Qualifiers and types

Often, a phrase that is to be translated using a universal quantifier is about a certain type of thing rather than about all things and so you want to *qualify* the quantifier. In the case of a universal quantifier this is done using an *implication*. For instance, 'all rational people abhor violence', or 'a rational person abhors violence', first becomes

$$\forall(\text{rational})x.\ [x \text{ abhors violence}]$$

where 'rational' is called a *qualifier*. This translates to

$$\forall x.\ [\mathit{rational}(x) \rightarrow \mathit{abhors\text{-}violence}(x)]$$

If the quantification is existential then a *conjunction* is used to link the main part with the qualifying part. For example if you want to make it certain that Mary likes people in 'Mary likes someone who likes logic', you could first write

$$\exists(\text{person who likes logic})y.\ \mathit{likes}(\mathit{Mary}, y)$$

and then

$$\exists y.\ [\mathit{person}(y) \wedge \mathit{likes}(y, \mathit{logic}) \wedge \mathit{likes}(\mathit{Mary}, y)]$$

Notice the way 'who' links the conjuncts together.

Another rule of thumb is therefore

> Get the structure of the sentence correct before dealing with the qualifiers.

Qualifiers can always be translated using $\rightarrow$ or $\wedge$ as appropriate. However, their use is quite convenient and so we will introduce a notation for them and write $\forall x : \mathit{typename}.\ [\cdots]$ or $\exists x : \mathit{typename}.\ [\cdots]$ and call the quantifiers *typed quantifiers*.

The notation is most often used for standard qualifiers, sometimes referred to as 'types', and sentences using it can always be rewritten with the type property made explicit. *Standard qualifiers* include persons, numbers (integers, reals, etc.) strings, times, lists, enumerated sets, etc.

For example,

$$\forall x : \mathit{time}.\ \forall y : \mathit{time}.\ [x \geq y \rightarrow \mathit{after}(x, y)]$$

would be shorthand for

$$\forall x \forall y.\ [\mathit{time}(x) \wedge \mathit{time}\ (y) \rightarrow (x \geq y \rightarrow \mathit{after}(x, y))]$$

Standard types are used extensively in writing program specifications, and they correspond to the various data structures such as list, num, etc., used in programs.

Earlier, we indicated that a sentence $\forall x.\ P[x]$ is true iff *every* sentence $P[t]$ that can be obtained from $P[x]$ by substituting a value t for every occurrence of x in $P[x]$ is true.

For example, 'all programs that work terminate', which in logic is

$$\forall x.\ [program(x) \rightarrow (works(x) \rightarrow terminates(x))]$$

is true if each sentence obtained by substituting a value for x is true. It is true if all sentences of the kind

$$program(quicksort) \rightarrow (works(quicksort) \rightarrow terminates(quicksort))$$
$$program(quacksort) \rightarrow (works(quacksort) \rightarrow terminates(quacksort))$$
$$program(Hessam) \rightarrow (works(Hessam) \rightarrow terminates(Hessam))$$

etc., are true. If the value t substituted is a program, so that $program(t)$ is true, the resulting sentence

$$program(t) \rightarrow (works(t) \rightarrow terminates(t))$$

is true if $works(t) \rightarrow terminates(t)$ is true. If the value t makes $program(t)$ false (that is, is not a program) then the resulting sentence is also true. In practice, we evaluate the truth of a sentence in a situation in which the values to be substituted for x are fixed beforehand. For example, they could be {all programs written by me}, {programs} or even {names of living persons}.

When qualified quantifiers are used they are suggestive of the range of values that should be substituted in order to test the truth of a sentence. The sentence 'All programs that work terminate' would become

$$\forall x : program.\ [works(x) \rightarrow terminates(x)].$$

and it is suggestive that the only values we should consider for x are those that name programs. As our analysis above showed, these are exactly the values that are useful in showing that the sentence is true.

Similarly, if instead the sentence had been 'Some programs that terminate work', which in logic is

$$\exists x.\ [program(x) \wedge works(x) \wedge terminates(x)]$$

then it would be true as long as *at least one* of the sentences obtained by substituting terms t for x were true. There is no point in trying values of t for which $program(t)$ is false for they cannot make the sentence true. This is suggested by the typed quantifier version

$$\exists x : program.\ [works(x) \wedge terminates(x)].$$

Even so, a difficulty may arise. Consider the statement

'Every integer is smaller than some natural number.'

which in logic is

$$\forall x : integer.\ \exists u : nat.\ [x < u].$$

This time there are an infinite number of sentences to consider, one for each integer. How can you check them all? Of course, you cannot check them all individually *and* finish the task. Instead, you would consider different cases. For example, you may consider two cases here, $x < 0$ and $x \geq 0$. Then, all negative integers are considered *at once*, as are all natural numbers. For the first case the sentence is true by taking $u = 0$, for 0 is a natural number and it is greater than any negative integer; in the second case $x+1$ is a natural number and will do for u. Sometimes, therefore, we have to use a proof to justify the truth of a sentence; we look at proof in the next two chapters.

Some paradoxes

Generally, the need for a universal quantifier is indicated by the presence of such words as all, every, any, anyone, everything, etc., and the words 'someone', 'something' indicate an existential quantifier, but it can happen that 'someone' corresponds to $\forall$. This phenomenon is most likely in connection with $\rightarrow$.

