3 *Mathematical Induction*

1. Introduction

In the previous two chapters, we discussed some of the basic ideas pertaining to formal languages. In most cases, the formal specification of the syntax of the language involved a 'nothing else' clause. The following is perhaps the simplest example.

> 'P' is an atomic formula;
> if σ is an atomic formula, then σ + '♯' is an atomic formula;
> nothing else is an atomic formula.

In Chapter 1, we developed a careful logical analysis of the first two clauses above, but we did not provide an analysis of the final clause. In the present chapter, we correct that situation.

2. Explicit Definitions

Before considering the *general* problem of 'nothing else', let us consider a much simpler example.

> every upper case Roman letter is an atomic formula;
> nothing else is an atomic formula.

Notice first that these two clauses can be rewritten as follows [where 'σ' is understood to be a universally quantified variable ranging over strings].

(i) if σ is an upper case Roman letter, then σ is an atomic formula;
(e) if σ is not an upper case Roman letter, then σ is not an atomic formula.

The two clauses may be called, respectively, the *inclusion clause* and the *exclusion clause*; between them, they tell us what is included, and what is excluded, from the class of atomic formulas.

Notice next that the inclusion and exclusion clauses can be combined into a single biconditional, as follows.

> σ is an atomic formula
> if
> and
> only if
> σ is an upper case Roman letter.

Here, the 'if' half corresponds to the inclusion clause, and the 'only if' half corresponds to the exclusion clause.

Finally, notice that, since there are exactly 26 upper case Roman letters, the notion of being such a letter can be rewritten explicitly as follows.

> a string σ is an upper case Roman letter
> if and only if
> σ = 'A', or σ = 'B', or …, or σ = 'Z'

Note carefully here that the ellipsis ('…') is *merely* one of laziness (or conciseness). In particular, it can be eliminated by explicitly writing in the 23 missing disjuncts.

3. Inductive Definitions

Now, let us reconsider our original definition.

(i1) ‘P’ is an atomic formula;
(i2) if σ is an atomic formula, then σ + ‘♯’ is an atomic formula;
(e) nothing else is an atomic formula.

Here, the first two clauses provide the inclusion information, and the third clause provides the exclusion information. Can we combine these into a single biconditional definition, like we did in the previous section? Well, sort of.

σ is an atomic formula
if and only if
σ = ‘P’, or σ = ‘P♯’, or σ = ‘P♯♯’, or …

Here, the ellipsis *cannot* be explicitly rewritten; there simply is no official formula that it is short for. Insofar as the third line above is a formula at all, it is an infinitely-long formula *merely suggested* by the above notation. Unfortunately, standard logic does not allow infinitely-long formulas. So, although the above “definition” is suggestive of its content, it cannot actually convey that content.

That is where mathematical induction comes to the rescue.

4. An Aside on Terminology

Like many other words with technical meanings, the word ‘induction’ means many different things in different fields (informal logic, psychology, physics, mathematics). For example, most informal logic texts mention the distinction between deductive inference and inductive inference, and the corresponding distinction between deduction and induction. However, in formal logic, and in particular in metalogic, ‘induction’ refers to ‘mathematical induction’, which is a specialized form of *deductive reasoning*.

There are two uses for mathematical induction — *inductive definitions*; and *inductive proofs*. Basically, inductive proofs are used to prove assertions about sets characterized by inductive definitions.

5. Induction in Arithmetic

Mathematical induction is used extensively in the formal theory of arithmetic. In this connection, recall that the natural numbers may be characterized by the Peano Axioms.

(p1) $N[0]$
(p2) $\forall x\{N[x] \rightarrow N[s(x)]\}$
(p3) $\sim\exists x[s(x)=0]$
(p4) $\forall x\forall y(s(x)=s(y) \rightarrow x=y)$
(p5) $\mathbb{P}[0]\ \&\ \forall x(\mathbb{P}[x] \rightarrow \mathbb{P}[s(x)])\ .\!\rightarrow \forall x(N[x] \rightarrow \mathbb{P}[x])$

Now, suppose we make the following correspondence.

0	⇆	‘P’
$N[x]$	⇆	$A[\sigma]$ [σ is an atomic formula]
$s(x)$	⇆	σ+‘♯’ [the ♯-successor of σ]

Then we can rewrite (p1)-(p5) as follows.

(a1) $A[\text{‘P’}]$
(a2) $\forall\sigma\{A[\sigma] \rightarrow A[\sigma + \text{‘♯’}]\}$
(a3) $\sim\exists\sigma[\text{‘P’} = \sigma + \text{‘♯’}]$
(a4) $\forall\sigma_1\forall\sigma_2\{\sigma_1+\text{‘♯’} = \sigma_2+\text{‘♯’} \;\rightarrow\; \sigma_1 = \sigma_2\}$
(a5) $\mathbb{P}[\text{‘P’}] \;\&\; \forall\sigma\{\mathbb{P}[\sigma] \rightarrow \mathbb{P}[\sigma+\text{‘♯’}]\} \;.\!\!\rightarrow \forall\sigma\{A[\sigma] \rightarrow \mathbb{P}[\sigma]\}$

First, notice that (a1) and (a2) are just the two inclusion clauses of our original definition. Next, notice that (a3) and (a4) are provable, insofar as formulas are a species of symbolic string. In that case, (a3) and (a4) are simply redundant.

That leaves clause (a5). As it turns out, (a5) is a formal and precise rendering of the exclusion clause.

nothing else is an atomic formula

Of course, if (a5) corresponds to the exclusion clause in the definition of atomic formula, then the Peano postulate (P5) corresponds to a similar exclusion clause, namely:

nothing else is a number

Exactly how (P5) captures ‘nothing else’ is discussed later in Section 10.

6. Inductive Proofs in Arithmetic

Proofs by induction are used to prove assertions about sets that are defined by induction. In arithmetic, for example, proofs of most familiar arithmetic theorems (‘x+y=y+x’, ‘x(y+z)=xy+xz’, etc.) are done by induction [see Appendix 3 of this chapter]. These theorems, of course, require that one has defined addition, multiplication, etc., but these are also defined by induction (what else!).

The basis for proofs by induction is the exclusion clause of the inductive definition, the clause that says that nothing else is a so-and-so. Once the exclusion clause is made precise, as it is done in the Peano Axioms, we have the basis for proofs by induction. Consider the exclusion clause of arithmetic rewritten somewhat informally.

(P5) suppose property ℙ holds of 0;
suppose further that, if ℙ holds of n, then it holds of n^+;
then ℙ holds of every number.

[Henceforth, we write ‘n^+’ for the successor of n.] In order to make this rendition of (P5) precise, we must write it as a single formula (a universal-conditional), we must make the quantifiers explicit, and we must make precise what ‘property’ means and what ‘holds’ means. There are at least four formal approaches to this [see Appendix 1 of this chapter.]

Irrespective of how we formally explicate it, the informal rendition (P5) provides the basic intuition. In particular, given (P5), if we want to prove that every number has a given property ℙ, it is sufficient to prove two things;

(1)	0 has ℙ	Base Case
(2)	for any n, if n has ℙ, then so does the successor of n	Inductive Case

As indicated, the proof divides into two cases — the *base case* (1), and the *inductive case* (2).

The inductive case is furthermore usually proven by universal-conditional derivation (UCD), which yields two further lines. Specifically,

we assume:

	n has ℙ	Inductive Hypothesis
to show:		
	n^+ has ℙ	Inductive Step

As indicated to the right, whereas the assumption is called the *inductive hypothesis* the subsequent subderivation is called the *inductive step*.

The following summarizes the inductive method as a natural deduction scheme.

	(1)	~~SHOW~~: $\forall x\{N[x] \rightarrow \mathbb{P}[x]\}$	2,4, MI(p5)
BC:	(2)	| ~~SHOW~~: $\mathbb{P}[0]$	
	(3)	| |	
IC:	(4)	| ~~SHOW~~: $\forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[x^+]\}$	UCD
IH:	(5)	| | $\mathbb{P}[n]$	As
IS:	(6)	| | ~~SHOW~~: $\mathbb{P}[n^+]$	
	(7)	| | |	

7. Inductive Proofs in Meta-Logic

Mathematical induction works for any inductively defined set, not just the natural numbers. Consider the earlier definition of atomic formula.

‘P’ is an atomic formula;
if σ is an atomic formula, then σ + ‘♯’ is an atomic formula;
nothing else is an atomic formula.

