	1	$\exists x.\ P(x)$	
$b\forall\mathcal{I}$	2		
$a\exists\mathcal{E}$	3	$P(a)$	
	4	$\vdots$	
	5	{cannot complete proof}	
	6	$P(b)$	
	7	$P(b)$	$\exists\mathcal{E}(1)$
	8	$\forall x.\ P(x)$	$\forall\mathcal{I}$

Figure 18.3 Failure to show $\exists x.\ P(x) \vdash \forall x.\ P(x)$

	1	$\exists z.\ \top$	
	2	$\forall x.\ \exists y.\ P(x,y)$	
$c\exists\mathcal{E}$	3	$\top$	
	4	$\exists y.\ P(c,y)$	$\forall\mathcal{E}(2)$
$d\exists\mathcal{E}$	5	$P(c,d)$	
$e\forall\mathcal{I}$	6		
	7	$\vdots$	
	8	{cannot fill gap}	
	9	$P(c,e)$	
	10	$\forall v.\ P(c,v)$	$\forall\mathcal{I}$
	11	$\exists u.\ \forall v.\ P(u,v)$	$\exists\mathcal{I}$
	12	$\exists u.\ \forall v.\ P(u,v)$	$\exists\mathcal{E}(4)$
	13	$\exists u.\ \forall v.\ P(u,v)$	$\exists\mathcal{E}(1)$

Figure 18.4 Failure to show $\{\exists z.\ \top, \forall x.\ \exists y.\ P(x,y)\} \vdash \exists u.\ \forall v.\ P(u,v)$

at least two elements — say $\{c,d\}$ with $P(c,d)$ and $P(d,c)$ both true and $P(c,c)$ and $P(d,d)$ both false. Then the premisses are true but the conclusion is false.

On the other hand, after line 5 has introduced d we can use it to deduce $\exists y.\ P(d,y)$, which leads to another $\exists$ to eliminate and so on. The alternative

proof attempt is shown in Figure 18.5. We can write down the constants that

	1	$\exists z.\ \top$	
	2	$\forall x.\ \exists y.\ P(x,y)$	
$c\exists\mathcal{E}$	3	$\top$	
	4	$\exists y.\ P(c,y)$	$\forall\mathcal{E}(2)$
$d\exists\mathcal{E}$	5	$P(c,d)$	
	6	$\exists y.\ P(d,y)$	$\forall\mathcal{E}(2)$
$e\exists\mathcal{E}$	7	$P(d,e)$	
	8	$\vdots$	
	9	{cannot fill gap}	
	10	$\exists u.\ \forall v.\ P(u,v)$	
	11	$\exists u.\ \forall v.\ P(u,v)$	$\exists\mathcal{E}(6)$
	12	$\exists u.\ \forall v.\ P(u,v)$	$\exists\mathcal{E}(4)$
	13	$\exists u.\ \forall v.\ P(u,v)$	$\exists\mathcal{E}(1)$

Figure 18.5 Failure to show $\{\exists z.\ \top, \forall x.\ \exists y.\ P(x,y)\} \vdash \exists u.\ \forall v.\ P(u,v)$

arise in Figure 18.5, with an arrow from x to y whenever $P(x,y)$:

$c \to d \to e \to \cdots$

This suggests an infinite model:

Domain = set of natural numbers $\{0,1,2,3,\cdots\}$
$P(x,y)$ means that $y = x+1$

It is indeed a counter-example. You cannot possibly choose u so that $\forall v.\ P(u,v)$, for you never obtain $P(u,0)$.

18.3 Intended structures

There is often, implicitly, an intended interpretation for the extralogical symbols. For example, the writer of '$\forall x : nat.\ less(zero, s(x))$' quite probably had in mind the interpretation in which the domain is the set of natural numbers, *less* is $<$, s is the successor function and *zero* is the number 0. Intended interpretations allow the possibility of domain-specific deductions that go beyond logic. In Part I of this book most of the arguments were not pure logic — they had intended structures (for example, numbers, lists, etc.)

in mind and freely used known properties of those structures. For instance, in the specific domain of lists we can reason that if $c \in (h{:}t)$ and $c \neq h$ then $c \in t$. Now, this deduction could be made by making the particular facts about lists explicit, such as

$$\forall u, v : *.\ \forall t : [*].\ [u \in (v{:}t) \rightarrow u = v \vee u \in t]$$

Or, we may think of the fact as being part of our stock of information about lists and quote it as the 'reason' for our deduction. The restricted interpretation gives us more powerful deductions.

In the case of program specifications, the pre- and post-conditions usually make clear what is the intended domain and interpretation. So if our specification indicated that the domain was integers, say, we might make use of sentences such as $\forall x : num.\ [x = 0 \vee x < 0 \vee x > 0]$.

We could in principle axiomitize (add extra premisses to constrain the structures to be sufficiently like the intended one) so that the arguments are pure logic, and this is often a good thing to do — it lays bare the logical structure of the mathematics — but we are not so formal. Hence we have used a 'mixture of logic and mathematics'. Natural deduction still helps one to get through the purely logical aspects of the argument.

Of course, any proof we make in pure logic is correct for any interpretation that satisfies the various sentences we have used, not just the particular one we had in mind. And this is really all we can expect, for when trying to show $S \models T$ by showing $S \vdash T$, the natural deduction rules know nothing of interpretations and so cannot be specific about any particular one.

18.4 Equivalences

In Chapter 15 we defined two sentences S and T to be equivalent ($S \equiv T$) if they had the same truth-value as each other in every situation. What we meant, was that

> $S \equiv T$ iff
> in each structure for $\{S, T\}$ S and T are either both true or both false
> that is, $S \leftrightarrow T$ is true in every structure (it is a tautology)
> that is, $S \models T$ and $T \models S$

The last property holds, since, if it is not possible to have S true in any structure of $\{S, T\}$ and T false, or T true and S false, then in any structure which makes S true T must be true, too, and in any structure which makes T true then S must be true, too. Hence $S \models T$ and $T \models S$.

We now take a second look at some quantifier equivalences and see how the important property of equivalent sentences, that they can be substituted for each other in any context, is affected.

In many cases, the same principles as before apply. A constituent of a sentence can be replaced by any other equivalent sentence. For example,

$\neg\forall x.\ P(x) \equiv \exists x.\ \neg P(x)$ and any occurrence of the first sentence can be replaced by the second, or vice versa. So from $S \vee \neg\forall x.\ P(x)$ we can obtain $S \vee \exists x.\ \neg P(x)$. This applies as long as there is no nested reuse of variables, for example, $\forall x.\ \exists x \cdots$, but remember we said we would not allow such forms. (They can always be avoided by renaming variables.)

