derived sentences. It is also often used in a backward direction, in which case some ingenuity is needed. Suppose a sentence $\neg P$ is already derived, and $\bot$ is required, for example to use $\neg\mathcal{I}$, then the $\neg\mathcal{E}$ rule requires P to be derived in order to obtain $\bot$. Thus P becomes the new conclusion.

$\neg A$ can be equivalently written as $A \to \bot$, and then the $\neg\mathcal{I}$ and $\neg\mathcal{E}$ rules become special cases of the $\to\mathcal{I}$ and $\to\mathcal{E}$ rules.

In the next example all three negation rules are used!

Show $\vdash A \vee \neg A$

1	$\neg(A \vee \neg A)$	
2	A	
3	$A \vee \neg A$	$\vee\mathcal{I}(2)$
4	$\bot$	$\neg\mathcal{E}(1,3)$
5	$\neg A$	$\neg\mathcal{I}$
6	$A \vee \neg A$	$\vee\mathcal{I}(5)$
7	$\bot$	$\neg\mathcal{E}(1,6)$
8	$\neg\neg(A \vee \neg A)$	$\neg\mathcal{I}$
9	$A \vee \neg A$	$\neg\neg(8)$

Figure 16.14 $\vdash A \vee \neg A$

In Figure 16.14 the crucial step is to realize that $A \vee \neg A$ will follow from $\neg\neg(A \vee \neg A)$. Some ingenuity is again needed at lines 5 and 6 in deciding that to prove $A \vee \neg A$ it is appropriate to show $\neg A$.

The $\neg\neg$ rule is obviously valid. For $\neg\mathcal{E}$, notice that a proof of P and of $\neg P$ gives $P \wedge \neg P$, which is always false. For $\neg\mathcal{I}$, we have to show that P must be false — well, it must be if P leads to a contradiction, $\bot$, for otherwise $\bot$ would have to be true, which it cannot be.

Using boxes to structure proofs

Boxes are used in the natural deduction rules to structure a proof; initially, any data that is given is placed at the top of the proof and the conclusion is placed at the bottom. As a proof progresses, the gap in between is gradually filled up, sometimes working downwards from the top as in $\wedge\mathcal{E}$, $\to\mathcal{E}$ or $\vee\mathcal{E}$, and sometimes working upwards from the bottom as in $\wedge\mathcal{I}$, $\vee\mathcal{I}$ or $\neg\neg$. Many of the steps are automatic, for example, $\to\mathcal{I}$, and only require

some preparation, in the form of some more boxes perhaps. Non-automatic steps, for example, $\vee\mathcal{I}$, cause more problems as they require insight and if the correct step is not seen the proof may not be found.

As boxes are introduced, the available sentences within each box will vary. Initially, only the initial data are available. Inside boxes additional sentences are also available if they are assumptions made when the box is formed; for example, in $\rightarrow\mathcal{I}$ to show $A \rightarrow B$, A is such an assumption. The structure imposed by boxes also means that any derived sentences that occur in a proof above a box X may be used within X, for their proof only required assumptions that are also available within X.

The system of box deductions is a very formal way of writing proofs; the finished product can be read from top to bottom but it gives no clue as to how the proof was derived. Doing the proof with proof boxes allows you to be more confident that your argument is correct. Eventually, you will be able to derive correct arguments every time and dispense with the explicit use of proof boxes, as is done in the majority of proofs in this book.

Derived rules

A tautology, such as $P \vee \neg P$, is a sentence that is always true. It can be derived as in Figure 16.14 using no data, and is also called a *theorem*. Theorems can be used anywhere in a proof if they are needed. Suppose you have derived the theorem $\neg(A \vee B) \rightarrow \neg A \wedge \neg B$, then, if the sentence $\neg(A \vee B)$ appears in a proof, the theorem can be used to derive, by $\rightarrow\mathcal{E}$, $\neg A \wedge \neg B$, which may be a more useful form.

When $\vdash \neg(A \vee B) \rightarrow \neg A \wedge \neg B$ is derived, A and B can be any sentences and the theorem is a *scheme* — any instance of the form of the scheme, obtained by substituting any sentences throughout for A and B, is also a theorem. If you become stuck in finding a derivation, you may find that using a theorem in order to transform a particular sentence makes everything easy again. Equivalences are especially useful for this purpose; for example, $\vdash \neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$ — so from $\neg(A \wedge B)$ and one half of the equivalence you can derive $\neg A \vee \neg B$.

Proving theorems and then including them in a proof can make finding derivations much easier than starting from first principles and using just the given rules. Using *derived rules* can also simplify derivations. As an example, consider the following scheme, which is a typical sequence of steps for deriving S by contradiction. The derived rule in this case will be called PC for *proof by contradiction*:

1	$\neg S$	
2	$\vdots$	
3	$\bot$	
4	$\neg\neg S$	$\neg\mathcal{I}$
5	S	$\neg\neg(4)$

The steps can be contracted into a new proof rule:

$\neg S$	
$\vdots$	
$\bot$	
S	PC

It is not *essential* to make use of any derived rules, for the preceding rules are enough for any proof; but they can be used to shorten a proof. The following are some more derived rules:

contrapositive	from $A \to B$ and $\neg B$ derive $\neg A$
simple resolution 1	from $A \vee B$ and $\neg A$ derive B
simple resolution 2	from $\neg A \vee B$ and A derive B
resolution	from $A \vee B$ and $\neg A \vee C$ derive $B \vee C$

As an example, the derivation of the resolution rule is given in Figure 16.15.

1	$A \vee B$					
2	$\neg A \vee C$					
3	A				B	
4	$\neg A$		C		$B \vee C$	$\vee\mathcal{I}$
5	$\bot$	$\neg\mathcal{E}(3,4)$	$B \vee C$	$\vee\mathcal{I}$		
6	$B \vee C$	$\bot\mathcal{E}$				
7	$B \vee C$			$\vee\mathcal{E}(2)$		
8	$B \vee C$					$\vee\mathcal{E}(1)$

Figure 16.15

Some hints for deriving natural deduction proofs

You have put the assumptions at the top of a proof and the conclusion at the bottom — what do you do next? You might be able to use some automatic steps, $\rightarrow\mathcal{I}$ for example, which yield a requirement for deriving various subproofs. Or, you might be able to use some insight, for example to prove $C \vee D$ using $\vee\mathcal{I}$, prove C. Since introduction rules produce conclusions they are usually used when filling in a proof from the bottom upwards — their use is dictated by the form of the conclusion. Elimination rules work on the data and so these are usually used when filling in a proof from the top downwards.