Given the formal rendering of the exclusion (‘nothing else’) clause,

$$\mathbb{P}[\text{‘P’}] \;\&\; \forall\sigma\{\mathbb{P}[\sigma] \rightarrow \mathbb{P}[\sigma+\text{‘}\sharp\text{’}]\} \;.\!\!\rightarrow \forall\sigma\{A[\sigma] \rightarrow \mathbb{P}[\sigma]\}$$

in order to prove that every atomic formula has a given property ℙ, we proceed by induction. In particular, in order to prove that every atomic formula has property ℙ, it is *sufficient* to prove two things;

(1)	‘P’ has ℙ	Base Case
(2)	for any σ, if σ has ℙ, then so does σ+‘♯’	Inductive Case

As in arithmetic, the inductive case is usually proven by universal-conditional derivation, which yields two further lines. Specifically,

we assume:

	σ has ℙ	Inductive Hypothesis
to show:		
	σ+‘♯’ has ℙ	Inductive Step

The following summarizes the inductive method as a natural deduction scheme.

	(1)	~~SHOW~~: $\forall x\{A[x] \rightarrow \mathbb{P}[x]\}$	2,4, MI(a5)
BC:	(2)	│ ~~SHOW~~: $\mathbb{P}[\text{'P'}]$	
	(3)	│ │	
IC:	(4)	│ ~~SHOW~~: $\forall\sigma\{\mathbb{P}[\sigma] \rightarrow \mathbb{P}[\sigma+\text{'}\sharp\text{'}]\}$	UCD
IH:	(5)	│ │ $\mathbb{P}[\sigma]$	As
IS:	(6)	│ │ ~~SHOW~~: $\mathbb{P}[\sigma+\text{'}\sharp\text{'}]$	
	(7)	│ │ │	

This is the simplest example, and corresponds very closely to induction in arithmetic. Meta-logic definitions, however, are generally much more complex. As a slightly more complex example, consider the following definition.

every atomic formula is a formula;
if σ is a formula, then so is '~'+σ;
nothing else is a formula.

How do we formalize the 'nothing else' clause? Written informally, it goes as follows.

suppose every atomic formula has property ℙ;
further suppose that, if σ has property ℙ, then so does '~'+σ;
then every formula has property ℙ.

On the other hand, the formal rendering goes as follows.

$\forall x\{A[x] \rightarrow \mathbb{P}[x]\}$ &
$\forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[\text{'}\sim\text{'}+x]$.→
$\forall x\{F[x] \rightarrow \mathbb{P}[x]\}$

This in turn yields the following natural deduction scheme.

	(1)	~~SHOW~~: $\forall x\{F[x] \rightarrow \mathbb{P}[x]\}$	2,4, MI
BC:	(2)	│ ~~SHOW~~: $\forall x\{A[x] \rightarrow \mathbb{P}[x]\}$	*
	(3)	│ │	
IC:	(4)	│ ~~SHOW~~: $\forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[\text{'}\sim\text{'}+x]\}$	UCD
IH:	(5)	│ │ $\mathbb{P}[\sigma]$	As
IS:	(6)	│ │ ~~SHOW~~: $\mathbb{P}[\text{'}\sim\text{'}+\sigma]$	
	(7)	│ │ │	

*Notice carefully that, insofar as the atomic formulas are defined inductively (e.g., as above), the base case will also be proved by math induction.

As a slightly more complicated example, consider the following definition.

every atomic formula is a formula;
if σ is a formula, then so is '~'+σ;
if σ_1 and σ_2 are formulas, then so is '(' + σ_1 + '&' + σ_2 + ')'
nothing else is a formula.

In this case, the 'nothing else' clause can be informally rewritten as follows.

suppose every atomic formula has property $\mathbb{P}$;
further suppose that, if σ has property $\mathbb{P}$, then so does '$\sim$'+σ;
further suppose that, if σ_1 and σ_2 have property $\mathbb{P}$, then so does '(' + σ_1 + '&' + σ_2 + ')';
then every formula has property $\mathbb{P}$.

So if we want to prove that every formula has a given property $\mathbb{P}$, we proceed by induction. The base case is the same as the previous example. On the other hand, the inductive case divides into two parts – one for negation, one for conjunction. In effect we show that forming negations and forming conjunctions "preserves" the given property $\mathbb{P}$.

Furthermore, if we add further molecular formation clauses (e.g., disjunctions, conditionals), then we must add corresponding clauses to the inductive case.

8. Expanded Induction in Arithmetic

In the current section, we show how the scheme for induction can be expanded slightly. What we show is that the following two schemes are interchangeable in arithmetic.

(m) $\mathbb{P}[0]\ \&\ \forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[x^+]\}\ .\!\rightarrow \forall x\{Nx \rightarrow \mathbb{P}[x]\}$

(e) $\mathbb{P}[0]\ \&\ \forall x\{Nx\ \&\ \mathbb{P}[x]\ .\!\rightarrow \mathbb{P}[x^+]\}\ .\!\rightarrow \forall x\{Nx \rightarrow \mathbb{P}[x]\}$

We call (m) the *minimal formulation* of induction, and we call (e) the *expanded formulation*. Whereas the minimal formulation is the exact formal rendering of the 'nothing else' clause in arithmetic, the expanded formulation is a further theorem of arithmetic that follows from the minimal formulation together with axioms (p1) and (p2).

First of all, for *any given* instance of $\mathbb{P}$, it is a matter of predicate logic that (e) implies (m). The following is a formal proof.

(1)	$\mathbb{P}[0]\ \&\ \forall x\{Nx\ \&\ \mathbb{P}[x]\ .\!\rightarrow \mathbb{P}[x^+]\}\ .\!\rightarrow \forall x\{Nx \rightarrow \mathbb{P}[x]\}$	Pr
(2)	~~SHOW~~: $\mathbb{P}[0]\ \&\ \forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[x^+]\}\ .\!\rightarrow \forall x\{Nx \rightarrow \mathbb{P}[x]\}$	CD
(3)	$\mid$ $\mathbb{P}[0]\ \&\ \forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[x^+]\}$	As
(4)	$\mid$ ~~SHOW~~: $\forall x\{Nx \rightarrow \mathbb{P}[x]\}$	DD
(5)	$\mid\ \mid$ $\forall x\{Nx\ \&\ \mathbb{P}[x]\ .\!\rightarrow \mathbb{P}[x^+]\}$	3b,QL
(6)	$\mid\ \mid$ $\forall x\{Nx \rightarrow \mathbb{P}[x]\}$	1,3a,5,SL

The other direction is slightly more subtle. We show that, in Peano Arithmetic, if every instance of (m) is true, then every instance of (e) is true.

To do this, suppose every instance of (m) is true. Consider an arbitrary property $\mathbb{P}$. First, form a new property $\mathbb{P}'$, defined as follows.

$$\mathbb{P}'[\alpha] =_{\mathrm{df}} (N[\alpha]\ \&\ \mathbb{P}[\alpha])$$

Next, apply (m) to $\mathbb{P}'$.

$$\mathbb{P}'[0]\ \&\ \forall x\{\mathbb{P}'[x] \rightarrow \mathbb{P}'[x^+]\}\ .\!\rightarrow \forall x\{Nx \rightarrow \mathbb{P}'[x]\}$$

Next, expand $\mathbb{P}'$ according to its definition.

(a) $(N[0] \mathbin{\&} \mathbb{P}[0])$
$\&$
(b) $\forall x\{(N[x] \mathbin{\&} \mathbb{P}[x]) \rightarrow (N[x^+] \mathbin{\&} \mathbb{P}[x^+])\}$
$.\!\rightarrow$
(c) $\forall x\{Nx \rightarrow (Nx \mathbin{\&} \mathbb{P}[x])\}$

Applying arithmetic equivalences [i.e., equivalences that follow from the Peano Axioms] to components (a)-(c), we obtain the following arithmetically equivalent conditional.

(a) $\mathbb{P}[0]$
$\&$
(b) $\forall x\{(N[x] \mathbin{\&} \mathbb{P}[x]) \rightarrow \mathbb{P}[x^+]\}$
$.\!\rightarrow$
(c) $\forall x\{Nx \rightarrow \mathbb{P}[x]\}$

This is scheme (s).

The formal proof proceeds as follows.