If you cannot remember a useful equivalence it does not matter, for you can always derive it each time you need it. The only disadvantage is the extra time taken! Several useful quantifier equivalences are given in Appendix B and although most of the equivalences were stated for unqualified quantifiers, qualified quantifiers present no problem and behave quite well. For example, the equivalence above also holds in the form $\neg\forall x : N.\ P(x) \equiv \exists x : N.\ \neg P(x)$.

In any quantifier-free sentence S any subsentence may be replaced by an equivalent sentence without affecting the meaning of S. This is very useful as one form of a sentence may be more convenient than another. For example, $\neg(P \vee Q)$ may not be as useful a sentence form in a natural deduction proof as the equivalent $\neg P \wedge \neg Q$, which can be broken into two smaller pieces, $\neg P$ and $\neg Q$, and $\forall x.\ \neg P(x)$ is almost always more useful than $\neg\exists x.\ P(x)$. Many equivalences, such as those given in Appendix B, once instantiated by replacing F, G etc., by particular sentences, can be used as they stand to replace one side of the equivalence by the other.

The quantifiers $\forall$ and $\exists$ also respect equivalences:

if $F(a) \equiv G(a)$ then
$\forall x.\ F(x) \equiv \forall x.\ G(x)$, and
$\exists x.\ F(x) \equiv \exists x.\ G(x)$

(Exercise 9 asks you to prove this.)

For example, since $(F(a) \wedge G(b)) \equiv (G(b) \wedge F(a))$, $\exists y.\ [F(a) \wedge G(y)] \equiv \exists y.\ [G(y) \wedge F(a)]$ and $\forall x.\ \exists y.\ [F(x) \wedge G(y)] \equiv \forall x.\ \exists y.\ [G(y) \wedge F(x)]$.

In Sections 18.6 and 18.7 we show that $A \models B$ iff $A \vdash B$ and hence we have $A \equiv B$ iff $A \vdash B$ and $B \vdash A$. An equivalence proof is therefore a good way to show $A \vdash B$ — show instead the stronger $A \equiv B$ using equivalences. Reasoning using equivalences can also be a useful way of making progress in a proof. That is, from

$S \equiv S_1$ and $S_1 \equiv S_2$ and $\cdots$ and $S_{n-1} \equiv S_n$

you can deduce $S \equiv S_n$ and hence that $S \vdash S_n$ and $S_n \vdash S$.

Example 18.3 As an example of the use of equivalences we show

$\exists y.\ \forall x.\ [F(x) \wedge G(y)] \equiv \forall x.\ \exists y.\ [F(x) \wedge G(y)]$

and

$\forall x.\ \exists y.\ [F(y) \wedge (G(x) \rightarrow H(y))] \equiv \exists y.\ \forall x.\ [F(y) \wedge (G(x) \rightarrow H(y))]$

In the proofs the particular equivalences used are left to the reader to supply as an exercise.

$\exists y.\ \forall x.\ [F(x) \wedge G(y)] \equiv \exists y.\ [\forall x.\ F(x) \wedge G(y)] \equiv \forall x.\ F(x) \wedge \exists y.\ G(y)$
$\equiv \forall x.\ [F(x) \wedge \exists y.\ G(y)] \equiv \forall x.\ \exists y.\ [F(x) \wedge G(y)]$

$$\begin{aligned}
&\forall x.\exists y.[F(y) \wedge (G(x) \rightarrow H(y))] \equiv \forall x.\exists y.[F(y) \wedge (\neg G(x) \vee H(y))] \\
&\equiv \forall x.\exists y.[(F(y) \wedge \neg G(x)) \vee [(F(y) \wedge H(y))] \\
&\equiv \forall x.[\exists y.[F(y) \wedge \neg G(x)] \vee \exists y.[F(y) \wedge H(y)]] \\
&\equiv \forall x.\exists y.[F(y) \wedge \neg G(x)] \vee \exists y.[F(y) \wedge H(y)]] \\
&\equiv \exists y.\forall x.[F(y) \wedge \neg G(x)] \vee \exists y.[F(y) \wedge H(y)] \\
&\equiv \exists y.[\forall x.[F(y) \wedge \neg G(x)] \vee (F(y) \wedge H(y))] \\
&\equiv \exists y.\forall x.[(F(y) \wedge \neg G(x)) \vee (F(y) \wedge H(y))] \\
&\equiv \exists y.\forall x.[F(y) \wedge (G(x) \rightarrow H(y))]
\end{aligned}$$

Equivalence proofs are very helpful within natural deduction proofs for they allow premisses and conclusions to be rewritten to more useful forms. There are many useful 'half-equivalences', that is, true sentences of the form $A \models B$, and some are shown in Figure 18.6.

1	$\exists x.\ \forall y.\ F(x,y) \models \forall y.\ \exists x.\ F(x,y)$
2	$\forall x.\ F(x) \vee \forall y.\ G(y) \models \forall x.\ [F(x) \vee G(x)]$
3	$\exists x.\ [F(x) \wedge G(x)] \models \exists x.\ F(x) \wedge \exists x.\ G(x)$
4	$\forall x.\ [F(x) \rightarrow G(x)] \models \forall x.\ F(x) \rightarrow \forall x.\ G(x)$
5	$\forall x.\ [F(x) \rightarrow G(x)] \models \exists x.\ F(x) \rightarrow \exists x.\ G(x)$
6	$\forall x.\ [F(x) \leftrightarrow G(x)] \models \forall x.\ F(x) \leftrightarrow \forall x.\ G(x)$
7	$\forall x.\ [F(x) \leftrightarrow G(x)] \models \exists x.\ F(x) \leftrightarrow \exists x.\ G(x)$

Figure 18.6 Useful implications

In particular, if the data contains ϕ, and $\phi \models \varphi$, then φ can be added to the data. Using half-equivalences to replace subsentences is possible but there are some dangers. Exercise 10 considers this.