In addition to these guidelines there are many useful tactics which you will discover for yourself. We describe an assortment of such tactics next.

- $\rightarrow$ as 'if' — If there is a sentence of the form $D \rightarrow C$ and the conclusion is C then try to show D. C follows using $\rightarrow\mathcal{E}$. $D \rightarrow C$ can be read as C if D, from which the tactic gets its name.
- make use of $\neg S$ — If the conclusion is $\bot$, then perhaps there is a negative sentence $\neg S$ that is available which could be used in a $\neg\mathcal{E}$ step once S had been proved.
- $\bot\mathcal{E}$ anywhere — If you cannot see what to do next perhaps you can derive $\bot$ and then use $\bot\mathcal{E}$. This often happens in some branches of a $\vee\mathcal{E}$ box, in those branches which 'are not what the argument is about' (for example, in the left-hand inner box of Figure 16.15).
- combined $\vee$ rules — The $\vee\mathcal{I}$ and $\vee\mathcal{E}$ rules often go together — first use $\vee\mathcal{E}$ and then $\vee\mathcal{I}$. Suppose the data is $X \vee Y$ and the conclusion is $C \vee D$. $\vee\mathcal{E}$ will force two subproofs, one using X and one using Y, and perhaps in one you can prove C and in the other D. In both cases $\vee\mathcal{I}$ will yield $C \vee D$, as you required.
- equivalence — Any sentence can be rewritten using an equivalence. When filling in a proof downwards, data can be rewritten into new data and when filling in a proof upwards, conclusions can be rewritten into new conclusions.
- theorem — Remember that it is possible to use theorems anywhere in a proof, for these are previously proved sequents that do not depend on any data and so could be used anywhere.
- lemma — In some cases a large proof can best be tackled by breaking it down into smaller steps. If your problem is to show $Data \vdash Conclusion$ then maybe you could show $Data \vdash Lemma$ and then make use of $Lemma$ to show $Conclusion$ — ($Data$ and $Lemma$) $\vdash Conclusion$. The choice of which lemma to prove is often called a 'Eureka' step for it sometimes requires considerable ingenuity.
- excluded middle — If there are no negative sentences, then perhaps you can introduce a theorem of the form $Z \vee \neg Z$ and immediately use $\vee\mathcal{E}$.

Of course, some ingenuity is needed to choose a suitable Z, but it is worth trying Z as the conclusion you are trying to prove.

- PC — Perhaps you can use the proof by contradiction derived rule.
- If all else fails, use PC, or excluded middle.
  And if all else does not fail then *do not* use PC — the negated assumptions it introduces often make the proof more difficult to understand.

Most practical proofs make use of three of the tactics on a large scale; they are the *lemma*, *equivalence* and *theorem* tactics:

- The lemma tactic is used to break the proof into smaller steps.
- The equivalence tactic is used to rewrite the data into the most appropriate form for the problem.
- The theorem tactic is used to make large steps in one go by appealing to a previous proof.

In practice, we make use of hundreds of theorems, some of which are exercises in this book and some of which you will discover for yourself. So watch out for them!

16.3 Examples

The various rules and tactics of this chapter are illustrated in the following examples.

Show $\neg P \vdash P \to Q$

1	$\neg P$	
2	P	
3	$\bot$	$\neg\mathcal{E}(1,2)$
4	Q	$\bot\mathcal{E}(3)$
5	$P \to Q$	$\to\mathcal{I}$

Figure 16.16 $\neg P \vdash P \to Q$

The derivation in Figure 16.16 is a useful one to remember. It is used in the following example which derives a famous law called 'Pierce's law' after the logician Charles Pierce.

Show $\vdash ((P \to Q) \to P) \to P$

Two proofs are given (in Figures 16.17 and 16.18) — the first uses $P \vee \neg P$ and the second uses PC. They both illustrate the benefit of planning in a proof. In the first proof it is clear that the sentence $(P \to Q) \to P$ will yield P, the conclusion, if $P \to Q$ can be proven. Also, the sentence $P \vee \neg P$ means that since P can be derived from P, $P \to Q$ will have to be proven from $\neg P$. And we have shown this in Figure 16.16. In the second proof a useful technique is used —'use PC if all else fails'. Applying it in this example leads to the goal of $\bot$ — the necessary $\neg\mathcal{E}$ step will require a sentence and its negation to be derived. $\neg P$ is already an assumption so consider deriving P. This can be done by deriving $P \to Q$, which follows from $\neg P$, again as in Figure 16.16. Notice that here we have had to use some insight in order to

1	$(P \to Q) \to P$			
2	$\neg P \vee P$			(Th)
3	$\neg P$		P	
4	$P \to Q$	(Fig. 16.16)	P	✓ (3)
5	P	$\to\mathcal{E}(1,4)$		
6	P			$\vee\mathcal{E}(2)$
7	$((P \to Q) \to P) \to P$			$\to\mathcal{I}$

Figure 16.17 $\vdash ((P \to Q) \to P) \to P$

1	$(P \to Q) \to P$	
2	$\neg P$	
3	$P \to Q$	(Fig. 16.16)
4	P	$\to\mathcal{E}(1,3)$
5	$\bot$	$\neg\mathcal{E}(2,4)$
6	P	PC
7	$((P \to Q) \to P) \to P$	$\to\mathcal{I}$

Figure 16.18 $\vdash ((P \to Q) \to P) \to P$

apply the heuristics in the correct order. If you tried to use '$\to$ as if' before PC, that is, tried to prove $P \to Q$ without obtaining $\neg P$, you would fail.