(1)	$\forall\mathbb{P}\{\mathbb{P}[0] \mathbin{\&} \forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[x^+]\} \;.\!\rightarrow \forall x\{Nx \rightarrow \mathbb{P}[x]\}\}$	Pr
(2)	~~SHOW~~: $\forall\mathbb{P}\{\mathbb{P}[0] \mathbin{\&} \forall x\{Nx \mathbin{\&} \mathbb{P}[x] \;.\!\rightarrow \mathbb{P}[x^+]\} \;.\!\rightarrow \forall x\{Nx \rightarrow \mathbb{P}[x]\}\}$	UCD
(3)	$\vert\;\; \mathbb{P}_0[0] \mathbin{\&} \forall x\{Nx \mathbin{\&} \mathbb{P}_0[x] \;.\!\rightarrow \mathbb{P}_0[x^+]\}$	As
(4)	$\vert$ ~~SHOW~~: $\forall x\{Nx \rightarrow \mathbb{P}_0[x]\}$	UCD
(5)	$\vert\;\vert\;\; Na$	As
(6)	$\vert\;\vert$ ~~SHOW~~: $\mathbb{P}_0[a]$	5,15,QL
(7)	$\vert\;\vert\;\vert\;\; (N[0] \mathbin{\&} \mathbb{P}_0[0]) \mathbin{\&} \forall x\{(N[x] \mathbin{\&} \mathbb{P}_0[x]) \rightarrow (N[x^+] \mathbin{\&} \mathbb{P}_0[x^+])\} \;.\!\rightarrow \forall x\{Nx \rightarrow (N[x] \mathbin{\&} \mathbb{P}_0[x])\}$	1,∀O
(8)	$\vert\;\vert\;\vert\;\; N[0]$	P1
(9)	$\vert\;\vert\;\vert\;\; \mathbb{P}_0[0]$	3a
(10)	$\vert\;\vert\;\vert$ ~~SHOW~~: $\forall x\{(N[x] \mathbin{\&} \mathbb{P}_0[x]) \rightarrow (N[x^+] \mathbin{\&} \mathbb{P}_0[x^+])\}$	UCD
(11)	$\vert\;\vert\;\vert\;\vert\;\; N[b] \mathbin{\&} \mathbb{P}_0[b]$	As
(12)	$\vert\;\vert\;\vert\;\vert$ ~~SHOW~~: $N[b^+] \mathbin{\&} \mathbb{P}_0[b^+]$	13,14,SL
(13)	$\vert\;\vert\;\vert\;\vert\;\vert\;\; N[b^+]$	11a,P2
(14)	$\vert\;\vert\;\vert\;\vert\;\vert\;\; \mathbb{P}_0[b^+]$	11,3b,QL
(15)	$\vert\;\vert\;\vert\;\; \forall x\{Nx \rightarrow (N[x] \mathbin{\&} \mathbb{P}_0[x])\}$	8,9,10,SL

We have shown that the two formulations of induction – the minimal form, and the expanded form – are both theorems of arithmetic. Accordingly, we can use either formulation of mathematical induction when we do proofs by mathematical induction.

9. Expanded Induction in Meta-Logic

The reasoning employed in the previous section can be generalized and adapted to all inductive definitions. For example, consider our simplest example.

(a1) ‘P’ is an atomic formula;
(a2) if σ is an atomic formula, then so is σ+‘♯’;
(a3) nothing else is an atomic formula.

Whereas the exact formal rendering of (a3) is given as follows,

(m) $\mathbb{P}[\text{‘P’}] \mathbin{\&} \forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[x+\text{‘}\sharp\text{’}]\} \;.\!\rightarrow \forall x\{Ax \rightarrow \mathbb{P}[x]\}$

we can prove the following, using (m) along with (a1) and (a2) [exercise!].

(e) $\mathbb{P}[\text{‘P’}] \;\&\; \forall x\{Ax \;\&\; \mathbb{P}[x] \;.\!\rightarrow \mathbb{P}[x+\text{‘}\sharp\text{’}]\} \;.\!\rightarrow \forall x\{Ax \rightarrow \mathbb{P}[x]\}$

Both are theorems about atomic formulas; either can be used to do a proof by induction.

10. How Induction Captures ‘Nothing Else’

In the present section, we examine how the principle of induction captures the notion of “nothing else”. Let us concentrate on arithmetic, the inductive axioms of which are given as follows.

(p1) 0 is a number.
(p2) if m is a number, then so is m^+.
(p5) nothing else is a number.

Notice first that

nothing else is a number

means that

if an object is not mentioned in (p1) and (p2), then it is not a number.

What are the objects mentioned in (p1) and (p2)? Well, they are

0, 1, 2, 3, etc.

where it is understood that 1 is the successor of 0, and 2 is the successor of 1, etc.

So if an object is not one of these, then it is not a number. In other words, we have the following quasi-formula.

$\forall x\{x\neq 0 \;\&\; x\neq 1 \;\&\; x\neq 2 \;\&\; \ldots \;:\!\rightarrow \sim N[x]\}$

Now, it seems plausible that the official (finite!) rendering of this quasi-formula goes as follows.

(n) $\forall x\{x\neq 0 \;\&\; \forall y(x\neq y \rightarrow x\neq y^+) \;.\!\rightarrow \sim N[x]\}$

Now, we can show this assuming the official Peano formulation of (p5).

(p5) $\mathbb{P}[0] \;\&\; \forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[x^+]\} \;.\!\rightarrow \forall x\{N[x] \rightarrow \mathbb{P}[x]\}$

where $\mathbb{P}$ is understood to be any property/formula. To show (n), we use universal derivation (UD), which reduces the problem to showing the following.

(c) $c\neq 0 \;\;\&\;\; \forall y(c\neq y \rightarrow c\neq y^+) \;\;.\!\rightarrow \sim N[c]$

Next, we substitute ‘$c\neq x$’ for ‘$\mathbb{P}[x]$’ in (p5) [remember, $\mathbb{P}[x]$ can be any property/formula.], which yields the following instance.

(i) $c\neq 0 \;\;\&\;\; \forall x\{c\neq x \rightarrow c\neq x^+\} \;\;.\!\rightarrow \;\forall x\{N[x] \rightarrow c\neq x\}$

Notice that the respective antecedents of (c) and (i) are obviously equivalent. Notice also that the respective consequents of (c) and (i) are logically equivalent (by Identity Logic). It follows that (c) and (i) are logically equivalent. Accordingly, since (c) is equivalent to (i), which follows from (p5), (c) follows from (p5). But (c) yields (n) by universal generalization. Thus, (p5) entails (n).

This demonstrates that the principle of induction implies the 'nothing else' clause. It does not demonstrate the converse – that the 'nothing else' clause implies the principle of induction. In order to see this, we argue *informally* as follows.

Suppose our proposed formal rendering of the 'nothing else' clause.

(n) $\forall x\{x{\neq}0 \ \& \ \forall y\{x{\neq}y \rightarrow x{\neq}y^{+}\} \ .\!\!\rightarrow \ \forall x\{N[x] \rightarrow y{\neq}x\}\}$

We wish to show the induction clause.

(p5) $\mathbb{P}[0] \ \& \ \forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[x^{+}]\} \ .\!\!\rightarrow \forall x\{N[x] \rightarrow \mathbb{P}[x]\}$

where $\mathbb{P}$ is an arbitrary property. So suppose the antecedent, to show the consequent. In order to show the consequent, suppose c is a number, to show that c has property $\mathbb{P}$. Or, equivalently, suppose that c does not have property $\mathbb{P}$ – i.e., $\sim\mathbb{P}[c]$ – to show that c is not a number.

Now our original antecedent

$$\mathbb{P}[0] \ \& \ \forall x\{\mathbb{P}[x] \rightarrow \mathbb{P}[x^{+}]\}$$

implies the following infinite list of formulas.

$\mathbb{P}[0]$	
$\mathbb{P}[0^{+}]$	$\mathbb{P}[1]$
$\mathbb{P}[0^{++}]$	$\mathbb{P}[2]$
$\mathbb{P}[0^{+++}]$	$\mathbb{P}[3]$
etc.	

Using this infinite list along with our assumption that $\sim\mathbb{P}[c]$, we can apply Identity Logic to deduce the following infinite list.

$c \neq 0$
$c \neq 1$
$c \neq 2$
$c \neq 3$
etc.

Since the objects mentioned (0, 1, 2, …) are all the numbers, and c is not one of the objects mentioned, c is not a number.

Note carefully, that the previous argument is informal. There is no formal proof that corresponds to it. The reason is that formal proofs are, by definition, finite sequences of formulas. By contrast, the previous argument employs two different infinite lists.

11. Exercises For Chapter 3

1. Consider formal language $\mathcal{L}$, specified as follows [using quote/plus notation].

(v) Vocabulary: ‘N’, ‘p’, ‘♯’.
(a1) ‘p’ is an atomic formula.
(a2) if σ is an atomic formula, then so is σ+‘♯’.
(a3) nothing else is an atomic formula.
(f1) every atomic formula is a formula.
(f2) if σ is a formula, then so is ‘N’+σ.
(f3) nothing else is a formula.

1. Formalize the exclusion clauses above.
2. Prove that ‘N♯p’ is not a formula of $\mathcal{L}$.
3. Prove that every formula of $\mathcal{L}$ is finite. You will find the following lemmas useful.