A natural deduction view of equivalence

Natural deduction gives another view of equivalences. For example, the proof obligations of the two sentences $\forall x.\ [F(x) \rightarrow S]$ and $\exists x.\ [F(x) \rightarrow S]$, which are shown in Figure 18.7, are essentially the same. Here, the proof obligation is to show S from the data $F(c)$, where c is a new constant in the proof. Hence either of the original sentences behaves as a conclusion in a proof essentially in the same way. If you try a similar exercise for other equivalences you will often see that they exhibit the same kind of pattern — the proof obligation for a pair of equivalent sentences is rather similar.

Equivalent sentences, however, also operate in essentially the same way when used as data. For example, if the two sentences $\forall x.\ [F(x) \rightarrow S]$ and $\exists x.\ [F(x) \rightarrow S]$ were part of the data their use would lead to the fragments shown in Figure 18.8. Here, the proof obligations amount to showing $F(a)$ for some a in the current context. These examples, although not a proof, should help to convince you that equivalent sentences often 'behave in a natural deduction proof in the same kind of way'.

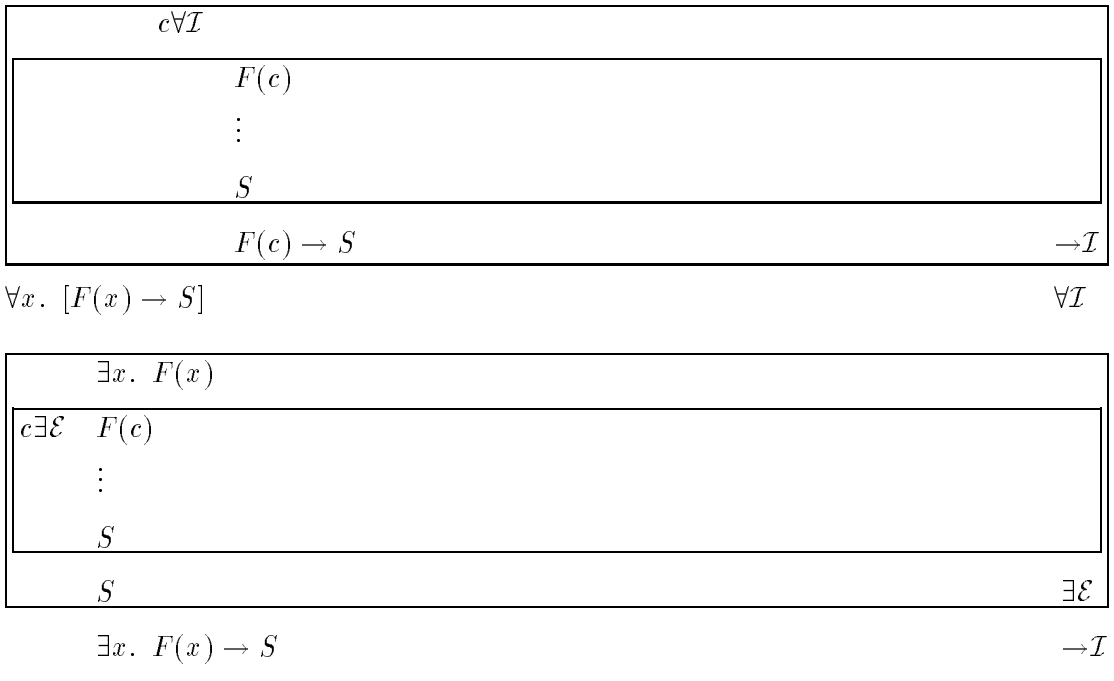

Figure 18.7

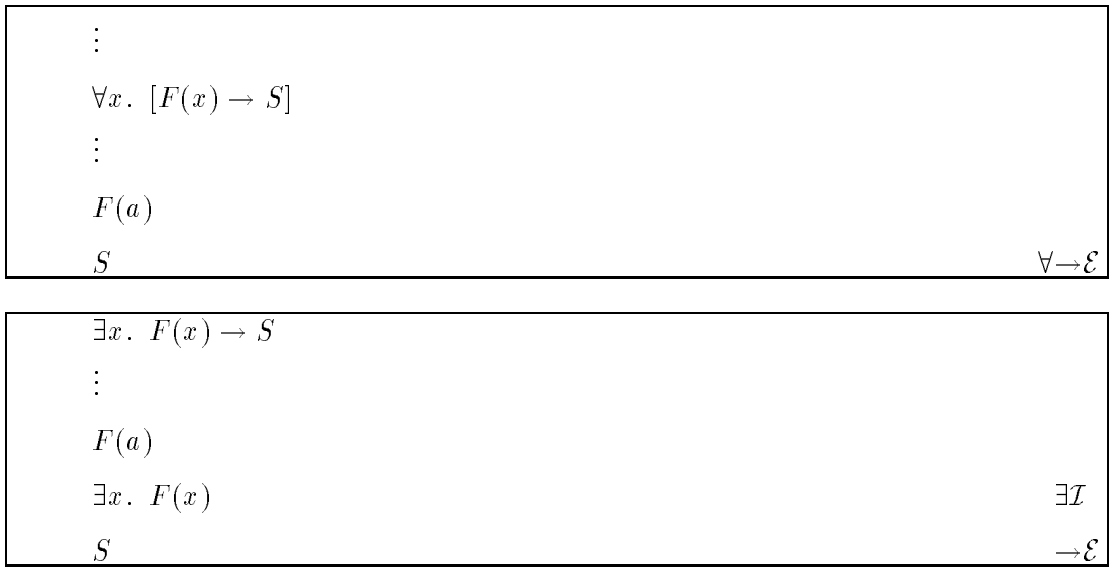

Figure 18.8

18.5 Soundness and completeness of natural deduction

In this section we consider the two important properties of natural deduction, *soundness* and *completeness*.

One of the uses of natural deduction is as a technique for showing that $S \models T$ for sentences S and T. It is successful mainly because natural deduction is *sound*:

If $S \vdash T$ then $S \models T$

This is obviously a necessary property, otherwise all manner of sentences T might be shown to be proven from S regardless of any semantic relationship, and natural deduction would be useless.

At least, therefore, we can be sure that natural deduction proofs are correct. But there could still be a problem. Perhaps, for a particular pair of sentences $S1$ and $S2$, we cannot seem to find a proof. We may ask whether we have enough natural deduction rules to make a deduction. Well, in fact we do, because of *completeness*:

If $S \models T$ then $S \vdash T$

So we know there should be a proof.

Since we probably do not happen to know whether or not $S1 \models S2$, and hence whether or not a deduction should be possible or not, then it might be worth looking for a counter-example model if our proof attempts were floundering. Completeness is not such a crucial property as soundness — for it might be good enough in practice to be able to find a proof in most of the cases for which we expect to find one.

