you may have tried to reduce the case of arbitrary finite valency trees to the case of binary trees by somehow 'encoding' arbitrary finite branching by a series of binary branches.

Exercise 1.13 Whichever method you used, have a go at proving the extension of König's Lemma by the other method.

Exercise 1.14 By giving an appropriate example of an infinite tree, show that König's Lemma is false for graphs with vertices of infinite valency.

König's Lemma is an elegant but nevertheless not very surprising or difficult result to see. Its truth, it seems, is reasonably clear, though a completely rigorous proof takes a moment or two to come up with. It is all the more surprising, therefore that there should be non-trivial applications. We will look at a few of these now, though nothing later in this book will depend on them.

Example 1.15 The set $X = [0, 1]$ has the property (called *sequential compactness*, equivalent to compactness for metric spaces) that every sequence (a_n) of elements of X has a subsequence converging to some element in X.

Proof Starting with $[0, 1]$ we continually divide intervals into equal halves, but at stage k of the construction we throw away any such interval that contains none of the a_n with $n > k$. More precisely, the nodes of the tree at depth k are identified with intervals $I = [(r-1)2^{-k}, r2^{-k}]$ for which $r \in \mathbb{N}$ and $\{a_n : n > k \text{ and } a_n \in I\}$ is non-empty, and two nodes are connected if one is a subset of the other.

This defines a binary tree. It is infinite because there are infinitely many a_n and each lies in an interval. By König's Lemma there is an infinite path through this tree, and by the construction of the tree we may take an infinite subsequence of a_n in this path, one at each level of the tree. This is the required convergent subsequence. □

Now consider infinite sequences $u_0u_1u_2\ldots$ of the digits $0, 1, 2, \ldots, k-1$. We will call such sequences *k-sequences*. Say a k-sequence s is *x^n-free* if there is no finite sequence, x, of digits $0, 1, 2, \ldots, k-1$, such that the finite sequence x^n (defined to be the result of repeating and concatenating x as $xxxx\ldots x$, where there are n copies of the string x) does not appear as a contiguous block of the sequence s.

Exercise 1.16 (a) Show that there is no x^2-free 2-sequence.

(b) Use König's Lemma to show that there is an x^3-free 2-sequence if and only if there are arbitrarily long finite x^3-free 2-sequences. State and prove a similar result for x^2-free 3-sequences.

(c) Define an operation on finite 2-sequences σ such that $\sigma(0) = 01$, $\sigma(1) = 10$, and $\sigma(u_0u_1\ldots u_m) = \sigma(u_0)\sigma(u_1)\ldots\sigma(u_m)$, where this is concatenation of sequences. Let $\sigma^n(s) = \sigma(\sigma(\ldots(\sigma(s))\ldots))$, i.e. σ iterated n times. Show that each of the finite sequences $\sigma^n(0)$ is x^3-free, and hence there is an infinite x^3-free 2-sequence.

(d) Show there is an x^2-free 3-sequence.

Another example of the use of König's Lemma is for graphs in the plane. A *graph* is a set V of vertices and a set E of edges, which are unordered subsets of V with exactly two vertices in each edge. In a *planar graph* the set of vertices V is a set of points of $\mathbb{R}^2$, and the edges joining vertices are lines which are 'smooth' (formed from finitely many straight-line segments) and may not cross except at a vertex.

A graph with set of vertices V can be *k-coloured* if there is a map $f\colon V \to \{0, 1, \ldots, k-1\}$ such that $f(u) \neq f(v)$ for all vertices u, v that are joined by an edge. You should think of the values $0, 1, \ldots, k-1$ as 'colours' of the vertices; the condition says two adjacent vertices must be coloured differently. Graph colourings, and especially colourings of planar graphs, are particularly interesting and have a long history [12]. A deep and difficult result by Appel and Haken shows that every finite planar graph is 4-colourable [10].

Exercise 1.17 Use König's Lemma to show that an infinite graph can be k-coloured if and only if every finite subgraph of it can be so coloured. (Make the simplification that the vertices of our infinite graph can be ordered as $v_0, v_1, \ldots$ with indices from $\mathbb{N}$. Construct a tree where the nodes at level n are all k-colourings of the subgraph with vertices $v_0, v_1, \ldots, v_{n-1}$, and edges join nodes if one colouring extends another.) Deduce from Appel and Haken's result that every infinite planar graph can be 4-coloured.

