1
Propositional logic

The aim of logic in computer science is to develop languages to model the situations we encounter as computer science professionals, in such a way that we can reason about them formally. Reasoning about situations means constructing arguments about them; we want to do this formally, so that the arguments are valid and can be defended rigorously, or executed on a machine.

Consider the following argument:

Example 1.1 If the train arrives late and there are no taxis at the station, then John is late for his meeting. John is not late for his meeting. The train did arrive late. *Therefore*, there were taxis at the station.

Intuitively, the argument is valid, since if we put the *first* sentence and the *third* sentence together, they tell us that if there are no taxis, then John will be late. The second sentence tells us that he was not late, so it must be the case that there were taxis.

Much of this book will be concerned with arguments that have this structure, namely, that consist of a number of sentences followed by the word 'therefore' and then another sentence. The argument is valid if the sentence after the 'therefore' logically follows from the sentences before it. Exactly what we mean by 'follows from' is the subject of this chapter and the next one.

Consider another example:

Example 1.2 If it is raining and Jane does not have her umbrella with her, then she will get wet. Jane is not wet. It is raining. *Therefore*, Jane has her umbrella with her.

This is also a valid argument. Closer examination reveals that it actually has the same structure as the argument of the previous example! All we have done is substituted some sentence fragments for others:

Example 1.1	Example 1.2
the train is late	it is raining
there are taxis at the station	Jane has her umbrella with her
John is late for his meeting	Jane gets wet.

The argument in each example could be stated without talking about trains and rain, as follows:

If p and not q, then r. Not r. p. *Therefore*, q.

In developing logics, we are not concerned with what the sentences really mean, but only in their logical structure. Of course, when we *apply* such reasoning, as done above, such meaning will be of great interest.

1.1 Declarative sentences

In order to make arguments rigorous, we need to develop a language in which we can express sentences in such a way that brings out their logical structure. The language we begin with is the language of propositional logic. It is based on *propositions*, or *declarative sentences* which one can, in principle, argue as being true or false. Examples of declarative sentences are:

(1) The sum of the numbers 3 and 5 equals 8.
(2) Jane reacted violently to Jack's accusations.
(3) Every even natural number >2 is the sum of two prime numbers.
(4) All Martians like pepperoni on their pizza.
(5) Albert Camus était un écrivain français.
(6) Die Würde des Menschen ist unantastbar.

These sentences are all declarative, because they are in principle capable of being declared 'true', or 'false'. Sentence (1) can be tested by appealing to basic facts about arithmetic (and by tacitly assuming an Arabic, decimal representation of natural numbers). Sentence (2) is a bit more problematic. In order to give it a truth value, we need to know who Jane and Jack are and perhaps to have a reliable account from someone who witnessed the situation described. In principle, e.g., if we had been at the scene, we feel that we would have been able to detect Jane's *violent* reaction, provided that it indeed occurred in that way. Sentence (3), known as Goldbach's conjecture, seems straightforward on the face of it. Clearly, a fact about *all* even numbers >2 is either true or false. But to this day nobody knows whether sentence (3) expresses a truth or not. It is even not clear whether this could be shown by some finite means, even if it were true. However, in this text we will be content with sentences as soon as they can, in principle, attain some truth value regardless of whether this truth value reflects the actual state of affairs suggested by the sentence in question. Sentence (4) seems a bit silly, although we could say that *if* Martians exist and eat pizza, then all of them will either like pepperoni on it or not. (We have to introduce predicate logic in Chapter 2 to see that this sentence is also declarative if *no* Martians exist; it is then true.) Again, for the purposes of this text sentence (4) will do. Et alors, qu'est-ce qu'on pense des phrases (5) et (6)? Sentences (5) and (6) are fine if you happen to read French and German a bit. Thus, declarative statements can be made in any natural, or artificial, language.

The kind of sentences we *won't* consider here are non-declarative ones, like

- Could you please pass me the salt?
- Ready, steady, go!
- May fortune come your way.