Natural deduction is just one method that can be used to answer the problem 'does $P \models C$'; and there are other methods which are not considered in this book. But natural deduction cannot be used to answer the question 'does $P \not\models C$'.

We say that a problem with the property that there is some method which can always decide correctly between 'yes' and 'no' answers is *decidable*. In our problem is there some method that, given P and C, always tells you 'yes' when $P \models C$ and 'no' when $P \not\models C$? In this case, there is no method that will *always* give the correct answer. Some methods may, like natural deduction, always answer *yes* correctly, and may even be able to answer *no* correctly for some cases, but no method can answer correctly in all cases. The problem, then, of checking whether $P \models C$ is called *semi-decidable*. A *decidable* problem would be one for which a method existed which correctly 'answered' both yes and no type questions.

The problem of checking whether $P \models C$ when P and C are propositional *is* decidable, for then a method that checks all interpretations for the symbols in $\{P, C\}$ is possible and is essentially the method of truth tables.

18.6 Proof of the soundness of natural deduction

In this section the important soundness property of Natural Deduction is proved:

$$\text{if } A \vdash B \text{ then } A \models B \qquad \textit{soundness}$$

that is, if a conclusion B is derivable from premisses A then it *should be* — the argument is valid.

The underlying idea is quite simple: when you read a proof from top to bottom (*not* jumping backwards and forwards in the way that it was constructed) you see a steady accumulation of true sentences. Each new one is justified on the basis that the preceding ones used as premisses by the rule you are applying have themselves already been proved and so are true. (This is an induction hypothesis! It uses induction on the length of the proof, because the earlier sentences were proved using shorter proofs. Also, disregard the fact that some parts of the proof are written out side by side — rearrange them one after the other.)

For instance: consider $\wedge\mathcal{E}$. If you have already proved $A \wedge B$, by induction you know that it is true (given the premisses) and it follows — check the truth tables if you are really in doubt — that the A delivered by $\wedge\mathcal{E}$ is also true.

This is the basic idea, but it is all made much more complicated by the boxes. The problem is that 'true' here means 'true in every model of the premisses', but the class of models varies throughout the proof. Each sentence A appearing in the proof is proved in a *context* of constants and premisses: the constants are not only those posed in the question (by being mentioned in the overall premisses and conclusion), but are also those introduced by $\forall\mathcal{I}$ or $\exists\mathcal{E}$ at the tops of boxes containing A; and the premisses are not only the overall premisses but are also the assumptions introduced for $\vee\mathcal{E}$, $\rightarrow\mathcal{I}$, $\neg\mathcal{I}$ or $\exists\mathcal{E}$ at the tops of boxes containing A.

What you introduce as a new constant or a new assumption at the top of a box is part of the context of everything inside the box.

To take proper account of both premisses and context, we shall, for the time being, use more refined notions of models and semantic entailment ($\models$). A model for a *context* (S, P) (S the set of constants, P the set of premisses; the constants in sentences in P must all be in S) is a model for P with interpretations given for all constants in S. Then we write $P \models_S C$ to mean that C is true in every model of (S, P).

Note the following: if (S', P') is a *bigger* context than (S, P) — all the constants and premisses from (S, P) and possibly some more — then any model of (S', P') is also a model of (S, P). (EXERCISE: prove this.) It follows that if $P \models_S C$ then $P' \models_{S'} C$. This is a technical explanation of why in a proof we are allowed to import sentences into boxes (smaller context to bigger), but not to export them out of boxes.

The basic result, proved by induction on the length of the proof, is this:

if natural deduction proves C in context (S, P), then $P \models_S C$.

A proof of length 0 is one that simply repeats an assumption (that is, the conclusion C is in P) and we have shown that this is always allowed from a smaller context into a larger one. Clearly, $P \models_S C$ in that case.

Let us see first how the $\wedge\mathcal{I}$ rule works, as it is typical of the rules that do not involve boxes (the boxes used for it are purely decorative, because they do not introduce new assumptions or constants). Suppose $A \wedge B$ is proved in the context (S, P). The rule relies on having proved A and B earlier, possibly in smaller contexts (and imported), so by induction we have $P \models_S A$ and $P \models_S B$. We want to prove $P \models_S A \wedge B$, so consider any model of (S, P). In it we know that both A and B are true, so $A \wedge B$ must be as well (again, use the truth tables if you do not believe this).

The reasoning is really just the same for $\wedge\mathcal{E}$, $\vee\mathcal{I}$, $\rightarrow\mathcal{E}$, $\neg\mathcal{E}$, $\bot\mathcal{E}$, $\neg\neg$, $\forall\mathcal{E}$ and $\exists\mathcal{I}$. We can safely leave most of these as exercises, but let us look at a few of the more subtle ones.

$\bot\mathcal{E}$ Suppose A is proved in the context (S, P) by $\bot\mathcal{E}$, so we already have $P \models_S \bot$. This means that in any model of (S, P), false is true — but that is impossible, so we conclude that there are *no* models of (S, P). Hence in *all* of them A is true, so $P \models_S A$.

$\forall\mathcal{E}$ We have $\forall x.\ A(x)$ in the context (S, P), and also t is a term in the context — that is to say it is built up from the function symbols provided and the constants in S. In any model of (S, P), those ingredients are all interpreted, and so t is interpreted as a value of the model. But $\forall x.\ A(x)$ is true in the model, that is, $A(v)$ is true for all possible values v, and in particular the $A(t)$ delivered by the rule is true.

$\exists\mathcal{I}$ This case is rather similar to $\forall\mathcal{E}$ and is left as an exercise.

We now turn to those rules that really do use boxes.

$\vee\mathcal{E}$ The rule gives C in a context (S, P), and we have already proved $A \vee B$ so we know $P \models_S A \vee B$. We have also already proved C twice *but in larger contexts*: once in a box headed by the assumption A — so the context is $(S, P \cup \{A\})$ — and once with B. From these what we know is that $P, A \models_S C$ and $P, B \models_S C$. We want $P \models_S C$, so consider a model of (S, P). $A \vee B$ is true in it, so we have either A true or B true. It follows that the model is also a model either of $(S, P \cup \{A\})$ or of $(S, P \cup \{B\})$, and in either case we can deduce that C is true. (Of course, this argument is just a formalization of the idea of case analysis by which we originally justified the rule.)