Show $A \wedge B \rightarrow C, \neg D \rightarrow \neg(E \rightarrow F), C \rightarrow (E \rightarrow F) \vdash A \rightarrow (B \rightarrow D)$
The derivation for this example (in Figure 16.19) proves, and then uses, the lemma $E \rightarrow F$ to help fill in the proof between lines 5 and 12. That is, $E \rightarrow F$ can be proved first and then it can be used to prove D. If the proof

1	$A \wedge B \rightarrow C$	
2	$\neg D \rightarrow \neg(E \rightarrow F)$	
3	$C \rightarrow (E \rightarrow F)$	
4	A	
5	B	
6	$A \wedge B$	$\wedge\mathcal{I}(4,5)$
7	C	$\rightarrow\mathcal{E}(6,1)$
8	$(E \rightarrow F)$	$\rightarrow\mathcal{E}(3,7)$(a lemma)
9	$\neg D$	
10	$\neg(E \rightarrow F)$	$\rightarrow\mathcal{E}(2,9)$
11	$\perp$	$\neg\mathcal{E}(10,8)$
12	D	PC
13	$B \rightarrow D$	$\rightarrow\mathcal{I}$
14	$A \rightarrow (B \rightarrow D)$	$\rightarrow\mathcal{I}$

Figure 16.19 $A \wedge B \rightarrow C, \neg D \rightarrow \neg(E \rightarrow F), C \rightarrow (E \rightarrow F) \vdash A \rightarrow (B \rightarrow D)$

were to be written in English it might look as follows.
Proposition 16.3 $A \wedge B \rightarrow C, \neg D \rightarrow \neg(E \rightarrow F), C \rightarrow (E \rightarrow F) \vdash A \rightarrow (B \rightarrow D)$
Proof To show $A \rightarrow (B \rightarrow D)$ assume A and show $B \rightarrow D$. So assume B and try to show D. (Next a little bit of ingenuity is required. You notice that to show D it would suffice to show that $E \rightarrow F$, as the assumption of $\neg D$ then leads to a contradiction.) So, try to show $E \rightarrow F$. From A and B derive C and hence $E \rightarrow F$. Finally, D can be shown by using proof by contradiction. $\neg D$ leads to $\neg(E \rightarrow F)$, which gives a contradiction with the lemma $E \rightarrow F$. □

A specification example

One of the Miranda programs considered earlier was `min :: num -> num -> num` with specification: $\forall x \forall y \forall z.\ [z \leq x \wedge z \leq y \wedge (z = x \vee z = y)]$, where `z = min x y`. This can be used to define a function `min3` that yields the smallest

value of three numbers. What is the specification of such a function `min3`? The result must certainly be one of the three numbers and should also be $\leq$ each number. A suitable program is

```
min3 ::  num -> num -> num -> num
min3 x y z = min ( min x y) z
```

That is, find the minimum of the first two numbers and then the minimum of this result and the third number. To show that the program meets the specification, we must show that:

$\forall x \forall y \forall z.\ [\texttt{min3} \leq x \wedge \texttt{min3} \leq y \wedge \texttt{min3} \leq z \wedge (\texttt{min3} = x \vee \texttt{min3} = y \vee \texttt{min3} = z)]$

that is

$$\forall x, y, z. \left[\begin{array}{c} \texttt{min (min } x\ y\texttt{) } z \leq x \wedge \texttt{min (min } x\ y\texttt{) } z \leq y \wedge \\ \texttt{min (min } x\ y\texttt{) } z \leq z \wedge \\ \texttt{(min (min } x\ y\texttt{) } z = x \vee \texttt{min (min } x\ y\texttt{) } z = y \vee \\ \texttt{min (min } x\ y\texttt{)} z = z) \end{array}\right]$$

To show that a sentence is true for all x, y, z we should show that it is true for any arbitrary values in place of x, y, z. (See Section 17.2.) Suppose X, Y, Z are arbitrary values for x, y, z. Then we have to show

$$\texttt{min (min } X\ Y\texttt{) } Z \leq X \wedge \texttt{min (min } X\ Y\texttt{) } Z \leq Y \wedge \texttt{min (min } X\ Y\texttt{) } Z \leq Z \wedge$$
$$\texttt{(min (min } X\ Y\texttt{) } Z = X \vee \texttt{min (min } X\ Y\texttt{) } Z = Y \vee \texttt{min (min } X\ Y\texttt{)} Z = Z)$$

First, what are the initial assumptions? The specification of `min` for a start. Any other assumptions can be added as the proof progresses. A look at the sentence to be proved reveals that it is a conjunction of four sentences, so each one has to be proved.

The first is $\texttt{min (min } X\ Y\texttt{) } Z \leq X$. Use the specification of `min` — write $\texttt{min } X\ Y$ as u, then $\texttt{min } u\ Z \leq u \wedge \texttt{min } u\ Z \leq Z$. (Since the result of $\texttt{min } X\ Y$ is a num, it satisfies the implicit pre-condition for the first argument of `min` in $\texttt{min (min } X\ Y\texttt{) } Z$.) Also, $u \leq X \wedge u \leq Y$. Hence, after using the fact that $\leq$ is transitive, $\texttt{min } u\ Z \leq X$, $\texttt{min } u\ Z \leq Y$, $\texttt{min } u\ Z \leq Z$. This gives the first three parts. The fourth is a disjunction.

One way to prove a disjunction is to use another. From the specification of `min`, $u = X \vee u = Y$, and $\texttt{min } u\ Z = u \vee \texttt{min } u\ Z = Z$. Take the second of these: $\texttt{min } u\ Z = Z$ will yield the result after $\vee\mathcal{I}$. Assuming now that $\texttt{min } u\ Z = u$, from the first disjunction there are two cases: $u = X$ for one case, and $u = Y$ for the other. Together, $u = X$ and $\texttt{min } u\ Z = u$ give $\texttt{min } u\ Z = X$, which again yields the result. The other case is similar. The box proof is shown in Figure 16.20. (Notice that lines 7, 8, 9 and 10—14 give the derivations of the four conjuncts in line 16.)