To see how this might happen, consider 'if someone is tall then the door frame will be knocked', which translates to

$$[\exists x.\ tall(x)] \rightarrow door\text{-}knocked.$$

'Someone' has become $\exists$ here, just as you would expect. But note that there is an equivalent translation using $\forall$. The original sentence could be rephrased as 'for anyone, if they are tall then the doorframe will be knocked', which becomes

$$\forall x.\ [tall(x) \rightarrow knocked(doorframe)]$$

Hence, in this example, 'someone' can possibly become $\forall$.

Now consider 'if someone is tall then he will bump his head'. This time the pronoun is linked to 'someone' across the implication and you have to deal with the quantification first. The only translation is

$$\forall x.\ [tall(x) \rightarrow bumphead(x)]$$

so that 'someone' has to become $\forall$.

15.6 Introducing equivalence

Often, English sentences can be translated into more than one equivalent formula in logic. For example, 'if Steve is a vegetarian then he does not eat chicken' might be translated directly as $vegetarian(Steve) \rightarrow \neg eat(Steve, chicken)$ but it could also be paraphrased as 'Steve is not both a vegetarian and a chicken-eater', which translates to $\neg(vegetarian(Steve) \wedge eat(Steve, chicken))$. The two logic sentences are *equivalent* and any conclusion that follows from using one form also follows from using the other. You will come across many useful equivalences and a selection is presented in Appendix B. We write $A \equiv B$ if A and B are equivalent. Two sentences are said to be *equivalent* ($\equiv$) iff they are both true in exactly the same situations. An important property of equivalent sentences is that they may safely be substituted for each other in any longer sentence without affecting the meaning of that sentence.

For example, if $A = S \vee T$ and $B = T \vee S$ then A is equivalent to B. If $E[A]$ is the sentence $S \vee T \rightarrow U$ $(=A \rightarrow U)$ then we can substitute B for A giving the sentence $E[B](= B \rightarrow U)$, or $T \vee S \rightarrow U$. We have $E[A] \equiv E[B]$. S and T can themselves be any sentence; for example, if $S = P \wedge Q$ and $T = R \vee \neg P$ then $(P \wedge Q) \vee (R \vee \neg P) \equiv (R \vee \neg P) \vee (P \wedge Q)$.

In general, then, if $A \equiv B$ then $E[A] \equiv E[B]$, where A, B, E are any sentences with no variable occurrences. $E[A]$ denotes that A occurs in E and $E[B]$ denotes the result of substituting B for A in none or more of those occurrences. This is so because if A evaluates to *tt* in a situation then so will B as they are equivalent, and the $E[A]$ and $E[B]$ have the same value. In particular, $E[A]$ could be just the sentence A, so $E[B]$ is the sentence B and B can be used in place of A.

Equivalences are frequently used, as it may be that one form of a sentence is more convenient than another in some derivation. More discussion can be found in Section 18.4, where we consider relaxing the condition on A and B.

Equivalences can be used in 'algebraic reasoning'. For example,

$$
\begin{aligned}
&(P \wedge Q) \wedge R \\
&\equiv \neg\neg((P \wedge Q) \wedge R), \text{ since } \neg\neg X \equiv X \\
&\equiv \neg(\neg(P \wedge Q) \vee \neg R), \text{ since } \neg(X \wedge Y) \equiv \neg X \vee \neg Y \\
&\equiv \neg(\neg P \vee \neg Q \vee \neg R)
\end{aligned}
$$

that is, $(P \wedge Q) \wedge R \equiv \neg(\neg P \vee \neg Q \vee \neg R)$.

As another example, the two sentence forms $A \vee (S \vee T)$ and $(A \vee S) \vee T$ are equivalent; that is, $\vee$ is an *associative* operator and hence the parentheses can be omitted. The operator $\wedge$ behaves similarly. Using this fact you can show easily that any number of sentences all disjoined by $\vee$, or all conjoined by $\wedge$, can be freely parenthesized; for example, $Q \vee R \vee S \vee T \equiv Q \vee (R \vee S \vee T) \equiv (Q \vee R) \vee (S \vee T) \equiv (Q \vee R \vee S) \vee T$.

If a sentence has a form which makes it always true it is called a *tautology*; for example $A \vee \neg A$ is a tautology. A sentence that is always false is called a *contradiction*, or falsehood, for example $A \wedge \neg A$. Both tautologies and contradictions will play an important role in the reasoning steps that we shall be introducing.

15.7 Some useful predicate equivalences

In this section we look briefly at some useful equivalences using quantified sentences.

The equivalences in Appendix B are schemes in which the constituents represent sentence forms. For example, $F(x)$ indicates a constituent sentence in which x occurs, whereas S (without an x) indicates a constituent sentence in which x does not occur. An instance of a scheme such as

$\forall x.\ [S \wedge F(x)] \equiv S \wedge \forall x.\ F(x)$ is obtained by replacing all occurrences of S and $F(x)$ by appropriate sentences, for example S could be $\exists y.\ G(y)$ and $F(x)$ could be $P(x, a) \vee Q(x, b)$, where a and b are constants. The variables x and y are like formal parameters and can be renamed. So $\neg\forall u.\ F(u, a)$ is an instance of the scheme $\neg\forall x.\ F(x)$ and rewrites to $\exists u.\ \neg F(u, a)$.

NOTE: $\exists x.\ \forall y.\ F(x, y)$ is *not* equivalent to $\forall y.\ \exists x.\ F(x, y)$. In order to help you to remember this one, find an interpretation for F that distinguishes clearly, for you, between the two sentences. For example, you could interpret F as 'father', so that the first sentence translates into 'there is some x that is the father of everyone' and the second into 'for each person y there is some x that is the father of y'.