$\rightarrow\mathcal{I}$ The rule gives $A \rightarrow B$ in a context (S, P) when we have already proved B in the larger context $(S, P \cup \{A\})$ and hence know $P, A \models_S B$. Consider

a model of (S, P). If A is false in it, then $A \to B$ is certainly true, whilst if A is true then it is also a model of $(S, P \cup \{A\})$ so that B, and hence also $A \to B$, are true.

$\neg\mathcal{I}$ The rule gives $\neg A$ in context (S, P) when we have already proved $\bot$ in a context $(S, P \cup \{A\})$ and hence know $P, A \models_S \bot$; in other words, there are no models of $(S, P \cup \{A\})$. A model of (S, P) cannot be a model of $(S, P \cup \{A\})$, so A must be false — $\neg A$ is true.

$\forall\mathcal{I}$ The rule gives $\forall x.\ A(x)$ in a context (S, P) when we have already proved $A(c)$ in a context $(S \cup \{c\}, P)$ and hence know that $P \models_{S\cup\{c\}} A(c)$. Consider a model of (S, P): we want to know that $A(v)$ is true for every possible v. But for any particular value v we can make the model into one for $(S \cup \{c\}, P)$ by interpreting c as v (note that c had to be a *new* constant, for otherwise c would already be interpreted as something else): then we know that $A(c)$, that is, $A(v)$, is true.

$\exists\mathcal{E}$ The rule gives B in a context (S, P) when we have already proved $\exists x.\ A(x)$ in the same context and have proved B in the context $(S \cup \{c\}, P \cup \{A(c)\})$. In any model of (S, P) we know that there is at least one value v such that $A(v)$ is true; if we pick one, then we can make the model into one of $(S \cup \{c\}, P \cup \{A(c)\})$ by interpreting c as v (again, c must be new); but then we deduce that B is true.

□

18.7 Proof of the completeness of natural deduction

In this section we give a proof of the completeness property for propositional sentences and outline the changes needed for quantifier sentences.

Our method is a traditional one but, as you will see, it does not seem to be fully in the spirit of Natural Deduction, for although it shows that a deduction of B from A exists when $A \models B$, the method does not show how to construct such a proof. Moreover, the proof that *is* guaranteed to exist is also rather contrived. There are other, constructive, methods, but they are beyond the scope of this book.

Theorem 18.4 *completeness* If $A \models B$ then $A \vdash B$, that is, if an argument is valid then the conclusion *can* be derived from the premisses.

Proof : First some definitions:

A set of sentences A is *inconsistent* iff $A \vdash \bot$.
A set of sentences A is *consistent* iff it is not inconsistent.

To show $A \vdash B$, we have to show Proposition 18.5:

if A is a consistent set of sentences then A has a model.

We can then argue:

> If $A \models B$ then $A \cup \{\neg B\}$ has *no* models. (Why?)
> Hence $A \cup \{\neg B\}$ cannot be consistent (by Proposition 18.5).
> Hence $A \cup \{\neg B\}$ is inconsistent.
> Hence $\{A, \neg B\} \vdash \bot$.
> Hence $A \vdash B$ by $\neg I$ and $\neg\neg$.

□

Notice that in the penultimate step the existence of a natural deduction proof is asserted but there are no means given to help you to find it.

We will first deal with a simple case in which the only logical symbols allowed in A are $\wedge, \vee$ and $\neg$ (called $\wedge$-$\vee$-$\neg$ form) and all negations are immediately before a proposition symbol or another negation. In the case when the sentences in $A \cup \{\neg B\}$ are in $\wedge$-$\vee$-$\neg$ form the natural deduction proof of $\bot$ will be one in which $\vee\mathcal{E}$, $\wedge\mathcal{E}$ and $\neg\mathcal{E}$ are used exclusively. In Exercise 3 you have a chance to find such a proof. This does not mean that the other rules are unnecessary, for, as Exercise 8 shows, they are all used in deriving the $\wedge$-$\vee$-$\neg$ form of a sentence by natural deduction.

Proposition 18.5 If A is a consistent set of sentences then there is some model for it.

Proof : The idea is to construct a larger set of consistent sentences, called A^+, that includes A and for which we can give a model. This model will be a model for A as well.

The construction of A^+ from A uses the rules given below:

1. $A^+ \supseteq A$

2. if $A1 \wedge A2 \in A^+$ then $A1 \in A^+$ and $A2 \in A^+$

3. if $A1 \vee A2 \in A^+$ then $A1 \in A^+$ or $A2 \in A^+$

4. if $\neg\neg A1 \in A^+$ then $A1 \in A^+$

Nothing else belongs to A^+ apart from the sentences forced to do so by (1)—(4). A^+ is constructed by applying the rules above to A until they can be applied no more, choosing in step (3) whichever of $A1$ or $A2$ will maintain consistency.

A^+ is consistent

Rule (2) obviously preserves consistency: if you could prove $\bot$ using $A1$ and $A2$ then you could also prove it without them using $A1 \wedge A2$ and $\wedge\mathcal{E}$. And what about rule (3)? The point is that you have at least one option that preserves consistency. For if you can deduce $\bot$ using $A1$ and you can also

deduce it using $A2$ then by $\vee\mathcal{E}$ you could also deduce $\bot$ using $A1 \vee A2$. Rule (4) is left for you to deal with.

An example

$A = \{((\neg P \vee Q) \wedge P) \vee \neg P\}$
$A^+ \supseteq \{((\neg P \vee Q) \wedge P) \vee \neg P\}$ (rule 1)
$A^+ \supseteq \{((\neg P \vee Q) \wedge P) \vee \neg P, (\neg P \vee Q) \wedge P\}$ (rule 3)
$A^+ \supseteq \{((\neg P \vee Q) \wedge P) \vee \neg P, (\neg P \vee Q) \wedge P, P, \neg P \vee Q\}$ (rule 2)
$A^+ \supseteq \{((\neg P \vee Q) \wedge P) \vee \neg P, (\neg P \vee Q) \wedge P, P, \neg P \vee Q, Q\}$ (rule 3)

All the sentences have now been dealt with and to find a model of A^+ just look at the atoms or their negations in A^+, in this case P and Q. The assignment $Q = tt, P = tt$ is a model, as you can check.