1 $\texttt{min}\ u\ Z = u \vee \texttt{min}\ u\ Z = Z$

2 $u = X \vee u = Y$

3 $u \leq X \wedge u \leq Y$

4 $u \leq X,\ u \leq Y$ $\wedge\mathcal{E}(3)$

5 $\texttt{min}\ u\ Z \leq u \wedge \texttt{min}\ u\ Z \leq Z$

6 $\texttt{min}\ u\ Z \leq u,\ \texttt{min}\ u\ Z \leq Z$ $\wedge\mathcal{E}(5)$

7 $\texttt{min}\ u\ Z \leq X$ by transitivity of $\leq$ (4, 6)

8 $\texttt{min}\ u\ Z \leq Y$ by transitivity of $\leq$ (4, 6)

9 $\texttt{min}\ u\ Z \leq Z$ $\checkmark(6)$

10 $\texttt{min}\ u\ Z = u$ | $\texttt{min}\ u\ Z = Z$

11	$u = X$	$u = Y$
12	$\texttt{min}\ u\ Z = X$ $(10, 11)$ (by equality)	$\texttt{min}\ u\ Z = Y$ $(10, 11)$ (by equality)
13	$\texttt{min}\ u\ Z = X \vee \texttt{min}\ u\ Z = Y \vee \texttt{min}\ u\ Z = Z$ $\vee\mathcal{I}(12)$	$\texttt{min}\ u\ Z = X \vee \texttt{min}\ u\ Z = Y \vee \texttt{min}\ u\ Z = Z$ $\vee\mathcal{I}(12)$

$\texttt{min}\ u\ Z = X \vee \texttt{min}\ u\ Z = Y \vee \texttt{min}\ u\ Z = Z$ $\vee\mathcal{I}(10)$

14 $\texttt{min}\ u\ Z = X \vee \texttt{min}\ u\ Z = Y \vee \texttt{min}\ u\ Z = Z$ $\vee\mathcal{E}(2)$

15 $\texttt{min}\ u\ Z = X \vee \texttt{min}\ u\ Z = Y \vee \texttt{min}\ u\ Z = Z$ $\vee\mathcal{E}(1)$

16 $\texttt{min}\ uZ \leq X \wedge \texttt{min}\ u\ Z \leq Y \wedge \texttt{min}\ u\ Z \leq Z \wedge$ $\texttt{min}\ u\ Z = X \vee \texttt{min}\ u\ Z = Y \vee \texttt{min}\ u\ Z = Z$ $\wedge\mathcal{I}$

Figure 16.20

16.4 Summary

- A valid argument consists of a collection of premisses and a conclusion such that if the premisses are true then the conclusion must be true, too.
- The basic natural deduction rules for propositional sentences are given in Appendix C.
- The $\vee\mathcal{I}$, $\wedge\mathcal{E}$, $\rightarrow\mathcal{E}$, $\neg\mathcal{E}$ rules require some ingenuity, choosing which rules to apply and when, whereas the $\wedge\mathcal{I}$, $\vee\mathcal{E}$, $\rightarrow\mathcal{I}$, $\neg\mathcal{I}$ rules are all automatic, requiring just some preparation, and should be applied as soon as you realize that they can be applied.
- Derived rules can be useful, especially the rule PC, proof by contradiction.
- Boxes are useful for structuring proofs and to show where assumptions hold.
- There are various tactics for finding derivations:

  $\rightarrow$ as 'if'

  making use of $\neg S$

  use $\bot\mathcal{E}$ anywhere

  PC

  excluded middle

  combined $\vee$ rules

  equivalence

  theorem

  lemma

16.5 Exercises

1. Show

   (a) $\vdash P \wedge Q \rightarrow P$ (b) $P \vdash Q \rightarrow (P \wedge Q)$
   (c) $P \rightarrow Q, \neg Q \vdash \neg P$ (d) $\neg P \vdash P \rightarrow Q$
   (e) $\neg P, P \vee Q \vdash Q$ (f) $\neg I \wedge \neg F \vdash \neg(I \vee F)$
   (g) $\vdash P \rightarrow (Q \rightarrow P)$ (h) $P \rightarrow S, (P \rightarrow Q) \rightarrow S \vdash S$
   (i) $F \rightarrow (B \vee W), \neg(B \vee P), W \rightarrow P \vdash \neg F$
   (j) $P \rightarrow Q, \neg P \rightarrow R, Q \rightarrow S, R \rightarrow S \vdash S$
   (k) $(C \wedge N) \rightarrow T, H \wedge \neg S, H \wedge \neg(S \vee C) \rightarrow P \vdash (N \wedge \neg T) \rightarrow P$
   (l) $R \rightarrow \neg I, I \vee F, \neg F \vdash \neg R$
   (m) $P \rightarrow (Q \rightarrow R) \vdash (P \rightarrow Q) \rightarrow (P \rightarrow R)$

2. For each of the equivalences $A \equiv B$ show $A \vdash B$ and $B \vdash A$.

   (a) $P \wedge (P \vee Q) \equiv P$ (b) $P \vee (P \wedge Q) \equiv P$
   (c) $P \to Q \equiv \neg Q \to \neg P$ (d) $P \to Q \equiv \neg P \vee Q$
   (e) $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ (f) $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
   (g) $(P \wedge Q) \to R \equiv P \to (Q \to R)$
   (h) $P \vee Q \equiv \neg(\neg P \wedge \neg Q)$
   (i) $P \vee Q \equiv (P \to Q) \to Q$
   (j) $\neg(\neg P \wedge \neg Q) \equiv P \vee Q$
   (k) $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
   (l) $(P \vee Q) \to R \equiv (P \to R) \wedge (Q \to R)$
   (m) $(P \to Q) \wedge (Q \to P) \equiv (P \wedge Q) \vee (\neg P \wedge \neg Q)$

3. Derive an introduction and elimination rule for $\leftrightarrow$ based on the equivalences $A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$ and $A \leftrightarrow B \equiv (A \wedge B) \vee (\neg A \wedge \neg B)$. Use your new rules to show:

   (a) $\neg(P \leftrightarrow Q) \equiv \neg P \leftrightarrow Q$

   (b) $P \leftrightarrow (P \wedge Q) \equiv P \to Q$

   (c) $P \leftrightarrow (P \vee Q) \equiv Q \to P$

   (d) $P \leftrightarrow Q \equiv Q \leftrightarrow P$

   (e) $P \leftrightarrow (Q \leftrightarrow R) \equiv (P \leftrightarrow Q) \leftrightarrow R$

4. Many tautologies of the form $\vdash A \to B$ give rise to derived rules of the form $A \vdash B$. Explain how.
5. Formulate a derived natural deduction rule for *if-then-else* $\mathcal{I}$ and *if-then-else* $\mathcal{E}$. The first will be based on the rules $\wedge\mathcal{I}$ and $\to\mathcal{I}$, the second on $\wedge\mathcal{E}$ and $\to\mathcal{E}$. (HINT: $if\text{-}then\text{-}else(x, y, z)$ is equivalent to $x \to y \wedge \neg x \to z$.)
   Use the rules to show

   (a) $if\text{-}then\text{-}else(A, B, C) \vdash if\text{-}then\text{-}else(\neg A, C, B)$

   (b) $if\text{-}then\text{-}else(A, if\text{-}then\text{-}else(D, B, C), C)$
   $\vdash if\text{-}then\text{-}else(D, if\text{-}then\text{-}else(A, B, C), C)$

6. (a) Derive the rules 'contrapositive' , 'simpler resolution 1' and 'simpler resolution 2'.

   (b) Prove the rule $\neg\neg$ as a derived rule using the schema $Q \vee \neg Q$.

   (c) Prove the inverse of $\neg\neg$ (that is, from Q derive $\neg\neg Q$) as a derived rule.

Chapter 17

Natural deduction for predicate logic

In the preceding chapter we looked at natural deduction rules for the various logical connectives. Each connective was associated with an introduction rule for use in deriving a sentence involving the connective, and an elimination rule for deriving further sentences from a sentence using the connective.

There are six more natural deduction rules to be introduced in this chapter. Four of them cover the quantifiers, which also have elimination and introduction rules — $\forall\mathcal{I}$, $\forall\mathcal{E}$, $\exists\mathcal{I}$, $\exists\mathcal{E}$. The other two are for reasoning with equality, which is an important predicate that has its own rules: *eqsub*, which acts rather like an equality elimination rule, and *reflex*, which acts like an equality introduction rule.

17.1 $\forall$-elimination ($\forall\mathcal{E}$) and $\exists$-introduction ($\exists\mathcal{I}$) rules

The rules

$\forall\mathcal{E}$ From a sentence $\forall x.\ P[x]$ you may derive $P[t]$ for any ground term t that is available, where t is substituted for x everywhere that it occurs in $P[x]$.

$$\frac{\forall x.\ P[x]}{P[t]\quad(\forall\mathcal{E})}$$

$\exists\mathcal{I}$ A sentence $\exists x.\ P[x]$ can be derived from $P[b]$, where b is any available ground term and x is substituted for one or more occurrences of b in $P[b]$, or to show $\exists x.\ P[x]$ try to show $P[b]$ for some available ground term b:

$$\frac{P[b]}{\exists x.\ P[x]\qquad(\exists\mathcal{I})}$$

A *ground term* is one that contains no variables. In addition, the terms t or b may only involve constants and/or function symbols that are already *available* in the current context.

Function symbols and constants appearing in proofs cannot be invented as the fancy takes you; rather, they must:

- either be occurring in sentences in the overall problem (that is, sentences which are mentioned in the premisses or conclusion);
- or be implicit because a particular interpretation of the predicates is known (for example, various numbers);
- or be introduced when using the rules $\forall\mathcal{I}$ or $\exists\mathcal{E}$ (see Section 17.2).

This means that at different places in a proof different constants may be available for substitution in the use of $\forall\mathcal{E}$ or $\exists\mathcal{I}$.

The $\forall\mathcal{E}$ rule is frequently used and allows a general sentence about all individuals to become a particular sentence about some individual t. The $\exists\mathcal{I}$ rule is mostly used when filling in a proof from the conclusion upwards. That is, to show $\exists x.\ P[x]$, first a particular b is chosen (using some ingenuity) and then an attempt to show $P[b]$ is made.

Notice that the term t in an application of $\forall\mathcal{E}$ must be substituted for *all* occurrences of the bound variable, for otherwise the resulting sentence would not be properly formed.

The $\exists\mathcal{I}$ rule can also be used forwards, for if a sentence $P[b]$ has been derived then certainly $\exists z.\ P[z]$ is true, too. In that case, any number of occurrences (≥ 1) of the selected term b can be replaced by the bound variable x. In order that the resulting sentence $\exists x.\ P[x]$ is properly formed the bound variable x should be new to $P[b]$.

Quite a bit of ingenuity is necessary in using these rules; in the use of the $\forall\mathcal{E}$ rule you need to prevent too many particular sentences being generated that are not going to be useful to the proof; in the backward use of the $\exists\mathcal{I}$ rule you need to pick an individual b for which $P[b]$ can indeed be proved.