Tiling problems provide another nice application of König's Lemma. Consider a finite set of *tiles* which are square, with special links like jigsaw pieces so that in a tiling with tiles fitting together, one edge of one tile must be next to one of certain edges of other tiles. A tiling of the plane is a tiling using any number of tiles of each of the finitely many types, so that the whole of the plane is covered. *Tiling problems* ask whether the plane can or cannot be tiled using a particular set.

Exercise 1.18 Prove that a finite set of tiles can tile the plane if and only if every finite portion of the plane can be so tiled.

Finally, for this section, trees are also useful for describing computations. We will not define any idealised computer here, nor provide any background in computability theory, so this next example is for readers with such background, or who are willing to suspend judgement until they have such background. Normally, computations are deterministic, that is every step is determined completely by the state of the machine. A *non-deterministic computation* is one where the computer has a fixed number, B, of possible 'next moves' at any stage. The machine is allowed to choose one of these 'at random', or by making a 'lucky guess' and in so doing it hopes to verify that some assertion is true. This gives rise to a computation tree of all possible computations. Suppose we somehow know in advance that whatever choices are made at any step, every computation of the machine will eventually halt and give an answer. That means that all paths through the computation tree are finite. Then by the contrapositive of König's Lemma the tree is finite. This means that the non-deterministic computation can be simulated in finite time by a deterministic one which constructs the computation tree in memory and analyses it.

1.3 König's Lemma and reverse mathematics*

König's Lemma is rather attractive and has some pretty applications. It has been 'traditional' in logic textbooks to give some of the examples above as applications of the much more powerful 'Completeness Theorem for first-order logic'. Whilst not incorrect, this has always seemed a pity to me, as it hardly does the Completeness Theorem justice when the applications can be proved directly from the more familiar König's Lemma. Suffice it to say for now that there will be plenty of interesting applications of the full Completeness Theorem that cannot be argued from König's Lemma alone.

It may be a good idea to say a few words about why König's Lemma is powerful, and where it does real mathematical work. The reason is that, although there may be an infinite path in a tree, it is not always clear how to find one, and in any case there are likely to be choices involved. In our proof of König's Lemma, to keep track of all these individual choices, we used the concept of a certain subtree T_s being infinite. Being 'infinite' is of course a powerful mathematical property, and one about which there is a lot that can be said, both within and outside the field of mathematical logic. This concept of an infinite subtree is doing quite a lot of work for us here, especially as it is being used infinitely many times in the course of an induction.

Some workers in the logic community study these ideas in more detail by trying to identify which theorems need which lemmas to prove them. This area of logic is often called *reverse mathematics* since the main aim is usually to prove the *axioms* from the *theorems*. I am not going to advocate reverse mathematics here, but there are plenty of times when it is nice to know that a complicated lemma cannot be avoided in a proof. It is certainly true for many of the exercises in the previous section that König's Lemma (or something very much like it) is necessary for their solution. In reverse mathematics one usually works from a weaker set of axioms, one where the concept of an infinite set is not available. It turns out, for example, that relative to this weak set of axioms the sequential compactness of $[0, 1]$ is actually equivalent to König's Lemma. For more information on reverse mathematics see the publications by Harvey Friedman, Stephen Simpson and others, in particular Simpson's 2001 volume [11].

The proof of König's Lemma works, as we have seen, by making a series of choices. The issue of making choices is also a very subtle one, but one that will come up in many places in this book. We can always make finitely many choices as part of a proof, by just listing them. (In this way, to make n choices in a proof you will typically need at least n lines of proof, for each $n \in \mathbb{N}$.) But making infinitely many choices in one proof, or even an unknown finite number of choices, will depend on being able to give a rule stating which choice is to be made and when. This might be more difficult to achieve. Some versions of König's Lemma do indeed involve infinitely many arbitrary choices as we turn 'left' or 'right' following an infinite path. This is a theme that will be taken up in the next chapter. As a taster, you could attempt the following exercise, a more difficult version of Exercise 1.12.

Exercise 1.19 Consider the generalisation of König's Lemma that says that an infinite tree T in which every vertex has finite valency has an infinite path. Do not make any simplifying assumptions on the elements of the sequences $s \in T$. What choices have to be made in the course of the proof, and how might you specify all of these choices unambiguously in your proof?