Primarily, we are interested in precise declarative sentences, or *statements* about the behaviour of computer systems, or programs. Not only do we want to specify such statements but we also want to *check* whether a given program, or system, fulfils a specification at hand. Thus, we need to develop a calculus of reasoning which allows us to draw conclusions from given assumptions, like initialised variables, which are reliable in the sense that they preserve truth: if all our assumptions are true, then our conclusion ought to be true as well. A much more difficult question is whether, given any true property of a computer program, we can find an argument in our calculus that has this property as its conclusion. The declarative sentence (3) above might illuminate the problematic aspect of such questions in the context of number theory.

The logics we intend to design are *symbolic* in nature. We translate a certain sufficiently large subset of all English declarative sentences into strings of symbols. This gives us a compressed but still complete encoding of declarative sentences and allows us to concentrate on the mere mechanics of our argumentation. This is important since specifications of systems or software are sequences of such declarative sentences. It further opens up the possibility of automatic manipulation of such specifications, a job that computers just love to do[1]. Our strategy is to consider certain declarative sentences as

[1] There is a certain, slightly bitter, circularity in such endeavours: in proving that a certain computer program `P` satisfies a given property, we might let some other computer program `Q` try to find a proof that `P` satisfies the property; but who guarantees us that `Q` satisfies the property of producing only correct proofs? We seem to run into an infinite regress.

being *atomic*, or *indecomposable*, like the sentence

'The number 5 is even.'

We assign certain distinct symbols $p, q, r, \ldots$, or sometimes $p_1, p_2, p_3, \ldots$ to each of these atomic sentences and we can then code up more complex sentences in a *compositional* way. For example, given the atomic sentences

p: 'I won the lottery last week.'
q: 'I purchased a lottery ticket.'
r: 'I won last week's sweepstakes.'

we can form more complex sentences according to the rules below:

- $\neg$: The *negation* of p is denoted by $\neg p$ and expresses 'I did **not** win the lottery last week,' or equivalently 'It is **not** true that I won the lottery last week.'
- $\vee$: Given p and r we may wish to state that *at least one of them* is true: 'I won the lottery last week, **or** I won last week's sweepstakes;' we denote this declarative sentence by $p \vee r$ and call it the *disjunction* of p and r[2].
- $\wedge$: Dually, the formula $p \wedge r$ denotes the rather fortunate *conjunction* of p and r: 'Last week I won the lottery **and** the sweepstakes.'
- $\rightarrow$: Last, but definitely not least, the sentence '**If** I won the lottery last week, **then** I purchased a lottery ticket.' expresses an *implication* between p and q, suggesting that q is a logical consequence of p. We write $p \rightarrow q$ for that[3]. We call p the *assumption* of $p \rightarrow q$ and q its *conclusion*.

Of course, we are entitled to use these rules of constructing propositions repeatedly. For example, we are now in a position to form the proposition

$$p \wedge q \rightarrow \neg r \vee q$$

which means that '**if** p **and** q **then not** r **or** q'. You might have noticed a potential ambiguity in this reading. One could have argued that this sentence has the structure 'p is the case **and if** q **then** . . . ' A computer would require the insertion of brackets, as in

$$(p \wedge q) \rightarrow ((\neg r) \vee q)$$

[2] Its meaning should not be confused with the often implicit meaning of **or** in natural language discourse as **either** . . . **or**. In this text **or** always means *at least one of them* and should not be confounded with *exclusive or* which states that *exactly one* of the two statements holds.

[3] The natural language meaning of '**if** . . . **then** . . . ' often implicitly assumes a *causal role* of the assumption somehow enabling its conclusion. The logical meaning of implication is a bit different, though, in the sense that it states the *preservation of truth* which might happen without any causal relationship. For example, 'If all birds can fly, then Bob Dole was never president of the United States of America.' is a true statement, but there is no known causal connection between the flying skills of penguins and effective campaigning.

to disambiguate this assertion. However, we humans get annoyed by a proliferation of such brackets which is why we adopt certain conventions about the *binding priorities* of these symbols.