The notation using typed quantifiers is widely used in specifying programs, especially for qualifiers such as 'person', 'lists', 'numbers', etc. The $\forall\mathcal{E}$ and $\exists\mathcal{I}$ rules each have a typed counterpart that is derived from the translations

$\forall x : \mathit{type}.\ P[x]$ translates to $\forall x.\ [\mathit{is\text{-}type}(x) \rightarrow P[x]]$

and

$\exists x : \mathit{type}.\ P[x]$ translates to $\exists x.\ [\mathit{is\text{-}type}(x) \wedge P[x]]$

The typed rules are

$$\frac{\mathit{is\text{-}type}(t) \qquad \forall x : \mathit{type}.\ P[x]}{P[t]}\ (\forall\mathcal{E}) \qquad\qquad \frac{\mathit{is\text{-}type}(b) \qquad P[b]}{\exists x : \mathit{type}.\ P[x]}\ (\exists\mathcal{I})$$

For $\forall\mathcal{E}$ the term t must be of the correct type and satisfy $\mathit{is\text{-}type}(t)$ in order for an implicit $\rightarrow\mathcal{E}$ step to be made to derive $P[t]$. For the $\exists\mathcal{I}$ rule the term b must satisfy $\mathit{is\text{-}type}(b)$ so that an implicit $\wedge\mathcal{I}$ step can be made. These conditions mean that an additional check must be made on the terms

being substituted. Suppose, as an example, that a term of type 'integer' was required in a $\forall\mathcal{E}$ step. The derivation so far may not mention any numbers explicitly, but implicitly the data includes a whole theory about integers, including all the facts we know about numbers such as $2 \neq 3$, 5 is prime, and so on. Any integer can be used as a substitute for t. Similarly, before using $\exists\mathcal{I}$ to derive $\exists x : int.\ P(x)$ from $P(2)$, say, you must check that *is-int*(2) is true, which of course it is.

The $\forall\mathcal{E}$ rule is often used together with $\rightarrow\mathcal{E}$ or $\neg\mathcal{E}$ to form combined rules called, respectively, $\forall\rightarrow\mathcal{E}$ and $\forall\neg\mathcal{E}$. In both of these cases the $\forall\mathcal{E}$ step is implicit. Moreover, just as $\rightarrow\mathcal{E}$ and $\neg\mathcal{E}$ can be used backwards as well as forwards, so, too, can the combinations be used backwards as well as forwards. We will see several examples of this in the next section.

The formats are

$$\frac{\forall x.\ [P[x] \rightarrow Q[x]] \qquad P[c]}{Q[c]}\ (\forall\rightarrow\mathcal{E}) \quad \text{and} \quad \frac{\forall x.\ \neg P[x] \qquad P[c]}{\bot}\ (\forall\neg\mathcal{E})$$

The $\forall\neg\mathcal{E}$ rule can be used to show a contradiction by showing some sentence $P[c]$ and then implicitly using $\forall\mathcal{E}$ to derive $\neg P[c]$ and the contradiction.

Some examples

In our first example, shown in Figure 17.1, we give a proof of $tired(lenny) \wedge lion(lenny) \rightarrow does(lenny, sleep)$. The initial data appears in lines 1—3 and, after the automatic step of $\rightarrow\mathcal{I}$, several non-automatic steps are made in lines 4—8. The $\forall\rightarrow\mathcal{E}$ rule is used several times. For example, at line 7 $\forall\mathcal{E}$ is first (implicitly) applied to line 1, to derive $lion(lenny) \rightarrow does(lenny,\ hunt) \vee does(lenny,\ sleep)$ and then $\rightarrow\mathcal{E}$ is applied to derive $does(lenny,\ hunt) \vee does\ (lenny,\ sleep)$. After that, another automatic step is made to prepare for $\vee\mathcal{E}$.

The second example, shown in Figures 17.2 and 17.3, is a proof of an existentially quantified sentence $\exists x.\ \neg shot(x, Diana)$. The initial data given in lines 1—4 can be used to show the conclusion in two different ways. The simpler way is given first. This example is typical of real situations when more data than is required to prove the given goal is available, making ingenuity even more necessary in finding the proof.

The first derivation proves that Diana did not shoot herself, and the second that Janet did not shoot Diana. The combined rule $\forall\neg\mathcal{E}$ is used in the second derivation at line 8 — the new conclusion $inhouse(Janet) \wedge ingarden(Janet)$ is derived because if this is proved then $\forall\mathcal{E}$ using line 3 will give a contradiction. All uses of $\forall\mathcal{E}$ and $\exists\mathcal{I}$ require some insight into which substitutions for the bound variable will prove suitable. In this case there are two names, *Janet* and *Diana*, and either might be appropriate.

1	$\forall x.\ [lion(x) \rightarrow does(x, hunt) \vee does(x, sleep)]$			
2	$\forall x.\ \forall y.\ [does(x, y) \rightarrow can(x, y)]$			
3	$\forall x.\ [tired(x) \wedge lion(x) \rightarrow \neg can(x, hunt)]$			
4	$tired(lenny) \wedge lion(lenny)$			
5	$tired(lenny)$			$\wedge\mathcal{E}(4)$
6	$lion(lenny)$			$\wedge\mathcal{E}(4)$
7	$does(lenny, hunt) \vee does(lenny, sleep)$			$\forall\rightarrow\mathcal{E}(1,6)$
8	$\neg can(lenny, hunt)$			$\forall\rightarrow\mathcal{E}(3,4)$
9	$does(lenny, hunt)$		$does(lenny, sleep)$	
10	$can(lenny, hunt)$	$\forall\rightarrow\mathcal{E}(9,2)$	$does(lenny, sleep)$	$\checkmark(9)$
11	$\bot$	$\neg\mathcal{E}(10,8)$		
12	$does(lenny, sleep)$	$\bot\mathcal{E}$		
13	$does(lenny, sleep)$			$\vee\mathcal{E}(7)$
14	$tired(lenny) \wedge lion(lenny) \rightarrow does(lenny, sleep)$			$\rightarrow\mathcal{I}$

Figure 17.1 Proof of $tired(lenny) \wedge lion(lenny) \rightarrow does(lenny, sleep)$

1	$\forall x.\ \neg shot(x, x)$	
2	$inhouse(Janet)$	
3	$\forall x.\ \neg(inhouse(x) \wedge ingarden(x))$	
4	$\forall x.\ [shot(x, Diana) \rightarrow ingarden(x)]$	
5	$\neg shot(Diana, Diana)$	$\forall\mathcal{E}(1)$
6	$\exists x.\ \neg shot(x, Diana)$	$\exists\mathcal{I}(5)$

Figure 17.2 Proof of $\exists x.\ \neg shot(x, Diana)$

Show $P(a) \vee P(b), \forall x.\ [P(x) \rightarrow Q(x)] \vdash \exists x.\ Q(x)$