L1. Every symbol is a finite string.
L2. For any strings σ_1, σ_2, if σ_1 and σ_2 are finite, then σ_1+σ_2 is finite.

2. Consider formal language $\mathcal{L}$, specified as follows [using minimal notation].

(v) Vocabulary: P, ♯.
(f1) P is a formula.
(f2) if σ is a formula, then so is σ♯.
(f3) nothing else is a formula.

1. Formalize the exclusion clause above.
2. Prove that ‘♯P’ is not a formula of $\mathcal{L}$.
3. Prove that every formula of $\mathcal{L}$ is finite.
4. Prove the following, where the Greek letters range over expressions in $\mathcal{L}$.
∀α[α♯ ≠ P]
∀α∀β{α♯ = β♯ → α=β}

3. Consider formal language $\mathcal{L}$, specified as follows [using quine/quote notation].

(v) Vocabulary: ‘P’, ‘♯’, ‘~’, ‘&’, ‘(’, ‘)’
(a1) ‘P’ is an atomic formula.
(a2) if σ is an atomic formula, then so is ¸ ⌜σ♯⌝ 1.
(a3) nothing else is an atomic formula.
(f1) every atomic formula is a formula.
(f2) if σ is a formula, then so is ¸ ⌜~σ 1⌝.
(f3) if σ_1 and σ_2 are formulas, then so is ⌜¸ (σ_1&σ_2) 1⌝.
(f4) nothing else is a formula.

1. Formalize the exclusion clauses above.
2. Prove that ‘P♯P’ is not a formula of $\mathcal{L}$
3. Prove that every formula of $\mathcal{L}$ is finite.
4. Prove the following, where the Greek letters range over expressions in $\mathcal{L}$.
∀α[¸ ⌜~α⌝ 1 ≠ ‘P’]
∀αβ{¸ ⌜~α⌝ 1 = ⌜¸ ~β⌝ 1 → α=β}
∀αβγδ{¸ ⌜(α&β)⌝ 1 = ⌜¸ (γ&δ)⌝ 1 →. α=γ & β=δ}

12. Answers to Selected Exercises

1. 1. exclusion clauses:

(a3) {ℙ[‘p’] & ∀x{ℙ[x]→ℙ[x+‘♯’]}}→∀x{A[x]→ℙ[x]}
(f3) {∀x{A[x]→ℙ[x]} & ∀x{ℙ[x]→ℙ[‘N’+x]}}→∀x{F[x]→ℙ[x]}

3. Prove that every formula of $\mathcal{L}$ is finite.

(1)	~~SHOW~~: ∀x{F[x] → fin[x]}	MI(f3)
(2)	│ ~~SHOW~~: ∀x{A[x] → fin[x]}	MI(a3)
(3)	│ │ ~~SHOW~~: fin[‘p’]	v+L01
(4)	│ │ ~~SHOW~~: ∀x{fin[x] → fin[x+‘♯’]}	UCD
(5)	│ │ │ fin[s]	As
(6)	│ │ │ ~~SHOW~~: fin[s+‘♯’]	5,7,L02
(7)	│ │ │ │ fin[‘♯’]	v+L01
(8)	│ ~~SHOW~~: ∀x{fin[x] → fin[‘N’+x]}	UCD
(9)	│ │ fin[s]	As
(10)	│ │ ~~SHOW~~: fin[‘N’+s]	9,11,L02
(11)	│ │ │ fin[‘N’]	v+L01

2. 1. exclusion clause:

(f3) ℙ[P] & ∀x{ℙ[x]→ℙ[x♯]} .→ ∀x{A[x]→ℙ[x]}

3. 1. exclusion clauses:

(a3) ℙ[P] & ∀x{ℙ[x]→ℙ[¸ x♯¹]} .→ ∀x{A[x]→ℙ[x]}
(f3) ∀x{A[x]→ℙ[x]} & ∀x{ℙ[x]→ℙ[¸ ~x¹]} & ∀x∀y{ℙ[x] & ℙ[y] .→ ℙ[¸ (x&y)¹]} .→ ∀x{F[x]→ℙ[x]}

13. Appendix 1 – Approaches to Properties

As mentioned in Section 6, the informal rendition of the exclusion clause requires explication. In this regard, there are four approaches, given as follows.

1. First-Order Logic

This approach takes properties to be formulas with one free variable, and takes holding as satisfaction; '$\mathbb{P}$ holds of e' means that e satisfies $\mathbb{P}$, $\mathbb{P}$ being a formula with one free variable. Thus, making the quantifiers explicit, we have the following rewrite of (P5).

(P5f) $\mathbb{F}[0] \;\&\; \forall x(\mathbb{F}[x] \rightarrow \mathbb{F}[x^+]) \;.\!\!\rightarrow \forall x(N[x] \rightarrow \mathbb{F}[x])\}$

Here, '$\mathbb{F}$' is not an object variable but a metalinguistic variable; (P5f) is a schema, which is short for infinitely many different formulas, one for each formula $\mathbb{F}$.

2. Higher-Order Logic

Takes properties to be one-place predicates, takes holding to be ordinary predication; the twist is that there are predicate *variables* over which one can quantify. '$\mathbb{P}$ holds of e' means the same as '$\mathbb{P}[e]$'. In this case, (P5) is written as follows.

(P5h) $\forall\mathbb{P}\{\mathbb{P}[0] \;\&\; \forall x(\mathbb{P}[x] \rightarrow \mathbb{P}[x^+]) \;.\!\!\rightarrow \forall x(N[x] \rightarrow \mathbb{P}[x])\}$

3. Set Theory

Takes properties to be sets, and takes holding as containing as an element; '$\mathbb{P}$ holds of e' means that $e\in\mathbb{P}$. In this case, (P5) is written as follows.

(P5s) $\forall\mathbb{P}\{0\in\mathbb{P} \;\&\; \forall x(x\in\mathbb{P} \rightarrow x^+\in\mathbb{P}) \;.\!\!\rightarrow \forall x(x\in N \rightarrow x\in\mathbb{P}\}$

Here, N is the set of natural numbers.

4. Property Theory

Takes properties to be primitive elements in the domain, just like individuals, and introduces an additional 2-place logical predicate 'instantiates', symbolized by 'Δ'. In this case, (P5) is written as follows.

(P5p) $\forall\mathbb{P}\{0\Delta\mathbb{P} \;\&\; \forall x(x\Delta\mathbb{P} \rightarrow x^+\Delta\mathbb{P}) \;.\!\!\rightarrow \forall x(x\Delta N \rightarrow x\Delta\mathbb{P}\}$

Here, N is the property of being a number. The set-theoretic and the property-theoretic formulations of (p5) are isomorphic. The difference concerns the respective theories of '$\in$' and 'Δ'. The most important difference between properties and sets is that sets satisfy the principle of extensionality and properties do not. For example, the set of unicorns is the same as the set of flying horses; however, the property of being a unicorn is different from the property of being a flying horse.

Often, when a logician does a proof by induction, it is not always clear which approach to properties is presupposed. However, it is usually safe to assume that the set-theoretic rendition will capture what is "really" going on.

14. Appendix 2 – The Basic Schemes of Induction

1. Introduction

In the present appendix, we present the techniques of mathematical induction within the framework of natural deduction. This will hopefully make the structure of inductive proofs more obvious.

2. Induction for the Natural Numbers

1. Minimal Form

	~~SHOW~~: $\forall x\{Nx \to \mathbb{P}[x]\}$	MI
BC:	~~SHOW~~: $\mathbb{P}[0]$	??
IC:	~~SHOW~~: $\forall x(\mathbb{P}[x] \to \mathbb{P}[x^+])$	UCD
IH:	$\mathbb{P}[n]$	As
IS:	~~SHOW~~: $\mathbb{P}[n^+]$	??

2. Expanded Form

	~~SHOW~~: $\forall x\{Nx \to \mathbb{P}[x]\}$	MI
BC:	~~SHOW~~: $\mathbb{P}[0]$	??
IC:	~~SHOW~~: $\forall x(Nx \,\&\, \mathbb{P}[x] \;.\!\!\to \mathbb{P}[x^+])$	UCD
IH:	$Nn \,\&\, \mathbb{P}[n]$	As
IS:	~~SHOW~~: $\mathbb{P}[n^+]$	??

3. Strong Induction

	~~SHOW~~: $\forall x\{Nx \to \mathbb{P}[x]\}$	MI
BC:	~~SHOW~~: $\mathbb{P}[0]$	??
IC:	~~SHOW~~: $\forall x\{\forall y(y<x \to \mathbb{P}[y]) \;\to\; \mathbb{P}[x]\}$	UCD
IH:	$\forall y(y<m \to \mathbb{P}[y])$	As
IS:	~~SHOW~~: $\mathbb{P}[m]$	??