An instance of the important equivalence $\exists x.\ F(x) \to B \equiv \forall x.\ [F(x) \to B]$ is used in the following:

$$\begin{aligned} &\exists c.\ \mathit{mother}(\mathit{Pam}, c) \to \mathit{parent}(\mathit{Pam}) \\ &\equiv \forall c.\ [\mathit{mother}(\mathit{Pam}, c) \to \mathit{parent}(\mathit{Pam})] \end{aligned}$$

The occurrence of $\exists x.\ F(x)$ is $\exists c.\ \mathit{mother}(\mathit{Pam}, c)$, in which the bound variable x is renamed to c, and of B is $\mathit{parent}(\mathit{Pam})$. Notice that c does not occur in $\mathit{parent}(\mathit{Pam})$.

It is also true that equivalent forms of sentences involving variables and quantifiers can be substituted for one another in any context as in the following example. After reading Section 18.4 you will be able to prove this.

In the following example the equivalences used and the scheme occurrences are not given. It is left as an exercise to list the equivalences at each step.

No student works all the time $\equiv$ All students fail to work some of the time.

$$\begin{aligned} &\neg\exists s.\ [\mathit{student}(s) \wedge \forall t.\ [\mathit{time\text{-}period}(t) \to \mathit{works\text{-}at}(s, t)]] \\ &\equiv \forall s.\ [\neg\mathit{student}(s) \vee \neg\forall t.\ [\mathit{time\text{-}period}(t) \to \mathit{works\text{-}at}(s, t)]] \\ &\equiv \forall s.\ [\neg\mathit{student}(s) \vee \exists t.\ [\mathit{time\text{-}period}(t) \wedge \neg\mathit{works\text{-}at}(s, t)]] \\ &\equiv \forall s.\ [\mathit{student}(s) \to \exists t.\ [\mathit{time\text{-}period}(t) \wedge \neg\mathit{works\text{-}at}(s, t)]] \end{aligned}$$

The equivalences also hold if the quantifiers are typed. The above example then becomes

$$\begin{aligned} &\neg\exists s : \mathit{student}.\ \forall t : \mathit{time\text{-}period}.\ [\mathit{works\text{-}at}(s, t)] \equiv \\ &\forall s : \mathit{student}.\ \exists t : \mathit{time\text{-}period}.\ [\neg\mathit{works\text{-}at}(s, t)] \end{aligned}$$

and the transformation is simpler.

15.8 Summary

- Logic uses connectives to express the logical structure of natural language.
- The syntax and meanings of propositional logic follow the principles of algebra.

- Atoms consist of predicates which have arguments called terms. Terms can be constants, or function symbols and their arguments.
- For reference, the meanings can be summarized using a *truth table*. For two propositions there are four different classes of situation: $\{tt, tt\}$, $\{tt, ff\}$, $\{ff, tt\}$, $\{ff, ff\}$. Each row of the truth table gives one situation.

A	$\neg A$
tt	ff
ff	tt

A	B	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
tt	tt	tt	tt	tt	tt
tt	ff	ff	tt	ff	ff
ff	tt	ff	tt	tt	ff
ff	ff	ff	ff	tt	tt

  For example, from this truth table it can be seen that $A \vee B$ is true unless both A and B are false.
- To facilitate translation from English into logic, typed quantifiers are introduced.
- The informal meaning of a sentence involving a quantifier is

  $\forall x.\ P[x]$ is true iff every sentence $P[t]$ obtained by substituting t for x, where t is taken from a suitable range of values, is true.
  $\forall x.\ P[x]$ is false if some $P[t]$ is false.

  $\exists x.\ P[x]$ is true iff some sentence $P[t]$ obtained by substituting t for x, where t is taken from a suitable range of values, is true.
  $\exists x.\ P[x]$ is false if no sentence $P[t]$ is true.
- Equivalent sentences can be substituted for one another.

15.9 Exercises

1. Suggest some predicate and function symbols to express the following propositions: Mary enjoys sailing; Bill enjoys hiking; Mabel is John's daughter; Ann is a student and Ann is Mabel's daughter
2. Translate the following sentences into logic. First get the sentence structure correct (where the $\wedge$, $\vee$, etc., go) and then structure the atoms, for example Frank likes grapes could become *likes*(*Frank*, *grapes*).

   (a) If there is a drought, standpipes will be needed.

   (b) The house will be finished only if the outstanding bill is paid or if the proprietor works on it himself.

   (c) James will work hard and pass, or he belongs to the drama society.

   (d) Frank bought grapes and either apples or pears.

   (e) Janet likes cricket, but she likes baseball too.

   (f) All out unless it snows!