Figure 17.4 illustrates a feature of the $\exists\mathcal{I}$ rule. Many problems are straightforward in that there is a particular term that makes $\exists x.\ A[x]$ follow from the current data. (For example, if the data had been $\forall x.\ [P(x) \rightarrow Q(x)], P(a)$ then $\exists x.\ Q(x)$ would follow because of $Q(a)$.) Sometimes, this is not the case, and although $\exists x.\ A[x]$ follows from the

1	$\forall x.\ \neg shot(x, x)$	
2	$inhouse(Janet)$	
3	$\forall x.\ \neg(inhouse(x) \wedge ingarden(x))$	
4	$\forall x.\ [shot(x, Diana) \rightarrow ingarden(x)]$	
5	$shot(Janet, Diana)$	
6	$ingarden(Janet)$	$\forall{\rightarrow}\mathcal{E}(4,5)$
7	$inhouse(Janet) \wedge ingarden(Janet)$	$\wedge\mathcal{I}(2,6)$
8	$\bot$	$\forall\neg\mathcal{E}(7,3)$
9	$\neg shot(Janet, Diana)$	$\neg\mathcal{I}$
10	$\exists x.\ \neg shot(x, Diana)$	$\exists\mathcal{I}(9)$

Figure 17.3 Another proof of $\exists x.\ \neg shot(x, Diana)$

1	$P(a) \vee P(b)$				
2	$\forall x.\ [P(x) \rightarrow Q(x)]$				
3	$P(a)$		$P(b)$		
4	$Q(a)$	$\forall{\rightarrow}\mathcal{E}(2,3)$	$Q(b)$		$\forall{\rightarrow}\mathcal{E}(3,2)$
5	$\exists x.\ Q(x)$	$\exists\mathcal{I}$	$\exists x.\ Q(x)$		$\exists\mathcal{I}$
6	$\exists x.\ Q(x)$				$\vee\mathcal{E}(1)$

Figure 17.4 $P(a) \vee P(b), \forall x.\ [P(x) \rightarrow Q(x)] \vdash \exists x.\ Q(x)$

available data there may be uncertainty as to which term makes it do so.

Typically, this occurs when there is a disjunction in the data and one 'witness' (substitution for x) is appropriate in the context of one disjunct and another in the context of a second. Our example has a disjunction in its data which is applied before the application of $\exists\mathcal{I}$. On the other hand, in the proof of $\forall x.\ [P(x) \rightarrow Q(x)], \neg P(b) \rightarrow P(a) \vdash \exists x.\ Q(x)$, shown in Figure 17.5, the disjunction $P(b) \vee \neg P(b)$ is added as a theorem. This is a common technique, but you may need several attempts before you find the correct disjunction to introduce. The one used here is not the only possibility for either of $P(a) \vee \neg P(a)$ or $\exists x.\ Q(x) \vee \neg\exists x.\ Q(x)$ could have been used instead.

1	$\neg P(b) \to P(a)$			
2	$\forall x.\ [P(x) \to Q(x)]$			
3	$\neg P(b) \vee P(b)$			(Th)
4	$\neg P(b)$		$P(b)$	
5	$P(a)$	$\to\mathcal{E}(1,4)$	$Q(b)$	$\forall\to\mathcal{E}(2,4)$
6	$Q(a)$	$\forall\to\mathcal{E}(2,5)$	$\exists x.\ Q(x)$	$\exists\mathcal{I}(5)$
7	$\exists x.\ Q(x)$	$\exists\mathcal{I}(6)$		
8	$\exists x.\ Q(x)$			$\vee\mathcal{E}(3)$

Figure 17.5 $\forall x.\ [P(x) \to Q(x)], \neg P(b) \to P(a) \vdash \exists x.\ Q(x)$

1	$\forall x : num.\ P(x)$	
2	$P(25)$	$\forall\mathcal{E}(1)$
3	$\exists x : num.\ P(x)$	$\exists\mathcal{I}(2)$

Figure 17.6 $\forall x : num.\ P(x) \vdash \exists x : num.\ P(x)$

Show $\forall x : num.\ P(x) \vdash \exists x : num.\ P(x)$ (Figure 17.6)

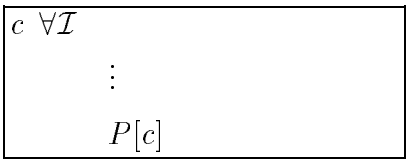
Here, in order to show the conclusion an assumption has to be made that there are some numbers, so suppose that there are. Two checks then have to be made — that '25' is a number in deriving line 2 from line 1, and that '25' is a number in deriving line 3.

17.2 $\forall$-introduction ($\forall\mathcal{I}$) and $\exists$-elimination ($\exists\mathcal{E}$) rules

$\forall$-introduction

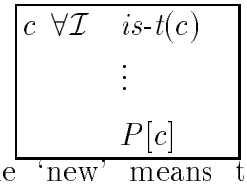
The next rule that we consider is $\forall\mathcal{I}$, and *its* use introduces a new constant into the proof. The rule is

> A proof of $\forall x.\ P[x]$ can be obtained from a proof of $P[c]$ for some *new* constant c.



$\forall x.\ P[x] \qquad (\forall\mathcal{I})$

or typed



The 'new' means that c is introduced for the first time inside the box that contains the subproof of $P[c]$; c is only available within that box and it cannot be mentioned outside it. So, in particular, c cannot occur in $\forall x.\ P[x]$. The $c\ \forall\mathcal{I}$ in the left-hand corner is a reminder that c must be new.

The version using a typed quantifier is derived from the untyped version and $\rightarrow\mathcal{I}$ using the translation of $\forall x : t.\ P[x]$ into $\forall x.\ [\mathit{is}\text{-}t(x) \rightarrow P[x]]$.

The $\forall\mathcal{I}$ rule is completely automatic and is used in a backwards direction from goal to subgoal. The motivation behind this rule is the commonly quoted law:

> If one can show $P[u]$ for an arbitrary u, then $\forall x.\ P[x]$ holds.

The use of a new term for c implements the 'arbitrary' part of the law.