Note: Officially, strong induction does not have a base case, since the base case follows from the inductive case. It is included here "for good measure"; this is because in many actual applications of strong induction, one ends up explicitly proving the base case.

4. Double Induction (Weak Form)

Note: the following scheme presumes that the quantifiers range over numbers.

	~~SHOW~~: $\forall x\forall y\mathbb{R}[x,y]$	MI
BC0:	│ ~~SHOW~~: $\forall y\mathbb{R}[0,y]$	MI
BC1:	│ │ ~~SHOW~~: $\mathbb{R}[0,0]$	??
	│ │ │	
	│ │ │	
IC1:	│ │ ~~SHOW~~: $\forall y(\mathbb{R}[0,y] \rightarrow \mathbb{R}[0,y^{+}])$	UCD *
IH1:	│ │ │ $\mathbb{R}[0,m]$	As
IS1:	│ │ │ ~~SHOW~~: $\mathbb{R}[0,m^{+}]$	??
	│ │ │	
	│ │ │	
IC0:	│ │ ~~SHOW~~: $\forall x(\forall y\mathbb{R}[x,y] \rightarrow \forall y\mathbb{R}[x^{+},y])$	UCD *
IH0:	│ │ │ $\forall y\mathbb{R}[m,y]$	As
IS0:	│ │ │ ~~SHOW~~: $\forall y\mathbb{R}[m^{+},y]$	MI
BC2:	│ │ │ │ ~~SHOW~~: $\mathbb{R}[m^{+},0]$	??
	│ │ │ │ │	
	│ │ │ │ │	
IC2:	│ │ │ │ ~~SHOW~~: $\forall y(\mathbb{R}[m^{+},y] \rightarrow \mathbb{R}[m^{+},y^{+}])$	UCD *
IH2:	│ │ │ │ │ $\mathbb{R}[m^{+},n]$	As
IS2:	│ │ │ │ │ ~~SHOW~~: $\mathbb{R}[m^{+},n^{+}]$	??
	│ │ │ │ │ │	
	│ │ │ │ │ │	

5. Induction for Formulas

Consider formal language $\mathcal{L}$ specified as follows.

(v) Vocabulary: ‘N’, ‘p’, ‘♯’.
(a1) ‘p’ is an atomic formula.
(a2) if σ is an atomic formula, then so is σ+‘♯’.
(a3) nothing else is an atomic formula.
(f1) every atomic formula is a formula.
(f2) if σ is a formula, then so is ‘N’+σ.
(f3) nothing else is a formula.

1. Atomic Formulas; Minimal Form

	~~SHOW~~: $\forall x\{Ax \rightarrow \mathbb{P}[x]\}$	MI(a3)
BC:	│ ~~SHOW~~: $\mathbb{P}[\text{‘p’}]$	?
	│ │	
	│ │	
IC:	│ ~~SHOW~~: $\forall x(\mathbb{P}[x] \rightarrow \mathbb{P}[x+\text{‘♯’}])$	UCD
IH:	│ │ $\mathbb{P}[s]$	As
IS:	│ │ ~~SHOW~~: $\mathbb{P}[s+\text{‘♯’}]$	?
	│ │ │	
	│ │ │	

2. Atomic Formulas; Expanded Form

	~~SHOW~~: $\forall x\{Ax \to \mathbb{P}[x]\}$	MI(a3)
BC:	~~SHOW~~: $\mathbb{P}[\text{‘p’}]$	?
IC:	~~SHOW~~: $\forall x(Ax \,\&\, \mathbb{P}[x] \;.\!\!\to \mathbb{P}[x+\text{‘}\sharp\text{’}])$	UCD
IH:	$As \,\&\, \mathbb{P}[s]$	As
IS:	~~SHOW~~: $\mathbb{P}[s+\text{‘}\sharp\text{’}]$	?

3. Formulas; Minimal Form

	~~SHOW~~: $\forall x\{Fx \to \mathbb{P}[x]\}$	MI(f3)
BC:	~~SHOW~~: $\forall x\{Ax \to \mathbb{P}[x]\}$	MI(a3)
	see above	
IC:	~~SHOW~~: $\forall x(\mathbb{P}[x] \to \mathbb{P}[\text{‘N’}+x])$	UCD
IH:	$\mathbb{P}[s]$	As
IS:	~~SHOW~~: $\mathbb{P}[\text{‘N’}+s]$	?

4. Formulas; Expanded Form

	~~SHOW~~: $\forall x\{Fx \to \mathbb{P}[x]\}$	MI(f3)
BC:	~~SHOW~~: $\forall x\{Ax \to \mathbb{P}[x]\}$	MI(a3)
	see above	
IC:	~~SHOW~~: $\forall x(Fx \,\&\, \mathbb{P}[x] \;.\!\!\to \mathbb{P}[\text{‘N’}+x])$	UCD
IH:	$Fs \,\&\, \mathbb{P}[s]$	As
IS:	~~SHOW~~: $\mathbb{P}[\text{‘N’}+s]$	?

6. Strong Induction

(1)	~~SHOW~~: $\forall y \forall x\{x<y \to \mathbb{P}[x]\} \to \forall x \mathbb{P}[x]$	CD
(2)	│ $\forall y \forall x\{x<y \to \mathbb{P}[x]\}$	As
(3)	│ ~~SHOW~~: $\forall x \mathbb{P}[x]$	MI(w)
(4)	│ │ ~~SHOW~~: $\mathbb{P}[0]$	DD
(5)	│ │ │ $\sim\exists x[x<0]$	Lemma
(6)	│ │ │ $\forall x\{x<0 \to \mathbb{P}[0]\}$	2,QL
(7)	│ │ │ $\mathbb{P}[0]$	5,6,QL
(8)	│ │ │ ~~SHOW~~: $\forall x\{\mathbb{P}[x] \to \mathbb{P}[x^+]\}$	UCD
(9)	│ │ │ │ $\mathbb{P}[m]$	As
(10)	│ │ │ │ ~~SHOW~~: $\mathbb{P}[m^+]$	2,11,QL
(11)	│ │ │ │ │ ~~SHOW~~: $\forall x\{x<m^+ \to \mathbb{P}[x]\}$	MI(w)
(12)	│ │ │ │ │ │ ~~SHOW~~: $0<m^+ \to \mathbb{P}[0]$	4,SL
(13)	│ │ │ │ │ │ ~~SHOW~~: $\forall x\{\{x<m^+ \to \mathbb{P}[x]\} \to \{x^+<m^+ \to \mathbb{P}[x^+]\}\}$	UCD
(14)	│ │ │ │ │ │ │ $n<m^+ \to \mathbb{P}[n]$	As
(15)	│ │ │ │ │ │ │ ~~SHOW~~: $n^+<m^+ \to \mathbb{P}[n^+]$	CD
(16)	│ │ │ │ │ │ │ │ $n^+<m^+$	As
(17)	│ │ │ │ │ │ │ │ ~~SHOW~~: $\mathbb{P}[n^+]$	
(18)	│ │ │ │ │ │ │ │ │ $n^+ < m \vee n^+ = m$	16,Lemma

15. Appendix 3 – Examples of Mathematical Induction From Arithmetic

1. Basic Scheme for Single Induction

Note: the following scheme presumes that the quantifiers range over numbers.

	~~SHOW~~: $\forall x \mathbb{P}[x]$	MI
BC:	│ ~~SHOW~~: $\mathbb{P}[0]$	??
	│ │	
IC:	│ ~~SHOW~~: $\forall x(\mathbb{P}[x] \to \mathbb{P}[x^+])$	UCD*
IH:	│ │ $\mathbb{P}[n]$	As
IS:	│ │ ~~SHOW~~: $\mathbb{P}[n^+]$	??
	│ │ │	

2. Basic Scheme for Double Induction

Note: the following scheme presumes that the quantifiers range over numbers.

	~~SHOW~~: $\forall x\forall y\mathbb{R}[x,y]$	MI
BC0:	~~SHOW~~: $\forall y\mathbb{R}[0,y]$	MI
BC1:	~~SHOW~~: $\mathbb{R}[0,0]$	??
IC1:	~~SHOW~~: $\forall y(\mathbb{R}[0,y] \rightarrow \mathbb{R}[0,y^+])$	UCD*
IH1:	$\mathbb{R}[0,m]$	As
IS1:	~~SHOW~~: $\mathbb{R}[0,m^+]$	??
IC0:	~~SHOW~~: $\forall x(\forall y\mathbb{R}[x,y] \rightarrow \forall y\mathbb{R}[x^+,y])$	UCD*
IH0:	$\forall y\mathbb{R}[m,y]$	As
IS0:	~~SHOW~~: $\forall y\mathbb{R}[m^+,y]$	MI
BC2:	~~SHOW~~: $\mathbb{R}[m^+,0]$	??
IC2:	~~SHOW~~: $\forall y(\mathbb{R}[m^+,y] \rightarrow \mathbb{R}[m^+,y^+])$	UCD*
IH2:	$\mathbb{R}[m^+,n]$	As
IS2:	~~SHOW~~: $\mathbb{R}[m^+,n^+]$	??