3. Translate the following into logic as faithfully as possible:
   (a) All red things are in the box.
   (b) Only red things are in the box.
   (c) No animal is both a cat and a dog.
   (d) Anyone who admires himself admires someone.
   (e) Every prize was won by a chimpanzee.
   (f) One particular chimpanzee won all the prizes.
   (g) Jack cannot run faster than anyone in the team.
   (h) Jack cannot run faster than everyone in the team.
   (i) A lecturer is content if she belongs to no committees.
   (j) All first year students have a programming tutor.
   (k) No student has the same mathematics tutor and programming tutor.
   (l) A number is a common multiple of two numbers if each divides it.
   (m) Mary had a little lamb, its fleece was white as snow. And everywhere that Mary went her lamb was sure to go!
4. (a) Let A be tt, B be tt, C be ff. Which of the following sentences are true and which are false?
      i. $((A \rightarrow B) \rightarrow \neg B)$
      ii. $((\neg A \rightarrow (\neg B \wedge C)) \vee B)$
      iii. $((((A \vee \neg C) \wedge \neg B) \rightarrow A) \rightarrow (\neg B \wedge \neg C))$
   (b) If A is ff, B is ff and C is tt, which of the sentences in part (a) are true and which are false?
   (c) If A is ff, B is tt and C is tt, which of the sentences in part (a) are true and which are false?
5. We mentioned, but did not prove, that associativity allows you to omit parentheses if all the connectives are $\vee$ or $\wedge$. Explain how associativity is used to show the equivalence of $((Q \vee R) \vee S) \vee T$ and $Q \vee (R \vee (S \vee T))$.
6. Show that the following are equivalent forms by considering all different situations and showing that the pairs of sentences have the same truth value in all of them. For example, for the equivalence $P \wedge ff \equiv ff$ there are two situations to consider — $P = tt$ and $P = ff$. When $P = ff$, $P \wedge ff = ff \wedge ff = ff$, and when $P = tt$, $P \wedge ff = tt \wedge ff = ff$. In both cases the sentence is ff. For the example $P \wedge Q \equiv \neg(P \rightarrow \neg Q)$ there are four situations to consider which can be tabulated as

P	Q	$P \wedge Q$	$P \rightarrow \neg Q$	$\neg(P \rightarrow \neg Q)$
tt	tt	tt	ff	tt
tt	ff	ff	tt	ff
ff	tt	ff	tt	ff
ff	ff	ff	tt	ff

You can see that the two sentences have the same value in all four situations and so are equivalent.

(a) $P \vee Q \equiv (P \rightarrow Q) \rightarrow Q$
(b) $P \wedge Q \equiv \neg(P \rightarrow \neg Q)$
(c) $P \leftrightarrow Q \equiv Q \leftrightarrow P$ (that is, $\leftrightarrow$ is commutative)
(d) $P \leftrightarrow (Q \leftrightarrow R) \equiv (P \leftrightarrow Q) \leftrightarrow R$ (that is, $\leftrightarrow$ is associative)
(e) $P \leftrightarrow Q \equiv \neg P \leftrightarrow \neg Q$
(f) $\neg(P \leftrightarrow Q) \equiv \neg P \leftrightarrow Q$
(g) $P \rightarrow (Q \rightarrow R) \equiv P \wedge Q \rightarrow R$
(h) $P \rightarrow (Q \wedge \ R) \equiv (P \rightarrow Q) \wedge \ (P \rightarrow R)$

7. Show that $R \equiv S$ iff $R \leftrightarrow S$ is a tautology. (HINT: consider the possible classes of situations for $R \leftrightarrow S$).
8. Discuss how you would decide the truth or falsity of the sentences below in the given situations. Also decide which are true in the given situations and which are false (if feasible). The situation indicates the possible values that can be substituted for the bound variables.

   (a) All living creatures, animal or not:

      i. $\forall x.\ [animal(x) \rightarrow \exists y.\ [animal(y) \wedge (eats(x,y) \vee eats(y,x))]]$
      ii. $\exists u.\ [animal(u) \wedge \forall v.\ [animal(v) \rightarrow eats(v,u)]]$
      iii. $\forall y.\ \forall x.\ [animal(x) \wedge animal(y) \rightarrow (eats(x,y) \leftrightarrow eats(y,x))]$
      iv. $\neg\exists v.\ [animal(v) \wedge \forall u.\ [animal(u) \rightarrow (\neg eats(u,v))]]$

   (b) There are three creatures Cat, Bird and Worm. Cat eats all three, Worm is eaten by all three and Bird only eats Worm. Use the sentences (i) through (iv) of part (a) of this question.

   (c) The universe of positive integers:

      i. $\exists x.\ [x$ is the product of two odd integers$]$
      ii. $\forall x.\ [x$ is the product of two odd integers$]$
      iii. $\forall x.\ \exists y.\ [y > x]$
      iv. $\forall x.\ \forall y.\ [x \times y \geq x]$

9. By using the appropriate equivalences and translation of $\forall x : T.\ P[x]$ into $\forall x.\ [\textit{is-}T(x) \rightarrow P[x]]$ and $\exists x : T.\ P[x]$ into $\exists x.\ [\textit{is-}T(x) \wedge P[x]]$, show that $\forall x : T.\ [P[x] \rightarrow S] \equiv (\exists x : T.\ P[x]) \rightarrow S$.
10. Show that the following pairs of sentences are equivalent by using equivalences. State the equivalences you use at each step:

   (a) $\forall x.\ [\neg\forall y.\ [woman(y) \rightarrow \neg dislikes(x,y)] \rightarrow dislikes(Jane,x)]$
   and $\forall x.\ [\exists y.\ [woman(y) \wedge dislikes(x,y)] \rightarrow dislikes(Jane,x)]$

   (b) $\neg\exists x.\ [Martian(x) \wedge \neg dislikes(x, Mary) \wedge age\text{-}more\text{-}than\text{-}25(x)]$
   and $\forall x.\ [Martian(x) \wedge age\text{-}more\text{-}than\text{-}25(x) \rightarrow dislikes(x, Mary)]$

Chapter 16

Natural deduction

16.1 Arguments

Now that you can express properties of your programs in logic we consider how to reason with them to form *correct* proofs. Initially, we will look at reasoning with sentences that do not include any quantifiers.

The method we use is called natural deduction and it formalizes the approach to reasoning embodied in the 'argument form'

> 'This is so, that is so, so something else is so and hence something else, and hence we have shown what we wanted to show.'