Convention 1.3 $\neg$ binds more tightly than $\vee$ and $\wedge$, and the latter two bind more tightly than $\to$. Implication $\to$ is *right-associative*: expressions of the form $p \to q \to r$ denote $p \to (q \to r)$.

1.2 Natural deduction

How do we go about constructing a calculus for reasoning about propositions, so that we can establish the validity of Examples 1.1 and 1.2? Clearly, we would like to have a set of rules each of which allows us to draw a conclusion given a certain arrangement of premises.

In natural deduction, we have such a collection of *proof rules*. They allow us to *infer* formulas from other formulas. By applying these rules in succession, we may infer a conclusion from a set of premises.

Let's see how this works. Suppose we have a set of formulas[4] ϕ_1, ϕ_2, ϕ_3, ..., ϕ_n, which we will call *premises*, and another formula, ψ, which we will call a *conclusion*. By applying proof rules to the premises, we hope to get some more formulas, and by applying more proof rules to those, to eventually obtain the conclusion. This intention we denote by

$$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi.$$

This expression is called a *sequent*; it is *valid* if a proof for it can be found. The sequent for Examples 1.1 and 1.2 is $p \wedge \neg q \to r, \neg r, p \vdash q$. Constructing such a proof is a creative exercise, a bit like programming. It is not necessarily obvious which rules to apply, and in what order, to obtain the desired conclusion. Additionally, our proof rules should be carefully chosen; otherwise, we might be able to 'prove' invalid patterns of argumentation. For

[4] It is traditional in logic to use Greek letters. Lower-case letters are used to stand for formulas and upper-case letters are used for sets of formulas. Here are some of the more commonly used Greek letters, together with their pronunciation:

Lower-case		Upper-case	
ϕ	phi	Φ	Phi
ψ	psi	Ψ	Psi
χ	chi	Γ	Gamma
η	eta	Δ	Delta
α	alpha		
β	beta		
γ	gamma		

example, we expect that we won't be able to show the sequent $p, q \vdash p \wedge \neg q$. For example, if p stands for 'Gold is a metal.' and q for 'Silver is a metal,' then knowing these two facts should not allow us to infer that 'Gold is a metal whereas silver isn't.'

Let's now look at our proof rules. We present about fifteen of them in total; we will go through them in turn and then summarise at the end of this section.

1.2.1 Rules for natural deduction

The rules for conjunction Our first rule is called the rule for conjunction ($\wedge$): and-introduction. It allows us to conclude $\phi \wedge \psi$, given that we have already concluded ϕ and ψ separately. We write this rule as

$$\frac{\phi \qquad \psi}{\phi \wedge \psi} \wedge\mathrm{i}.$$

Above the line are the two premises of the rule. Below the line goes the conclusion. (It might not yet be the final conclusion of our argument; we might have to apply more rules to get there.) To the right of the line, we write the name of the rule; $\wedge$i is read 'and-introduction'. Notice that we have introduced a $\wedge$ (in the conclusion) where there was none before (in the premises).

For each of the connectives, there is one or more rules to introduce it and one or more rules to eliminate it. The rules for and-elimination are these two:

$$\frac{\phi \wedge \psi}{\phi} \wedge\mathrm{e}_1 \qquad \frac{\phi \wedge \psi}{\psi} \wedge\mathrm{e}_2. \tag{1.1}$$

The rule $\wedge\mathrm{e}_1$ says: if you have a proof of $\phi \wedge \psi$, then by applying this rule you can get a proof of ϕ. The rule $\wedge\mathrm{e}_2$ says the same thing, but allows you to conclude ψ instead. Observe the dependences of these rules: in the first rule of (1.1), the conclusion ϕ has to match the first conjunct of the premise, whereas the exact nature of the second conjunct ψ is irrelevant. In the second rule it is just the other way around: the conclusion ψ has to match the second conjunct ψ and ϕ can be any formula. It is important to engage in this kind of *pattern matching* before the application of proof rules.