* The inductive case line is optional.

3. Definitions

(Def +) $m+0 = m$
$m+n^+ = (m+n)^+$

(Def ×) $m\times 0 = 0$
$m\times n^+ = m+(m\times n)$

4. Parenthesis Conventions

$(a\times b)+c$ is written $ab+c$
$a\times(b+c)$ is written $a(b+c)$
$a+b^+$ is a plus the successor of b
$(a+b)^+$ is the successor of $a+b$

5. Examples of Single Induction from Arithmetic

1. The Associative Law for Addition

	(1)	~~SHOW~~: $\forall x\forall y\forall z[(x+y)+z = x+(y+z)]$	UD2
	(2)	~~SHOW~~: $\forall z[(a+b)+z = a+(b+z)]$	MI
BC:	(3)	~~SHOW~~: $(a+b)+0 = a+(b+0)$	DD
	(4)	$(a+b)+0 = a+b$	Def +
	(5)	$b+0 = b$	Def +
	(6)	$a+(b+0) = a+b$	5,IL
	(7)	$(a+b)+0 = a+(b+0)$	4,6,IL
IC:	(*)	~~SHOW~~: $\forall x[(a+b)+x = a+(b+x) \rightarrow (a+b)+x^{+} = a+(b+x^{+})]$	UCD*
IH:	(8)	$(a+b)+m = a+(b+m)$	As
IS:	(9)	~~SHOW~~: $(a+b)+m^{+} = a+(b+m^{+})$	DD
	(10)	$(a+b)+m^{+} = [(a+b)+m]^{+}$	Def +
	(11)	$= [a+(b+m)]^{+}$	IH
	(12)	$b+m^{+} = (b+m)^{+}$	Def +
	(13)	$a+(b+m^{+}) = a+(b+m)^{+}$	12,IL
	(14)	$= [a+(b+m)]^{+}$	Def +
	(15)	$(a+b)+m^{+} = a+(b+m^{+})$	11,14,IL

2. The Distributive Law

	(1)	~~SHOW~~: $\forall x\forall y\forall z[x(y+z) = xy+xz]$	UD
	(2)	~~SHOW~~: $\forall z[a(b+z) = ab+az]$	MI
BC:	(3)	~~SHOW~~: $a(b+0) = ab+a0$	DD
	(4)	$b+0 = b$	Def +
	(5)	$a(b+0) = ab$	4,IL
	(6)	$a0 = 0$	Def ×
	(7)	$ab+a0 \quad = ab+0$	6,IL
	(8)	$= ab$	Def +
	(9)	$a(b+0) = ab+a0$	5,8,IL
IC:	(*)	~~SHOW~~: $\forall x[a(b+x) = ab+ax \rightarrow a(b+x^{+}) = ab + ax^{+}$	UCD*
IH:	(10)	$a(b+m) = ab+am$	As
IS:	(11)	~~SHOW~~: $a(b+m^{+}) = ab+am^{+}$	DD
	(12)	$b+m^{+} = (b+m)^{+}$	Def +
	(13)	$a(b+m^{+}) \quad = a(b+m)^{+}$	12,IL
	(14)	$= a+(a(b+m))$	Def ×
	(15)	$= a+(ab+am)$	IH,IL
	(16)	$am^{+} = a+am$	Def ×
	(17)	$ab+am^{+} \quad = ab+(a+am)$	16,IL
	(18)	$= (ab+a)+am$	Ass(+)
	(19)	$= (a+ab)+am$	Com(+),IL
	(20)	$= a+(ab+am)$	Ass(+)
	(21)	$a(b+m^{+}) = ab+am^{+}$	15,20,IL

3. The Associative Law for Multiplication

	(1)	~~SHOW~~: $\forall x\forall y\forall z[x(yz) = (xy)z]$	UD2
	(2)	~~SHOW~~: $\forall z[a(bz) = (ab)z]$	MI
BC:	(3)	~~SHOW~~: $a(b0) = (ab)0$	ILD
	(4)	$a(b0) \quad = a0$	def ×
	(5)	$= 0$	def ×
	(6)	$(ab)0 \quad = 0$	def ×
IC:	(*)	~~SHOW~~: $\forall x\{a(bx)=(ab)x \rightarrow a(bx^+)=(ab)x^+\}$	UCD*
IH:	(7)	$a(bm) = (ab)m$	As
IS:	(8)	~~SHOW~~: $a(bm^+) = (ab)m^+$	ILD
	(9)	$a(bm^+) \quad = a(b+bm)$	def ×
	(10)	$= ab + a(bm)$	dist
	(11)	$= ab + (ab)m$	IH (7)
	(12)	$(ab)m^+ = ab + (ab)m$	def ×

6. Examples of Double Induction from Arithmetic

1. The Commutative Law for Addition

	(1)	~~SHOW~~: $\forall x \forall y(x+y = y+x)$	MI
BC0:	(2)	│ ~~SHOW~~: $\forall y(0+y = y+0)$	MI
BC1:	(3)	│ │ ~~SHOW~~: $0+0 = 0+0$	IL
IC1:	(*)	│ │ ~~SHOW~~: $\forall y(0+y=y+0 \rightarrow 0+y^{+}=y^{+}+0$	UCD
IH1:	(4)	│ │ │ $0+m = m+0$	As
IS1:	(5)	│ │ │ ~~SHOW~~: $0+m^{+} = m^{+}+0$	DD
	(6)	│ │ │ │ $0+m^{+} = (0+m)^{+}$	Def +
	(7)	│ │ │ │ $0+m = m+0$	IH1
	(8)	│ │ │ │ $m+0 = m$	Def +
	(9)	│ │ │ │ $0+m^{+} = m^{+}$	6,7,8, IL
	(10)	│ │ │ │ $m^{+}+0 = m^{+}$	Def +
	(11)	│ │ │ │ $0+m^{+} = m^{+}+0$	9,10,IL
IC0:	(*)	│ ~~SHOW~~: $\forall x[\forall y(m+y = y+m) \rightarrow \forall y(m^{+}+y = y+m^{+})]$	UCD
IH0:	(12)	│ │ $\forall y(m+y = y+m)$	As
IS0:	(13)	│ │ ~~SHOW~~: $\forall y(m^{+}+y = y+m^{+})$	MI
BC2:	(14)	│ │ │ ~~SHOW~~: $m^{+}+0 = 0+m^{+}$	DD
	(15)	│ │ │ │ $m^{+}+0 = m^{+}$	Def +
	(16)	│ │ │ │ $0+m^{+} = (0+m)^{+}$	Def +
	(17)	│ │ │ │ $0+m = m+0$	IH0, IL
	(18)	│ │ │ │ $m+0 = m$ Def +	
	(19)	│ │ │ │ $0+m = m$ 17,18,IL	
	(20)	│ │ │ │ $0+m^{+} = m^{+}$	16,19,IL
	(21)	│ │ │ │ $m^{+}+0 = 0+m^{+}$	15,20,IL
IC2:	(*)	│ │ │ ~~SHOW~~: $\forall x\{m^{+}+x = n+m^{+} \rightarrow m^{+}+x^{+} = x^{+}+m^{+}\}$	UCD
IH2:	(22)	│ │ │ │ $m^{+}+n = n+m^{+}$	As
IS2:	(23)	│ │ │ │ ~~SHOW~~: $m^{+}+n^{+} = n^{+}+m^{+}$	DD
	(24)	│ │ │ │ │ $m^{+}+n^{+} = (m^{+}+n)^{+}$	Def +
	(25)	│ │ │ │ │ $= (n+m^{+})^{+}$	24,IH2
	(26)	│ │ │ │ │ $= (n+m)^{++}$	Def +
	(27)	│ │ │ │ │ $n^{+}+m^{+} = (n^{+}+m)^{+}$	Def +
	(28)	│ │ │ │ │ $= (m+n^{+})^{+}$	IH0 (12)
	(29)	│ │ │ │ │ $= (m+n)^{++}$	Def +
	(30)	│ │ │ │ │ $= (n+m)^{++}$	IH0
	(31)	│ │ │ │ │ $m^{+}+n = n+m^{+}$	26,30,IL