An argument leads from some statements, called the premisses, to a final statement, called the conclusion. It is valid if whenever circumstances make the premisses true then they make the conclusion true as well. The only way in which the conclusion of a *valid* argument can be rejected is by rejecting the premisses (a useful way out).

We justify a potential argument by putting it together from small reasoning steps that are all known to be valid. We write $A \vdash B$ (pronounced 'A proves B') to indicate that B can be derived from A using some correct rules of reasoning. So, if we can find a derivation, then $A \vdash B$ is true.

Schematically:

$$P_1 \vdash P_2, \quad \{P_1, P_2\} \vdash P_3 \quad \cdots \quad \{P_1, P_2, \ldots, P_{n-1}\} \vdash P_n.$$

The steps are supposed to be so simple that there is no doubting the validity of each one.

The following is a valid argument:

1. If Hessam's program is less than 10 lines long then it is correct.
2. Hessam's program is not correct.
3. Therefore Hessam's program is more than 10 lines long.

The first two lines are the premisses and the last the conclusion. A derivation of the conclusion in this case is the following: suppose Hessam's program is less than 10 lines long; then it is correct. But this contradicts the second premiss so we conclude that Hessam's program is more than 10 lines long. These reasoning steps mean that $1,\ 2 \vdash 3$.

Sometimes, we may be tempted to use invalid reasoning steps, in which the conclusion does not always have to be the case even if the premisses are true. Any justification involving such steps will not be correct.

The following is an invalid argument:

> If I am wealthy then I give away lots of money.
> I give away lots of money.
> Therefore I am wealthy.

The reasoning is not valid because from the premisses you cannot derive the conclusion; the premisses could be true and yet I could be poor and generous.

If $A \vdash B$ then the sentence $A \rightarrow B$ is a tautology because whenever A is true B must be true also. The various tautologies such as $A \wedge B \rightarrow A$ each give rise to simple and valid arguments. This one yields the valid argument $A \wedge B \vdash A$.

An informal example

The natural deduction rules to be introduced in this chapter are quite formal. This is a good thing for it enables a structure to be imposed on a proof so that you can be confident it is valid. When you are quite sure of the structure imposed by the rules it is possible to present proofs in a more relaxed style using English. Typical of such an English proof is the following proof of the valid argument:

> If Chris is at home then he is working.
> If Ann is at work then she is working.
> Ann is at work or Chris is at home.
> Therefore someone is working.

A justification of this argument might follow the steps: to show someone is working, find a person who is working — there are two cases to consider: if Ann is at work, she is working and if Chris is at home, he is working. Either way, someone is working.

16.2 The natural deduction rules

About the rules

There are two kinds of rule. The first kind tells us how to reason using a sentence with a given connective, that is, how to exploit a premiss. For example, from $A \wedge B$ we can deduce each of A and B. The second kind tells us how to deduce a sentence with a given connective, that is, how to prove a conclusion. For example, to deduce $A \wedge B$ we must prove both A and B. The first kind are called *elimination rules* and the second are called *introduction rules*. They are labelled $\wedge\mathcal{E}$ (pronounced *and elimination*), $\vee\mathcal{E}$, $\wedge\mathcal{I}$ (pronounced *and introduction*), $\vee\mathcal{I}$, etc.

If a formula is derived using the rules, the notation

$$\vdash \langle \mathit{formula} \rangle$$

will be used. When initial data is needed to prove a formula the notation is

$$\langle \mathit{assumptions} \rangle \vdash \langle \mathit{formula} \rangle.$$

$S \vdash C$ is called a *sequent* and can be read as:

A proof exists of goal sentence C from data sentences S.

The initial data sentences S are placed at the top of the proof and the conclusion C is placed at the bottom. The actual proof goes in the middle. Frequently, a proof will consist of subproofs, which will be written inside boxes.

As you read a proof from top to bottom, you see more and more consequences of the earlier sentences. However, that is not the way in which a proof is constructed in the first place. As you will see, when proving something we can work both forwards from the data and backwards from the conclusion so that the middle part is not usually filled in straight from data to conclusion. When a proof is written 'in English' it is written to reflect this 'construction order' of the proof.

Each of the rules will be presented in the following style:

$$\frac{\text{one or more antecedents}}{\text{a conclusion} \qquad \text{(rule name)}}$$

'Antecedent' just means 'something that has gone before'.

Often, it is just an earlier sentence, though sometimes it is a bigger chunk of proof. The rules can either be read *downwards* — from the antecedents the conclusion can be derived, or *upwards* — to derive the conclusion, you must derive the antecedents. We will frequently omit the line between the antecedents and the conclusion.

$\wedge$-introduction ($\wedge\mathcal{I}$) and $\wedge$-elimination ($\wedge\mathcal{E}$) rules

The two rules of this section, $\wedge\mathcal{I}$ and $\wedge\mathcal{E}$, correspond closely to everyday deduction.

The first rule is $\wedge\mathcal{I}$:

> From each of $P_1, \ldots, P_n$ as data or derived sentences, conclude $P_1 \wedge \ldots \wedge P_n$ or, to give a proof of $P_1 \wedge \ldots \wedge P_n$, derive proofs for $P_1, \ldots, P_n$

The proof is structured using *boxes*:

$$\begin{array}{l}
\boxed{\begin{array}{c}\quad\vdots\quad\\ P_1\end{array}} \;\cdots\; \boxed{\begin{array}{c}\quad\vdots\quad\\ P_n\end{array}} \\
\quad P_1 \wedge \ldots \wedge P_n \qquad (\wedge\mathcal{I})
\end{array}$$

The boxes are introduced to contain the proofs of $P_1, \ldots, P_n$ prior to deriving $P_1 \wedge \ldots \wedge P_n$. The vertical dots indicate the proof that is to be filled in. There is one box to contain the proof for each of P_1 to P_n. The use of the $\wedge\mathcal{I}$ rule is automatic — there is a standard plan which you always use when proving $P_1 \wedge \ldots \wedge P_n$.