Example 1.4 Let's use these rules to prove that $p \wedge q,\ r \vdash q \wedge r$ is valid. We start by writing down the premises; then we leave a gap and write the conclusion:

$$\begin{array}{l} p \wedge q \\ r \\ \\ q \wedge r \end{array}$$

The task of constructing the proof is to fill the gap between the premises and the conclusion by applying a suitable sequence of proof rules. In this case, we apply $\wedge e_2$ to the first premise, giving us q. Then we apply $\wedge i$ to this q and to the second premise, r, giving us $q \wedge r$. That's it! We also usually number all the lines, and write in the justification for each line, producing this:

1	$p \wedge q$	premise
2	r	premise
3	q	$\wedge e_2$ 1
4	$q \wedge r$	$\wedge i$ 3, 2

Demonstrate to yourself that you've understood this by trying to show on your own that $(p \wedge q) \wedge r,\ s \wedge t \vdash q \wedge s$ is valid. Notice that the ϕ and ψ can be instantiated not just to atomic sentences, like p and q in the example we just gave, but also to compound sentences. Thus, from $(p \wedge q) \wedge r$ we can deduce $p \wedge q$ by applying $\wedge e_1$, instantiating ϕ to $p \wedge q$ and ψ to r.

If we applied these proof rules literally, then the proof above would actually be a tree with root $q \wedge r$ and leaves $p \wedge q$ and r, like this:

$$\dfrac{\dfrac{p \wedge q}{q}\,\wedge e_2 \qquad r}{q \wedge r}\,\wedge i$$

However, we flattened this tree into a linear presentation which necessitates the use of pointers as seen in lines 3 and 4 above. These pointers allow us to recreate the actual proof tree. Throughout this text, we will use the flattened version of presenting proofs. That way you have to concentrate only on finding a proof, not on how to fit a growing tree onto a sheet of paper.

If a sequent is valid, there may be many different ways of proving it. So if you compare your solution to these exercises with those of others, they need not coincide. The important thing to realise, though, is that any putative proof can be *checked* for correctness by checking each individual line, starting at the top, for the valid application of its proof rule.

The rules of double negation Intuitively, there is no difference between a formula ϕ and its *double negation* $\neg\neg\phi$, which expresses no more and nothing less than ϕ itself. The sentence

'It is **not** true that it does **not** rain.'

is just a more contrived way of saying

'It rains.'

Conversely, knowing 'It rains,' we are free to state this fact in this more complicated manner if we wish. Thus, we obtain rules of elimination and introduction for double negation:

$$\frac{\neg\neg\phi}{\phi}\ \neg\neg\mathrm{e} \qquad \frac{\phi}{\neg\neg\phi}\ \neg\neg\mathrm{i}.$$

(There are rules for single negation on its own, too, which we will see later.)

Example 1.5 The proof of the sequent $p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$ below uses most of the proof rules discussed so far:

1	p	premise
2	$\neg\neg(q \wedge r)$	premise
3	$\neg\neg p$	$\neg\neg$i 1
4	$q \wedge r$	$\neg\neg$e 2
5	r	$\wedge\mathrm{e}_2$ 4
6	$\neg\neg p \wedge r$	$\wedge$i 3, 5

Example 1.6 We now prove the sequent $(p \wedge q) \wedge r,\ s \wedge t \vdash q \wedge s$ which you were invited to prove by yourself in the last section. Please compare the proof below with your solution:

1	$(p \wedge q) \wedge r$	premise
2	$s \wedge t$	premise
3	$p \wedge q$	$\wedge\mathrm{e}_1$ 1
4	q	$\wedge\mathrm{e}_2$ 3
5	s	$\wedge\mathrm{e}_1$ 2
6	$q \wedge s$	$\wedge$i 4, 5