This is not the only consistent set that can be constructed by applying the rules. Another one is

$A = \{((\neg P \vee Q) \wedge P) \vee \neg P\}$
$A^+ \supseteq \{((\neg P \vee Q) \wedge P) \vee \neg P\}$ (rule 1)
$A^+ \supseteq \{((\neg P \vee Q) \wedge P) \vee \neg P, \neg P\}$ (rule 3)

This time, $\neg P$ was chosen from $((\neg P \vee Q) \wedge P) \vee \neg P$ to satisfy the third rule. You can check that the assignment $P = ff$ and $Q = ff$ is also a model of A^+ and A.

(Since A^+ is consistent it cannot contain C and $\neg C$ for any C. Why?)

A^+ has a model

We now show that A^+ has a model I (say). For each proposition symbol X used in sentences in A^+:

If $X \in A^+$ then X is assigned tt in I.
If $\neg X \in A^+$ then $X = ff$ in I.
If $X \notin A^+$ and $\neg X \notin A^+$ then X is assigned ff in I.

I *is* a model of A^+:

Suppose not, and that Y in A^+ is the smallest sentence in A^+ that is not true in I.

(Use the ordering: a proposition symbol and its negation are the smallest sentences; the constituents of a sentence are smaller than it; so A is smaller than $A \wedge B$, etc.)

Y could be an atom? No, as Y would have been assigned tt.
Y could be $\neg Y'$, Y' an atom? No, as Y' would have been assigned false in I and so $\neg Y'$ is true in I.
Y could be $A1 \wedge B1$ or $A1 \vee B1$? No, as either $A1$ or $B1$ (or both) would be false in I and both are smaller than Y, the supposed smallest false sentence.

Y could be $\neg\neg A1$? No, as $A1$ would have been in A^+, too, and also false in I.

Since I is a model for A^+ it is a model for A. □

If A and B are general propositional sentences then Proposition 18.5 can still be used. It does not matter if you replace A by an equivalent set of sentences A': A is consistent iff A' is consistent. Any propositional sentence A is equivalent to one in the $\wedge$-$\vee$-$\neg$ form used in Proposition 18.5 and the $\wedge$-$\vee$-$\neg$ form can be deduced by natural deduction from A and vice versa (see Exercise 8). So every sentence in $A \cup \{\neg B\}$ can be replaced by an equivalent sentence in $\wedge$-$\vee$-$\neg$ form before applying Proposition 18.5.

What has been proved here is often called *weak completeness*. That is, it simply shows that a natural deduction proof exists. But suppose you are trying to derive a sequent and do not follow this 'correct' path (as given by the theorem), whatever it is. You want to know that under reasonable circumstances, the conclusion can still be derived. This is indeed the case, but showing it is belongs to the realm of automated deduction.

Completeness for quantifier sentences

The proof method for propositional sentences can be extended to quantifier sentences as outlined next. Suppose that the problem is to show $A \vdash B$. The construction of A^+ has to be extended so that it includes sentences prefixed by a quantifier. Initially, the context of A^+ is just the context of A, S say. The rule for dealing with $\exists$ will increase this context and so the final context of A^+ will not, in general, be the same as the context of A. We have to take this into consideration when showing that the $\exists$ rule maintains consistency of A^+.

The rules for constructing A^+ now include

5. If $\forall x.\ P[x] \in A^+$ then $P[a] \in A^+$, for all a formed from symbols in the current context S of A^+.
6. If $\exists x.\ P[x] \in A^+$ then $P[e] \in A^+$, for a new constant $e \notin S$. The context is updated to $S \cup \{e\}$.

We can show that rules (5) and (6) maintain consistency:

5. A' is the result of the construction so far, $\forall x.\ B[x] \in A'$ and A' is consistent. $A' \cup \{B[t/x]\}$ is consistent, where t is a term constructed from symbols in the context S' of A'. If not, a proof of $A' \cup \{B[t/x]\} \vdash \bot$ could be converted to a proof of $A' \vdash \bot$ by an additional use of $\forall\mathcal{E}$, giving a contradiction.
6. A' is the result of the construction so far, S' is the context so far and $\exists x.\ B[x] \in A'$ and A' is consistent. $A' \cup \{B[e/x]\}$ is consistent, where

e is a new constant $\notin S'$. If not, a proof of $A' \cup \{B[e/x]\} \vdash \bot$ could be converted to a proof of $A' \vdash \bot$ by using $\exists\mathcal{E}$, which would then be contradictory.

The construction of A^+ will be an infinite process unless there are no function symbols in A (because of step (5)).

Finally, we have to show that the model formed by considering atoms and their negations in A^+ is still a model of A^+. The atoms we consider are all atoms formed from predicates in A and terms using symbols in the final context S^+ of A^+. The domain of the interpretation I is just the set of terms formed from symbols in S^+ and each term is interpreted by itself.

The additional cases cover Y being either of the form $\exists x.\ P[x]$ or $\forall x.\ P[x]$:

Y could be of the form $\forall x.\ P[x]$? No, as then some sentence of the form $P[t/x]$ would also be false and this is smaller than Y.

Y could be of the form $\exists x.\ P[x]$? No, as then every sentence of the form $P[d/x]$ would be false, where $d \in$ domain of I. In particular, $P[e/x]$ would be false, a contradiction as this is smaller than Y.

18.8 Summary

- A *signature* is a collection of extralogical symbols (predicates, functions and constants) with their arities.
- A *structure* (for a signature or for some sentences) gives concrete *interpretations* for those symbols as relations, functions or elements from some particular set, the *domain*.
  Once this is done, any sentence using those symbols is interpreted and it can be determined whether it is true or false.
- A *model* for a sentence is a structure in which the sentence is true.
- The 'failed natural deduction by counter-example' technique can be used to show that $P \nvdash C$.
- Intended interpretations correspond to extralogical deductions.
- Quantifier equivalences can be applied to transform sentences.
- Natural deduction is *sound*:

  If $P \vdash C$ then $P \models C$

- Natural deduction is *complete*:

  If $P \models C$ then $P \vdash C$

18.9 Exercises

1. (a) If $A \vdash B$ then $A \models B$ (soundness of natural deduction). Hence, if $A \not\models B$ then ...?
   (b) If $A \not\models B$ does $A \models \neg B$?
   (c) If $A \models B$ does $A \not\models \neg B$?
   (d) If $A \not\vdash B$ what about $A \vdash \neg B$?
   (e) If $A \not\vdash \neg B$ does $A \vdash B$?
   (f) If $\{S1, S2, \ldots, S_n\} \models T$ is valid does $\{S2, \ldots, S_n\} \not\models T$?
   (g) If S is true in no situations then $\neg S$ is true in every situation. True or false?
2. Complete the missing cases in the proof of soundness of Natural Deduction given in Section 18.5.
3. (a) Apply the method used in the completeness proof to derive a model of the sentences $\{C \wedge N \rightarrow T, H \wedge \neg S, (H \wedge \neg(S \vee C)) \rightarrow P, N, \neg P\}$. First convert the sentences to the restricted form using equivalences and then apply the method.
   (b) Find a natural deduction proof of $\perp$ from the converted sentences.
4. Show that the following arguments are not valid, that is, the premisses $\not\models$ the conclusion. Find two structures in each case in which the premisses are true but the conclusion false. Try the 'failed natural deduction by counter-example' technique in order to help you to find the structures:
   (a) $likes(Mary, John), \forall x.\ [likes(John, x)] \not\models \neg\exists y.\ \neg(likes(Mary, y))$.
   (b) $\neg\forall x.\ \forall y.\ [Diff(x, y) \wedge R(x, y) \rightarrow R(y, x)] \not\models \forall u.\ \forall v.\ [Diff(u, v) \wedge R(u, v) \rightarrow \neg R(v, u)]$.
   (c) $\forall x.\ [F(x) \vee G(x)] \not\models \forall x.\ F(x) \vee \forall y.\ G(y)$.
   (d) $\exists v.\ F(v) \wedge \exists u.\ G(u) \not\models \exists x.\ [F(x) \wedge G(x)]$.
   (e) $\forall x.\ \exists y.\ M(x, y) \not\models \exists v.\ \forall u.\ M(u, v)$.
5. For each structure and each set of sentences decide the truth/falsity of the sentences in the structure:
   (a) $\{\forall x.\ R(x, x), \forall x.\ \forall y.\ [R(x, y) \rightarrow R(y, x)]\}$ Structures:
      i. $D = \{a, b, c\}$, $R(a, b) = R(a, c) = R(b, c) = R(c, b) = tt$, $R(a, a) = R(b, b) = R(c, c) = R(b, a) = R(c, a) = ff$
      ii. $D = \{1, 2, 3, 4, \ldots\}$, R is the relation $<$
      iii. $D = \{1, 2, 3, \ldots\}$, R is the relation $divides(x, y)$
   (b) $\{\forall x.\ \exists y.\ [P(x) \rightarrow Q(x, y)],\ \exists z.\ P(z),\ \exists z.\ [Q(b, z) \rightarrow \forall u.\ P(u)]\}$ Structures:
      i. $D = \{1, 2, 3, \ldots\}$, b is the number 2, $P(x)$ is the relation x is even, $Q(x, y)$ is the relation $divides(x, y)$

ii. $D = \{Fred,\ Susan,\ Mary\}$, b is $Mary$, $P(Fred) = Q(Mary, Fred) = Q(SusanFred) = tt$, $P(Susan) = P(Mary) = ff$, all other pairs for $Q = ff$

(c) $\{\exists z.\ \forall u.\ P(f(u), z)\}$ Structures:

i. $D = \{0,\ 1,\ -1,\ 2,\ -2,\ \ldots\}$, P is the relation $<$, f is the function: $f(u) = |\ u\ |$

ii. $D = \{1,\ 2,\ 3,\ \ldots\}, P$ is the relation $<$, f is the successor function.

6. Find as many different models as you can for the sentences: $\{\forall x.\ \forall y.\ \forall z.\ [P(x, y, z) \rightarrow P(s(x), y, s(z))], \forall x.\ P(a, x, x)\}$
7. Decide on the truth values of the sentences of Example 18.2 in the structure with domain= $\{0,\ \pm 1,\ \pm 2,\ \cdots\}$ and in which A means 0, $P(n)$ means $n \geq 0$, and $Q(m, n)$ means $m^2 = n$.
8. The completeness proof for propositional sentences given in the text can be extended to include all logical operators by using the fact that the following (ND) equivalences can be found:

$$\neg(A \wedge B) \equiv \neg A \vee \neg B \qquad \neg(A \vee B) \equiv \neg A \wedge \neg B$$
$$\neg(A \rightarrow B) \equiv A \wedge \neg B \qquad A \rightarrow B \equiv \neg A \vee \neg B \qquad \neg\neg A \equiv A$$

That is (for example), $A \rightarrow B \vdash \neg A \vee B$ and $\neg A \vee B \vdash A \rightarrow B$.

(a) Prove each of the above (ND) equivalences.

(b) Once you have proofs of the equivalences they can be used to rewrite any sentence into $\wedge$- $\vee$ -$\neg$ form. The A and B can be any sentences. In particular, prove that if $A \vdash B$, $B \vdash A$, $A' \vdash B'$ and $B' \vdash A'$ then

$\neg A \vdash \neg B$ and $\neg B \vdash \neg A$
$A \wedge A' \vdash B \wedge B'$ and $B \wedge B' \vdash A \wedge A'$
$A \vee A' \vdash B \vee B'$ and $B \vee B' \vdash A \vee A'$
$A \rightarrow A' \vdash B \rightarrow B'$ and $B \rightarrow B' \vdash A \rightarrow A'$

9. Show that quantifiers respect equivalences. That is, if $A(a) \equiv B(a)$ for sentences A and B and some constant a, then $\forall x.\ A(x) \equiv \forall x.\ B(x)$ and $\exists x.\ A(x) \equiv \exists x.\ B(x)$. (HINT: use induction on the structure of A and B.)
10. We say that A occurs *positively* in a sentence F if it is within an even number (or zero) of negations. It occurs *negatively* otherwise. Show that, if A occurs positively in a sentence F and $A \models B$ and replacing A by B in F gives G, then $F \models G$. Also, show that if A occurs negatively in F then $G \models F$.

Appendix A

Well-founded induction

Find a simplest counter-example

One justification for induction arguments is that they say

1. Find a simplest possible counter-example: in other words, all simpler possibilities work correctly.
2. But then from that we manage to deduce that the counter-example, too, works correctly — it is not a counter-example at all.
3. Contradiction: so there are no counter-examples.

(3) is just logic, and (2) depends entirely on the problem to hand (what we are trying to prove). It is the induction step. But (1) depends not so much on what we are trying to prove, as on the things we are proving something about: it says that there is some notion of 'simplicity', and that we can indeed find a simplest. For instance, for numbers, 'simpler' might be 'less than'. Then finding a smallest number is something you can always do with sets of natural numbers but not necessarily with sets of integers or reals.