2. The Commutative Law for Multiplication

	(1)	~~SHOW~~: $\forall x \forall y[xy = yx]$	MI
BC0	(2)	│ ~~SHOW~~: $\forall y[0y = y0]$	MI
BC1	(3)	│ │ ~~SHOW~~: $00 = 00$	IL
IH1	(4)	│ │ │ $0m = m0$	As
IS1	(5)	│ │ │ ~~SHOW~~: $0m^+ = m^+0$ DD (IL)	
	(6)	│ │ │ $0m^+ = 0+0m$	def ×
	(7)	│ │ │ $= 0+m0$	IH (4)
	(8)	│ │ │ $= 0+0$	def ×
	(9)	│ │ │ $= 0$	def +
	(10)	│ │ │ $m^+0 = 0$	def ×
IH0	(11)	│ │ $\forall y[my = ym]$	As
IS0	(12)	│ │ ~~SHOW~~: $\forall y[m^+y = ym^+]$	MI
BC2	(13)	│ │ ~~SHOW~~: $m^+0 = 0m^+$	2,QL
IH2	(14)	│ │ │ $m^+n = nm^+$	As
IS2	(15)	│ │ │ ~~SHOW~~: $m^+n^+ = n^+m^+$	DD (IL)
	(16)	│ │ │ $m^+n^+ = m^+ + m^+n$	def ×
	(17)	│ │ │ $= m^+ + nm^+$	IH(14)
	(18)	│ │ │ $= m^+ + (n+nm)$	def ×
	(19)	│ │ │ $= (n+nm) + m^+$	com[+]
	(20)	│ │ │ $= [(n+nm)+m]^+$	def +
	(21)	│ │ │ $n^+m^+ = n^+ + n^+m$	def ×
	(22)	│ │ │ $= n^+ + mn^+$	IH(11)
	(23)	│ │ │ $= n^+ + (m + mn)$	def ×
	(24)	│ │ │ $= (m+mn) + n^+$	com[+]
	(25)	│ │ │ $= [(m+mn)+n]^+$	def +
	(26)	│ │ │ $= [(m+nm)+n]^+$	IH(11)
	(27)	│ │ │ $= [n+(m+nm)]^+$	ass[+]
	(28)	│ │ │ $= [n+(nm+m)]^+$	com[+]
	(29)	│ │ │ $= [(n+nm)+m]^+$	ass[+]

16. Appendix 4 – Strings

1. Some Theorems about Strings (String Theory)

S1 $\text{E!}[\sigma_1 + \sigma_2] \;\leftrightarrow.\; \text{finite}[\sigma_1] \;\&\; \text{finite}[\sigma_2]$
S2 $\text{finite}[\sigma] \;\leftrightarrow\; \text{E!}[\text{last}(\sigma)]$
S3 $\text{atomic\_string}[1^{st}(\sigma)]$
S4 $\text{atomic\_string}[\text{last}(\sigma)]$
S5 $\text{atomic\_string}[\sigma] \;\to\; 1^{st}(\sigma) = \sigma$
S6 $\text{atomic\_string}[\sigma] \;\to\; \text{last}(\sigma) = \sigma$
S7 $\text{E!}[\sigma_1 + \sigma_2] \;\to\; 1^{st}(\sigma_1 + \sigma_2) \;=\; 1^{st}(\sigma_1)$
S8 $\text{E!}[\sigma_1 + \sigma_2] \;\to\; \text{last}(\sigma_1 + \sigma_2) \;=\; \text{last}(\sigma_2)$
S9 $\sigma_1+(\sigma_2+\sigma_3) \;=\; (\sigma_1+\sigma_2)+\sigma_3$
S10 $\sigma_1+\sigma_2 = \sigma_2+\sigma_1 \;\to\; \sigma_1 = \sigma_2$
S11 $\sigma_1+\sigma_2 = \sigma_1+\sigma_3 \;\to\; \sigma_2 = \sigma_3$
S12 $\sigma_1+\sigma_3 = \sigma_2+\sigma_3 \;\to\; \sigma_1 = \sigma_2$

T1. (An example of a proof in String Theory based on S1-S12.)

(1)	~~SHOW~~: $\text{finite}[\sigma_1] \;\&\; \text{finite}[\sigma_2] \;.\!\!\to\; \text{finite}[\sigma_1 + \sigma_2]$	CD
(2)	│ $\text{finite}[\sigma_1] \;\&\; \text{finite}[\sigma_2]$	As
(3)	│ ~~SHOW~~: $\text{finite}[\sigma_1 + \sigma_2]$	4,S2
(4)	│ ~~SHOW~~: $\text{E!}[\text{last}(\sigma_1 + \sigma_2)]$	6,7,IL
(5)	│ │ $\text{E!}[\sigma_1 + \sigma_2]$	2,S0
(6)	│ │ $\text{last}(\sigma_1 + \sigma_2) = \text{last}(\sigma_2)$	4,S8
(7)	│ │ $\text{E!}[\text{last}(\sigma_2)]$	2b,S2

T2. (Another example)

(1)	~~SHOW~~: $\forall x\{\text{atomic\_string}[x] \to \text{finite}[x]\}$	UCD
(2)	│ atomic_string[s]	As
(3)	│ ~~SHOW~~: finite[s]	4,S2
(4)	│ ~~SHOW~~: E![last(s)]	5, def(E!)
(5)	│ ~~SHOW~~: $\exists x[x = \text{last}(s)]$	6,QL
(6)	│ │ last(s) = s	2,S6

2. Some Theorems about a Simple Formal Language, Proved by Induction

In the present section, we look at a few examples of inductive proofs in meta-logic. We employ the following definition of language $\mathcal{L}_0$, to which subsequent uses of 'atomic formula' and 'formula' refer. In symbolizing, we set 'A[α]' $=_{df}$ 'α is an atomic formula', and we set 'F[α]' $=_{df}$ 'α is a formula'.

The Language $\mathcal{L}_0$

(a1) 'p' is an atomic formula;
(a3) if σ is an atomic formula, then so is σ+'♯'
(a3) nothing else is an atomic formula.

(f1) every atomic formula is a formula;
(f2) if σ is a formula, then so is 'N'+σ;
(f3) nothing else is a formula.

T3.

(1)	~~SHOW~~: ∀x{A[x] → finite[x]}	MI(a3)	
(2)	│ ~~SHOW~~: finite[‘p’]	3, T2	BC
(3)	│ │ atomic_string(‘p’)	DI*	
(4)	│ ~~SHOW~~: ∀x{finite[x] → finite[x+‘♯’]}	UCD	IC
(5)	│ │ finite[s]	As	IH
(6)	│ │ ~~SHOW~~: finite[s+‘♯’]	5,8,T1	IS
(7)	│ │ │ atomic_string[‘♯’]	DI	
(8)	│ │ │ finite[‘♯’]	7,T2	

*‘DI’ = ‘direct inspection’ which works only on directly quoted material. A single-quote-expression names the object (betokened) within the quotes, so presumably we can assess *some* truths by direct inspection of the literal material. For example, ‘cat’ begins with ‘c’, and ‘cat’ ≠ ‘dog’.

T4.

(1)	~~SHOW~~: ∀x{F[x] → finite[x]}	MI(f3)	
(2)	│ ~~SHOW~~: ∀x{A[x] → finite[x]}	T3	BC
(3)	│ ~~SHOW~~: ∀x{finite[x] → finite[‘N’+x]}	UCD	IC
(4)	│ │ finite[s]	As	IH
(5)	│ │ ~~SHOW~~: finite[‘N’+s]	4,7,T1	IS
(6)	│ │ │ atomic_string[‘N’]	DI	
(7)	│ │ │ finite[‘N’]	6,T2	

T5.

(1)	~~SHOW~~: ∼A[‘♯p’]	2,IL	
(2)	~~SHOW~~: ∀x{A[x]→x≠‘♯p’}	MI(a3)	
(3)	│ ~~SHOW~~: ‘p’ ≠ ‘♯p’	DI	BC
(4)	│ ~~SHOW~~: ∀x{x ≠ ‘♯p’ → x+‘♯’ ≠ ‘♯p’}	UCD	IC
(5)	│ │ s ≠ ‘♯p’	As	IH
(6)	│ │ ~~SHOW~~: s+‘♯’ ≠ ‘♯p’	7-10,IL	IS
(7)	│ │ │ last(s+‘♯’) = last(‘♯’)	S8	
(8)	│ │ │ last(‘♯’) = ‘♯’	DI	
(9)	│ │ │ last(‘♯p’) = ‘p’	DI	
(10)	│ │ │ ‘♯’ ≠ ‘p’	DI	

T6.