When a proof is presented, it is usually read from the top to the bottom, but when you are actually proving something, you may work backwards from the conclusion. So, in a proof, you will probably read an application of $\wedge\mathcal{I}$ downwards, but when you have to prove $P \wedge Q$, you ask 'how do I do it?', and the answer is by proving P and Q separately. We can say that you work backwards from the conclusion, deriving a new conclusion to achieve.

The second rule is $\wedge\mathcal{E}$:

> from data or derived sentences $P_1 \wedge \ldots \wedge P_n$ conclude any of $P_1, \ldots, P_n$, or
>
> $$\frac{P_1 \wedge \ldots \wedge P_n}{P_i \quad (\wedge\mathcal{E})}$$
>
> for each of P_i, $i = 1, \ldots, n$.

This time the rule is used exclusively in a forward direction, deriving new data.

Figure 16.1 contains the first steps in a proof of $A \wedge B \vdash B \wedge A$. If we need to refer to lines in proofs then each row in the proof will be labelled for reference. In the diagram, the given sentence $A \wedge B$ is initial data and is placed at the start of the deduction, and the conclusion, or goal, is $B \wedge A$, which appears at the end. Our task is to fill in the middle.

There are now two ways to proceed — either forwards from the data or backwards from the goal. In general, a natural deduction derivation involves working in both directions. Here, as soon as you see the $\wedge$ in the conclusion,

1	$A \wedge B$	
2	$\vdots$	
3	$B \wedge A$	

Figure 16.1

1	$A \wedge B$		
2	$\vdots$	$\vdots$	
3	B	A	
4	$B \wedge A$		$\wedge\mathcal{I}$

Figure 16.2

think (automatic step) $\wedge\mathcal{I}$ and prepare for it by making *the preparation* as in Figure 16.2. Working backwards from the conclusion is generally applicable when introduction rules are to be used. This example will require the use of the $\wedge\mathcal{I}$ rule. The boxes are introduced to contain the subproofs of A and B. It needs a tiny bit of ingenuity to notice that each of the subgoals can now be derived by $\wedge\mathcal{E}$ from the initial data $A \wedge B$ by working forwards. The completed proof appears in Figure 16.3. Lesson — the $\wedge\mathcal{I}$ step is automatic

1	$A \wedge B$				
2	B	$\wedge\mathcal{E}(1)$	A	$\wedge\mathcal{E}(1)$	
3	$B \wedge A$				$\wedge\mathcal{I}$

Figure 16.3 $A \wedge B \vdash B \wedge A$

— to prove $A \wedge B$ you must prove A and B separately. But to use $\wedge\mathcal{E}$ requires ingenuity — which conjunct should you choose?

An alternative proof construction for $A \wedge B \vdash B \wedge A$ is shown in Figure 16.4. It works forwards only — first derive each of A and B from $A \wedge B$ and then derive $B \wedge A$.

You can see that these two rules are valid, from the definition of true sentences of the form $P \wedge Q$ given in Chapter 15. For if $P \wedge Q$ is true then so must each of P and Q be ($\wedge\mathcal{E}$), and vice versa ($\wedge\mathcal{I}$).

1	$A \wedge B$	
2	A	$\wedge\mathcal{E}(1)$
3	B	$\wedge\mathcal{E}(1)$
4	$B \wedge A$	$\wedge\mathcal{I}$

Figure 16.4 Another proof of $A \wedge B \vdash B \wedge A$

$\vee$-elimination ($\vee\mathcal{E}$) and $\vee$-introduction ($\vee\mathcal{I}$) rules

The $\vee$-elimination rule is frequently used in everyday deduction and is often called a case analysis — a disjunction $P_1 \vee P_2$ (say) represents two possible cases and in order to conclude C, C should be proven from both cases, so that it is provable whichever case actually pertains. It can be generalized to $n > 2$ arguments and is

$\vee\mathcal{E}$ If C can be derived from each of the separate cases $P_1, \ldots, P_n$, then from $P_1 \vee \ldots \vee P_n$, derive goal C.

$$
\begin{array}{r|c|c|c|}
\cline{2-4}
 & P_1 & & P_n \\
 & \vdots & \ldots & \vdots \\
P_1 \vee \ldots \vee P_n & C & & C \\
\cline{2-4}
\end{array}
$$

$$C \qquad (\vee\mathcal{E})$$

There is one box for each of P_i, $i = 1, n$.

Each box that is part of the preparation for the $\vee\mathcal{E}$ step represents a subproof for one of the cases, and contains as an additional assumption the disjunct P_i that represents its case. The assumptions P_i are only available inside the box and their use corresponds to the English phrase 'suppose that $P_i \ldots$'. Once the proof leaves the box we forget our supposition. Hence the box says something significant: P_i is true in here.

The $\vee\mathcal{I}$ rule is

$\vee\mathcal{I}$ From any one of $P_1, \ldots, P_n$ derive $P_1 \vee \ldots \vee P_n$

$$\frac{P_i}{P_1 \vee \ldots \vee P_n \qquad (\vee\mathcal{I})}$$

for each of P_i, $i = 1, \ldots, n$.