Well-founded orderings

Suppose we are interested in proving 'by induction', that is, using (1)—(3) above, statements of the form $\forall x : A.\ P(x)$, where A is some set such as *nat*. We formalize the idea of simplicity with the notion of *well-founded ordering*.

Definition A.1 Let A be a set, and $<$ a binary relation on A. $<$ is a *well-founded ordering* iff every non-empty subset X of A has a minimal element, that is, some $x \in X$ such that if $y < x$ then $y \notin X$.

Note that although $<$ is called an ordering, there is no requirement for it to be transitive or to have any other of the usual properties of orderings.

Theorem A.2 Let A be a set and $<$ a binary relation on A. Then the following are equivalent:

1. $<$ is a well-founded ordering.
2. A contains no infinite descending chains $a_1 > a_2 > a_3 > \ldots$
   (Of course, $a > b$ means $b < a$.)
3. (*Principle of well-founded induction*) Let $P(x)$ be a property of elements of A such that for any $a \in A$, if P holds for every $b < a$ then P also holds for a. Then P holds for every a.

Proof

1 $\Longrightarrow$ **3** (This is really an abstraction of the induction idea presented informally above. The condition on P is the formalization of the step finding that the counter-example is not a counter-example.) Let P be a property as stated, and let X be the set $\{x \in A : \neg P(x)\}$. If $X \neq \varnothing$ then by well-foundedness there is a minimal element a in X ('a simplest counter-example'). For any $b < a$ we have $b \notin X$, so $P(b)$ holds; hence by the conditions on P we have $P(a)$, which contradicts $a \in X$. The only way out is that $X = \varnothing$, that is, $P(a)$ for all a.

2 $\Longrightarrow$ **1** Choose $a_1 \in X$ (possible, because $X \neq \varnothing$). If a_1 is minimal in X, then we are done; otherwise, we can find $a_1 > a_2 \in X$. Again, either a_2 is minimal or we can find $a_2 > a_3 \in X$. We can iterate this, and it must eventually give us an element minimal in X, because otherwise we would obtain an infinite descending chain, contradicting (2).

3 $\Longrightarrow$ **2** Let $P(x)$ be the property 'there is no infinite descending chain starting with x'. Then P satisfies the condition of (3), and so P holds for every a. Hence there are no infinite descending chains at all. □

These three equivalent conditions play different conceptual roles. (1), as in the definition of well-foundedness, is the direct formalization of the ability to 'find simplest counter-examples'. (2) is usually the most useful way of checking that some relation $<$ is well-founded, and (3) is the logical principle.

Box proofs

We can put the induction principles into natural deduction boxes. This is not so much because we want to formalize everything, as to show the proof obligations, the assumptions and goals when we use induction.

The general principle of well-founded induction, given a set A and a well-founded ordering $<$, is shown in Figure A.1.

Figure A.1

The box, with the piece of proof that you have to supply, is the *induction step*. The formula labelled (IH) is the *induction hypothesis*, and it is a valuable free gift. If it weren't there, then the proof would just be ordinary $\forall\mathcal{I}$ introduction and the goal in the box would be more difficult (or impossible). We shall now look at examples of well-founded orderings, with their corresponding induction principles.

nat

This is the most basic example. You cannot have an infinite descending sequence of natural numbers, so the ordinary numeric ordering $<$ is well-founded. Figure A.2 gives the principle of *course of values induction*:

Figure A.2

A variant on this is obtained by taking $<$ to be not the ordinary numeric order, but a different relation defined by m '$<$'n if $n = m + 1$. Then the induction hypothesis is $\forall m : nat(n = m + 1 \to P(m))$, which works out in two different ways according to the value of n. If $n = 0$, it is vacuously true — there are no natural numbers m for which $0 = m + 1$. If $n > 1$, the only possible m is $n - 1$, and so it tells us $P(n - 1)$. Separating these two cases out, and in the second case replacing m by $n - 1$, we obtain in Figure A.3 the principle of *simple induction*.

It is no coincidence that these two boxes (the base case and the induction step) correspond to the two alternatives in the datatype definition for natural

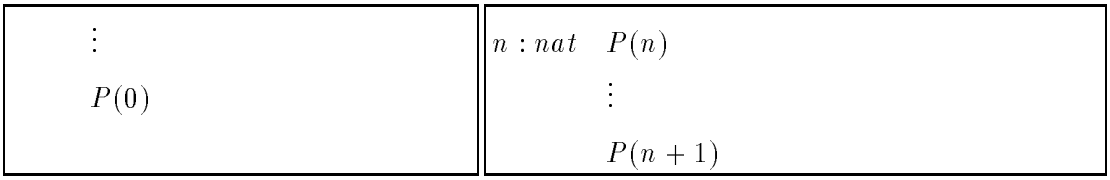

Figure A.3

numbers:

```
num ::= 0 | suc num
```

Note two *non*-examples of well-founded orderings.

1. The integers under numeric $<$: for there are infinite descending chains such as

   $$0 > -1 > -2 > -3 > \ldots$$

2. The positive rationals under numeric $<$:

   $$1 > 1/2 > 1/3 > 1/4 > 1/5 > \ldots$$

Recursion variants

Let A be any set, and $v : A \to nat$ any function. Then we can define a well-founded ordering $<$ on A by

$x < y$ iff $v(x) < v(y)$ (numerically)

The induction principle is given in Figure A.4.

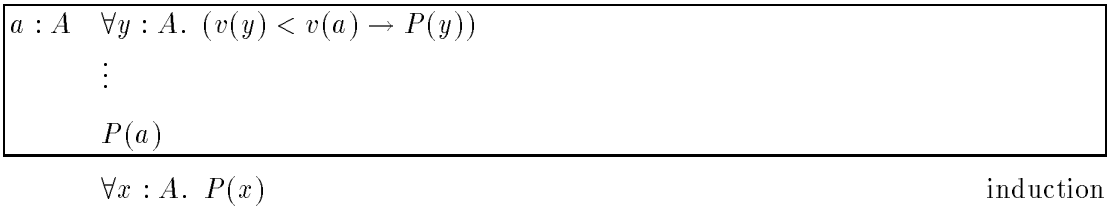

Figure A.4

This is course of values induction 'on v'. Plainly *nat* here could be replaced by any other set with a well-founded ordering. The programming examples