The following is an informal explanation of why the rule 'works': in order to derive $\forall x.\ P[x]$, the derivation should work for whatever value v could be substituted for x and should not depend on properties of a particular v. Since c is new, any data that is used to prove $P[c]$ will not mention c and the derivation cannot rely on special properties of c (apart from that it is of type t), as there are none. Properties are either not relevant or are completely general, of the form $\forall\cdots$, in which case they apply to any value.

A very common pattern used in quantified sentences is $\forall x.\ [P[x] \rightarrow Q[x]]$. If this sentence is a conclusion then two automatic steps are immediately applicable — first a $\forall\mathcal{I}$ step and then a $\rightarrow\mathcal{I}$ step. These can be combined into one step, $\forall{\rightarrow}\mathcal{I}$, that requires just one box instead of two, as is done implicitly in deriving a typed version of the $\forall\mathcal{I}$ rule.

Remember that in Chapter 15 we encountered a difficulty in checking whether a universal sentence was true when there was an infinite number of values to check? Well, now we have an alternative approach. The sentence is checked for one or more arbitrary values which between them cover all the possible cases. For example, to show that $\forall x : \mathit{int}.\ P[x]$, we might try to show that $P[c]$ for an arbitrary integer c. Now, any integer is either < 0, $= 0$ or > 0, so we could try to show that $P[c]$ is true in each of the three cases. (Alternatively, any integer is also prime or non-prime, so we could try to show that $P[c]$ is true in those two cases.)

∃-elimination

The $\exists\mathcal{E}$ rule is another completely automatic rule that introduces a new constant into a proof. It may seem a little difficult at first sight and you should thus learn it by heart and understand why it appears as it does.

> To derive Q using $\exists x.\ P[x]$, derive Q using $P[c]$, where c is a new constant.

The format for the $\exists\mathcal{E}$ rule is:

$$\exists x.\ P[x]$$

$c\exists\mathcal{E}$	$P[c]$	
	$\vdots$	
	Q	
	Q	$(\exists\mathcal{E})$

or typed

$$\exists x : t.\ P[x]$$

$c\exists\mathcal{E}$	$P[c]$	
	$is\text{-}t(c)$	
	$\vdots$	
	Q	
	Q	$(\exists\mathcal{E})$

The version using a typed quantifier is derived from the untyped version and $\wedge\mathcal{E}$ using the translation of $\exists x : t.\ P[x]$ into $\exists x.\ [is\text{-}t(x) \wedge P[x]]$.

Again, c must be a new constant and the box is used to indicate where c is available. In particular, the conclusion Q must not mention c. Notice that the conclusion appears twice; outside the box it is justified by $\exists\mathcal{E}$ and inside by something else. The rule is best applied as soon as possible in a proof so that the new constant c is available as soon as possible.

An informal explanation of why the rule works is as follows: in order to use $\exists x.\ P[x]$ a name has to be given to fix the 'x that makes $P[x]$ true'. Although it would be possible to keep referring to this value as 'the x that makes $P[x]$ true', this is a very cumbersome name and also one that could be ambiguous if there were more than one such x, so a new constant c is introduced. c must be new since all that is known about it is that $P[c]$ is true (and if the quantifier is typed that c is of type t). If c were not a new constant, then the proof of Q might inadvertently use some additional properties that were true of some values but not all, and it could be that the 'x that makes $P[x]$ true' was one of those values for which these additional properties were not true.

Some more examples

In this section we look at some typical examples involving sentences with quantifiers.

Show $\exists y.\ \forall x.\ P(x,y) \vdash \forall u.\ \exists v.\ P(u,v)$ (Figure 17.7)
'If there is some y that makes $P(x,y)$ true for all x, then for every u there is some v (the same one for each case) that makes $P(u,v)$ true.'

The first two steps, $\forall\mathcal{I}$ and $\exists\mathcal{E}$, are automatic but could easily have been in the opposite order. Once a and b have been introduced there are enough clues in the proof so far (lines 1—3 and 5—7) to fill in the gap. Notice that the reverse deduction is not valid:

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

	1	$\exists y.\ \forall x.\ P(x,y)$	
$b\forall\mathcal{I}$	2		
$a\exists\mathcal{E}$	3	$\forall x.\ P(x,a)$	
	4	$P(b,a)$	$\forall\mathcal{E}(3)$
	5	$\exists v.\ P(b,v)$	$\exists\mathcal{I}(4)$
	6	$\exists v.\ P(b,v)$	$\exists\mathcal{E}(1)$
	7	$\forall u.\ \exists v.\ P(u,v)$	$\forall\mathcal{I}$

Figure 17.7 $\exists y.\ \forall x.\ P(x,y) \vdash \forall u.\ \exists v.\ P(u,v)$

In the next example, shown in Figure 17.8, lines 1—3 form the initial data. The data include a commonly occurring pattern of quantifiers — $\forall x.\ \exists y.$ Each time the $\forall\mathcal{E}$ rule is applied to a sentence such as $\forall x.\ \exists y.\ \mathit{likes}(x,y)$, the $\exists\mathcal{E}$ rule can be applied to generate a new constant. In turn, the new constant can be used in another application of $\forall\mathcal{E}$, which generates yet another new constant, and so on. In this case only one round is needed. Also, note that as B must be new it cannot be A. After that, the rest can be filled in fairly easily. NOTE: If a sentence has the form $Qx.Qy.\ [\cdots]$, where Q is either $\forall$ or $\exists$, then, usually, you will want to eliminate both the quantifiers in one elimination step or introduce them in one introduction step. This is quite acceptable and the two steps together are still labelled by $\forall\mathcal{E}$, $\exists\mathcal{E}$, $\exists\mathcal{I}$ or $\forall\mathcal{I}$ (and not by $\forall\forall\mathcal{E}$, for example!)

Show $\forall x.\ \forall y : num.\ [(\exists z : num.\ x^z = y) \rightarrow R(x,y)] \vdash \forall w : num.\ R(w,w)$ (Figure 17.9). Here, there are two lines where checks must be made that the terms being substituted are of the correct type. The information at line