The $\vee$-introduction rule is usually used in a backward direction — in order to show $P \vee Q$ one of P or Q must be shown. In the forward direction the rule is rather weak — if P is known then it does not seem very useful to derive the weaker $P \vee Q$ (unless such a deduction is needed to obtain a particular desired sentence, as in the next example). This rule, too, can be generalized to $n > 2$ arguments.

This time, the $\vee\mathcal{E}$ rule is automatic, whereas the $\vee\mathcal{I}$ rule is the one that requires ingenuity — when proving $P_1 \vee \ldots \vee P_n$ which disjunct should we choose to prove?

In the next example, a proof of $A \wedge (B \vee C) \vdash (A \wedge B) \vee (A \wedge C)$, we illustrate how a proof might be found. The first step is to place the initial assumption at the top and the conclusion at the bottom as in Figure 16.5. Now, where

$$\begin{array}{l} A \wedge (B \vee C) \\ \vdots \\ (A \wedge B) \vee (A \wedge C) \end{array}$$

Figure 16.5

do we go from here? There are no automatic steps — $\wedge\mathcal{E}$, and $\vee\mathcal{I}$ need ingenuity. Can we obtain the conclusion by $\vee\mathcal{I}$? Does either of the sentences $A \wedge B$ or $A \wedge C$ follow from the premiss? A little insight says *no*, so try $\wedge\mathcal{E}$ on $A \wedge (B \vee C)$ — it is not so difficult and the result is given in Figure 16.6. Now an automatic step is available — exploit $B \vee C$ by $\vee\mathcal{E}$ (case analysis).

$$\begin{array}{ll} A \wedge (B \vee C) & \\ A & \wedge\mathcal{E} \\ B \vee C & \wedge\mathcal{E} \\ \vdots & \\ (A \wedge B) \vee (A \wedge C) & \end{array}$$

Figure 16.6

The preparation is given in Figure 16.7. Look at the left-hand box. There are no automatic steps, but look, we can prove $A \wedge B$ by using B and then use $\vee\mathcal{I}$ to show $(A \wedge B) \vee (A \wedge C)$. Similarly in the right-hand box, proving $A \wedge C$. The complete proof is given in Figure 16.8. It is often the case that a disjunctive conclusion can be derived by exploiting a disjunction in the data. Sometimes, an inspired guess can yield a result, as inside the boxes of the example.

$A \wedge (B \vee C)$		
A		$\wedge\mathcal{E}$
$B \vee C$		$\wedge\mathcal{E}$
B	C	
$\vdots$	$\vdots$	
$(A \wedge B) \vee (A \wedge C)$	$(A \wedge B) \vee (A \wedge C)$	
$(A \wedge B) \vee (A \wedge C)$		$\vee\mathcal{E}$

Figure 16.7

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

Figure 16.8 $A \wedge (B \vee C) \vdash (A \wedge B) \vee (A \wedge C)$

As an example of how a box proof is translated into English, we will give the same proof in its more usual form.

Proposition 16.1 $A \wedge (B \vee C) \vdash (A \wedge B) \vee (A \wedge C)$

Proof Since $A \wedge (B \vee C)$, then A and $B \vee C$. Consider $B \vee C$: suppose B, then to show $(A \wedge B) \vee (A \wedge C)$ we have to show either $A \wedge B$ or $A \wedge B$. In this case we can show $A \wedge B$. On the other hand, suppose C. In that case we can show $A \wedge C$ and hence $(A \wedge B) \vee (A \wedge C)$. So in both cases we can show $(A \wedge B) \vee (A \wedge C)$. $\square$

From now on you will have to work through the examples in order to see how they are derived, as only the final stage will usually be given.

It is easy to see that the $\vee\mathcal{I}$ rule is valid; for $X \vee Y$ is true as long as either X or Y is. If $X \vee Y$ is true then we know only that either X is true or Y is true, but we cannot be sure which one is true. For the $\vee\mathcal{E}$ case, therefore, we must be able to show C from both so as to be sure that C must be true.

It is tempting to try to ignore the $\vee\mathcal{E}$ rule because it looks complicated. But you must learn it by heart! It is automatic — as soon as you see $\vee$ in a premiss you should consider preparing for $\vee\mathcal{E}$. Writing the conclusion in $n+1$ places seems odd at first, but this is what you must do. Each occurrence has a different justification; it is $\vee\mathcal{E}$ outside the boxes and other reasons inside.

There is a special case of $\vee\mathcal{E}$ in which the number of disjuncts is zero. A disjunction of n sentences says 'at least one of the disjuncts is true', but if $n = 0$ that is impossible. To represent an impossible sentence, a *contradiction*, we use the symbol $\perp$, which is pronounced *bottom* and is always false. If you look at $\vee\mathcal{E}$ when $n = 0$ you see that there are no cases to analyze and all you are left with is

$$\frac{\perp}{C} \quad (\perp\mathcal{E})$$

$\rightarrow$-elimination ($\rightarrow\mathcal{E}$) and $\rightarrow$-introduction ($\rightarrow\mathcal{I}$) rules

The first rule is

$\rightarrow\mathcal{E}$ (pronounced *arrow elimination*)

from P and $P \rightarrow Q$ derive Q.

$$\frac{P \qquad P \rightarrow Q}{Q} \quad (\rightarrow\mathcal{E})$$

It can be used both forwards from data and backwards from the conclusion. To work backwards, suppose the conclusion is Q, then any data of the form $P \rightarrow Q$ can be used to derive Q if P can be derived. So P becomes a new conclusion. In neither direction is the rule completely automatic — some ingenuity is needed. The $\rightarrow\mathcal{E}$ rule is commonly used in everyday arguments and is also referred to as *Modus Ponens*.