(1)	~~SHOW~~: ∀x{A[x] →. 1^{st}(x) = ‘p’}	MI(a3)	
(2)	│ ~~SHOW~~: 1^{st}(‘p’) = ‘p’	DI	BC
(3)	│ ~~SHOW~~: ∀x{1^{st}(x) = ‘p’ → 1^{st}(x+‘♯’) = ‘p’}	UCD	IC
(4)	│ │ 1^{st}(s) = ‘p’	As	IH
(5)	│ │ ~~SHOW~~: 1^{st}(s+‘♯’) = ‘p’	4,6,IL	IS
(6)	│ │ │ 1^{st}(s+‘♯’) = 1^{st}(s)	S7	

T7.

(1)	~~SHOW~~: ∀x{F[x] →. 1^{st}(x) = ‘p’ or 1^{st}(x) = ‘N’}	MI(f3)	
(2)	~~SHOW~~: ∀x{A[x] →. 1^{st}(x) = ‘p’ or 1^{st}(x) = ‘N’}	T6+QL	BC
(3)	~~SHOW~~: ∀x{1^{st}(x) = ‘p’ or 1^{st}(x) = ‘N’ → 1^{st}(x+‘♯’) = ‘p’ or 1^{st}(x+‘♯’) = ‘N’}	UCD	IC
(4)	1^{st}(s) = ‘p’ ∨ 1^{st}(s) = ‘N’	As	IH
(5)	1^{st}(s+‘♯’) = 1^{st}(s)	S7	
(6)	~~SHOW~~: 1^{st}(s+‘♯’) = ‘p’ or 1^{st}(s+‘♯’) = ‘N’	4,SC	IS
(7)	c1: 1^{st}(s) = ‘p’	As	
(8)	1^{st}(s+‘♯’) = ‘p’	5,7,IL	
(9)	1^{st}(s+‘♯’) = ‘p’ or 1^{st}(s+‘♯’) = ‘N’	8,SL	
(10)	c2: 1^{st}(s) = ‘N’	As	
(11)	1^{st}(s+‘♯’) = ‘N’	5,10,IL	
(12)	1^{st}(s+‘♯’) = ‘p’ or 1^{st}(s+‘♯’) = ‘N’	11,SL	

The previous theorems are proven by minimal induction. In the next example, we see that minimal induction is not always enough. Here, we try to prove that the string ‘N♯p’ is not a formula of $\mathcal{L}_0$, using minimal induction.

(1)	~~SHOW~~: ~F[‘N♯p’]	2,IL	
(2)	SHOW: ∀x{F[x]→x≠‘N♯p’}	MI(f3)	
(3)	~~SHOW~~: ∀x{A[x]→x≠‘N♯p’}	MI(a3)	BC$_1$
(4)	~~SHOW~~: ‘p’≠‘N♯p’	DI	BC$_2$
(5)	~~SHOW~~: ∀x{x≠‘N♯p’ → x+‘♯’≠‘N♯p’}	UCD	IC$_2$
(6)	s≠‘N♯p’	As	IH$_2$
(7)	~~SHOW~~: s+‘♯’≠‘N♯p’	7-9,IL	IS$_2$
(8)	last(s+‘♯’) = ‘♯’	L2c	
(9)	last(‘N♯p’) = ‘p’	DI	
(10)	‘♯’ ≠ ‘p’	DI	
(11)	SHOW: ∀x{x≠‘N♯p’ → ‘N’+x ≠ ‘N♯p’}	UCD	IC$_1$
(12)	s≠‘N♯p’	As	IH$_1$
(13)	SHOW: ‘N’+s ≠ ‘N♯p’	???	IS$_1$

At this point we are stymied – one cannot prove the inductive case [line 11], because it is not true! A simple counter-example goes as follows.

Let s = ‘♯p’;
then s ≠ ‘N♯p’;
but ‘N’+s = ‘N♯p’.

Accordingly we turn to the expanded scheme of induction, which is always an option. Then, the inductive case goes as follows. (The shaded material is what is different in the inductive case.)

T8.

(1)	~~SHOW~~: ∼F[‘N♯p’]	2,IL	
(2)	~~SHOW~~: ∀x{F[x]→x≠‘N♯p’}	MI(f1-f3)	
(3)	~~SHOW~~: ∀x{A[x]→x≠‘N♯p’}	MI(a3)	BC_1
(4)	~~SHOW~~: ‘p’≠‘N♯p’	DI	BC_2
(5)	~~SHOW~~: ∀x{x≠‘N♯p’ → x+‘♯’≠‘N♯p’}	UCD	IC_2
(6)	s≠‘N♯p’	As	IH_2
(7)	~~SHOW~~: s+‘♯’≠‘N♯p’	7-9,IL	IS_2
(8)	last(s+‘♯’) = last(‘♯’) = ‘♯’	S6/DI	
(9)	last(‘N♯p’) = ‘p’	DI	
(10)	‘♯’ ≠ ‘p’	DI	
(11)	~~SHOW~~: ∀x{F[x] & x≠‘N♯p’ .→ ‘N’+x ≠ ‘N♯p’}	UCD	IC_2
(12)	F[s] & s≠‘N♯p’	As	IH_2
(13)	~~SHOW~~: ‘N’+s ≠ ‘N♯p’	ID	IS_2
(14)	‘N’+s = ‘N♯p’	As	
(15)	~~SHOW~~: ×	20,22,23,SL	
(16)	‘N♯p’ = ‘N’ + ‘♯p’	DI	
(17)	’N’ + s = ‘N’ + ‘♯p’	14,16,IL	
(18)	s = ‘♯p’	17,S11	
(19)	F[‘♯p’]	12,18,IL	
(20)	1^{st}(‘♯p’) = ‘p’ ∨ 1^{st}(‘♯p’) = ‘N’	19,T7	
(21)	1^{st}(‘♯p’) = ‘♯’	DI	
(22)	‘♯’ ≠ ‘p’	DI	
(23)	‘♯’ ≠ ‘N’	DI	

The following is another example that uses expanded induction.

T9.

(1)	~~SHOW~~: ∀x{F[‘N’+x] → F[x]}	2,QL	
(2)	~~SHOW~~: ∀x{∼F[x] → ∼F[‘N’+x]}	UCD	
(3)	∼F[s]	As	
(4)	~~SHOW~~: ∼F[‘N’+s]	5,IL	
(5)	~~SHOW~~: ∀x{F[x] → x≠‘N’+s}	MI(f1-f3)	
(6)	~~SHOW~~: ∀x{A[x] → x≠‘N’+s}	UCD	BC
(7)	A[a]	As	
(8)	~~SHOW~~: a ≠ ‘N’+s	9-12,IL	
(9)	1^{st}(a) = ‘p’	7,T6	
(10)	1^{st}(‘N’+s) = 1^{st}(‘N’)	S7	
(11)	1^{st}(‘N’) = ‘N’	DI	
(12)	‘p’ ≠ ‘N’	DI	
(13)	~~SHOW~~: ∀x{F[x] & x≠‘N’+s .→ ‘N’+x≠‘N’+s}	UCD	IC
(14)	F[b] & b ≠ ‘N’+s	As	IH
(15)	~~SHOW~~: ‘N’+b ≠ ‘N’+s	ID	IS
(16)	‘N’+b = ‘N’+s	As	
(17)	~~SHOW~~: ×	3,18	
(18)	b = s	15,S11	
(19)	F[s]	13a,17,IL	

T10.

(1)	~~SHOW~~: ∀x{A[x] → fin[x]}	MI (atomic formulas)	
(2)	│ ~~SHOW~~: fin[‘P’]	3, def ‘fin’	BC
(3)	│ │ len(‘P’) = 1	string theory	
(4)	│ ~~SHOW~~: ∀x{fin[x] → fin[x+‘♯’]}	UCD	IC
(5)	│ │ fin[s]	As	IH
(6)	│ │ ~~SHOW~~: fin[s+‘♯’]	6, def ‘fin’	IS
(7)	│ │ let: n = len(s)	5,def fin, ∃O	
(8)	│ │ len(s+‘♯’) = len(s) + len(‘♯’)	string theory	
(9)	│ │ len(‘♯’) = 1	string theory	
(10)	│ │ len(s+‘♯’) = n+1	8,9,IL	

T11.

(1)	~~SHOW~~: ∀x{F[x] & 1^{st}(x) = ‘N’ → ∃y{F[y] & x = ‘N’+y}}	UCD
(2)	│ F[s] & 1^{st}(s) = ‘N’	As
(3)	│ ~~SHOW~~: ∃y{F[y] & s = ‘N’+y}	5,7,QL
(4)	│ │ ∃x[s = ‘N’+x]	2b,ET
(5)	│ │ s = ‘N’+b	4,∃O
(6)	│ │ F[‘N’+b]	2,5,IL
(7)	│ │ F[b]	6,ET