The second rule is

$\rightarrow\mathcal{I}$ from a proof of Q using the additional assumption P, derive $P \rightarrow Q$.

$$\boxed{\begin{array}{l} P \\ \vdots \\ Q \end{array}}$$

$$P \rightarrow Q \qquad (\rightarrow\mathcal{I})$$

The $\rightarrow\mathcal{I}$ rule appears at first sight to be less familiar. In common with other introduction rules $\rightarrow\mathcal{I}$ requires preparation — in this case, to derive

$P \to Q$, a box is drawn to contain the assumption P and the subgoal Q has to be derived in this box. The English form of $P \to Q$, 'if P then Q', indicates the proof technique exactly: if P holds then Q should follow, so assume P and show that Q does follow. Note that the box shows exactly where the temporary assumption is available. $\to\mathcal{I}$ is an automatic rule and is always used by working backwards from the conclusion.

The next example is to prove $A \wedge B \to C \vdash A \to (B \to C)$. The first steps in this example are automatic. First, a preparation is made to prove $A \to (B \to C)$, and then a second preparation is made to prove $B \to C$, both by $\to\mathcal{I}$. These result in Figure 16.9. There are then two possibilities —

$A \wedge B \to C$	
A	
B	
$\vdots$	
C	
$B \to C$	$\to\mathcal{I}$
$A \to (B \to C)$	$\to\mathcal{I}$

Figure 16.9

you can either use A and B to give $A \wedge B$ and hence C, or you can use $A \wedge B \to C$ to reduce the goal C to the goal $A \wedge B$.

The final proof is given in Figure 16.10. How might this proof appear in

1	$A \wedge B \to C$	
2	A	
3	B	
4	$A \wedge B$	$\wedge\mathcal{I}(2,3)$
5	C	$\to\mathcal{E}(1,4)$
6	$B \to C$	$\to\mathcal{I}$
7	$A \to (B \to C)$	$\to\mathcal{I}$

Figure 16.10 $A \wedge B \to C \vdash A \to (B \to C)$

English?

Proposition 16.2 $A \wedge B \rightarrow C \vdash A \rightarrow (B \rightarrow C)$

Proof To show $A \rightarrow (B \rightarrow C)$, assume A and show $B \rightarrow C$. To do this, assume B and show C. Now, to show C, show $A \wedge B$. But we can show $A \wedge B$ since we have assumed both A and B. □

The next three examples illustrate the use of the $\rightarrow\mathcal{E}$ and $\rightarrow\mathcal{I}$ rules. They also use the useful ✓ rule — if you want to prove A, and A is in the data, then you can just 'check' A.

$$\frac{A}{A \quad (\checkmark)}$$

Show $\vdash A \rightarrow A$

There is only one real step in this example, and no initial data (Figure 16.11).

1	A	
2	A	$\checkmark(1)$
3	$A \rightarrow A$	$\rightarrow\mathcal{I}$

Figure 16.11 $\vdash A \rightarrow A$

Show $A \vdash B \rightarrow A$

1	A	
2	B	
3	A	$\checkmark(1)$
4	$B \rightarrow A$	$\rightarrow\mathcal{I}$

Figure 16.12 $A \vdash B \rightarrow A$

Notice that the assumption B is not used inside the box (Figure 16.12).

Show $P \vee Q \vdash (P \rightarrow Q) \rightarrow Q$

In Figure 16.13 the preparation for $\rightarrow\mathcal{I}$ is made before that for $\vee\mathcal{E}$. If the preparation for using $P \vee Q$ were made before the preparation for the conclusion, then the latter preparation would have to be made twice within each of the boxes enforced by the preparation for $\vee\mathcal{E}$.

1	$P \vee Q$			
2	$P \to Q$			
3	P		Q	
4	Q	$\to\mathcal{E}(2,3)$	Q	$\checkmark(3)$
5	Q			$\vee\mathcal{E}$
6	$(P \to Q) \to Q$			$\to\mathcal{I}$

Figure 16.13 $P \vee Q \vdash (P \to Q) \to Q$

The validity of $\to\mathcal{E}$ is easy to see, for the truth of $P \to Q$ and P force Q to be true by the definition of $\to$. For the $\to\mathcal{I}$ rule, remember that $P \to Q$ is true if P is false, *or* if P and Q are both true. So, in case P is true we have to show Q as well.

Rules for negation

There are three rules for negation, two of which are special cases of earlier rules, whereas the third is new and does not conform to the introduction/elimination pattern. The rules are

$\neg\mathcal{I}$ If the assumption of P leads to a contradiction (written as $\bot$) then conclude $\neg P$
$\neg\mathcal{E}$ From P and $\neg P$ derive $\bot$
$\neg\neg$ From $\neg\neg P$ derive P

with formats

$$\begin{array}{|c|}\hline P \\ \vdots \\ \bot \\ \hline \end{array}$$
$$\neg P \qquad (\neg\mathcal{I})$$

$$\frac{P \quad \neg P}{\bot \quad (\neg\mathcal{E})} \qquad \frac{\neg\neg P}{P \qquad (\neg\neg)}$$

The $\neg\mathcal{I}$ rule is very commonly used and is another example of an automatic rule:

> to show $\neg P$ show that the assumption of P leads to a contradiction.

The $\neg\mathcal{E}$ rule can be used in a straightforward way in a forward direction, in which case it simply 'recognizes' that a contradiction is present amongst the