The rule for eliminating implication There is one rule to introduce $\rightarrow$ and one to eliminate it. The latter is one of the best known rules of propositional logic and is often referred to by its Latin name *modus ponens*. We will usually call it by its modern name, implies-elimination (sometimes also referred to as arrow-elimination). This rule states that, given ϕ and knowing that ϕ implies ψ, we may rightfully conclude ψ. In our calculus, we write this as

$$\frac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow\text{e}.$$

Let us justify this rule by spelling out instances of some declarative sentences p and q. Suppose that

p : It rained.
$p \rightarrow q$: If it rained, then the street is wet.

so q is just 'The street is wet.' Now, *if* we know that it rained and *if* we know that the street is wet in the case that it rained, then we may combine these two pieces of information to conclude that the street is indeed wet. Thus, the justification of the $\rightarrow$e rule is a mere application of common sense. Another example from programming is:

p : The value of the program's input is an integer.
$p \rightarrow q$: If the program's input is an integer, then the program outputs a boolean.

Again, we may put all this together to conclude that our program outputs a boolean value if supplied with an integer input. However, it is important to realise that the presence of p is absolutely essential for the inference to happen. For example, our program might well satisfy $p \rightarrow q$, but if it doesn't satisfy p – e.g. if its input is a surname – then we will not be able to derive q.

As we saw before, the formal parameters ϕ and the ψ for $\rightarrow$e can be instantiated to any sentence, including compound ones:

1	$\neg p \wedge q$	premise
2	$\neg p \wedge q \rightarrow r \vee \neg p$	premise
3	$r \vee \neg p$	$\rightarrow$e 2, 1

Of course, we may use any of these rules as often as we wish. For example, given p, $p \to q$ and $p \to (q \to r)$, we may infer r:

1	$p \to (q \to r)$	premise
2	$p \to q$	premise
3	p	premise
4	$q \to r$	$\to$e 1, 3
5	q	$\to$e 2, 3
6	r	$\to$e 4, 5

Before turning to implies-introduction, let's look at a hybrid rule which has the Latin name *modus tollens*. It is like the $\to$e rule in that it eliminates an implication. Suppose that $p \to q$ *and* $\neg q$ are the case. Then, *if* p holds we can use $\to$e to conclude that q holds. Thus, we then have that q *and* $\neg q$ hold, which is impossible. Therefore, we may infer that p must be false. But this can only mean that $\neg p$ is true. We summarise this reasoning into the rule *modus tollens*, or MT for short:[5]

$$\frac{\phi \to \psi \quad \neg\psi}{\neg\phi}\ \mathrm{MT}.$$

Again, let us see an example of this rule in the natural language setting:

'If Abraham Lincoln was Ethiopian, then he was African. Abraham Lincoln was not African; therefore he was not Ethiopian.'

Example 1.7 In the following proof of

$$p \to (q \to r),\ p,\ \neg r \vdash \neg q$$

we use several of the rules introduced so far:

1	$p \to (q \to r)$	premise
2	p	premise
3	$\neg r$	premise
4	$q \to r$	$\to$e 1, 2
5	$\neg q$	MT 4, 3

[5] We will be able to *derive* this rule from other ones later on, but we introduce it here because it allows us already to do some pretty slick proofs. You may think of this rule as one on a higher level insofar as it does not mention the lower-level rules upon which it depends.

Examples 1.8 Here are two example proofs which combine the rule MT with either ¬¬e or ¬¬i:

1	$\neg p \to q$	premise
2	$\neg q$	premise
3	$\neg\neg p$	MT 1, 2
4	p	¬¬e 3

proves that the sequent $\neg p \to q, \neg q \vdash p$ is valid; and

1	$p \to \neg q$	premise
2	q	premise
3	$\neg\neg q$	¬¬i 2
4	$\neg p$	MT 1, 3

shows the validity of the sequent $p \to \neg q, q \vdash \neg p$.

Note that the order of applying double negation rules and MT is different in these examples; this order is driven by the structure of the particular sequent whose validity one is trying to show.

The rule implies introduction The rule MT made it possible for us to show that $p \to q, \neg q \vdash \neg p$ is valid. But the validity of the sequent $p \to q \vdash \neg q \to \neg p$ seems just as plausible. That sequent is, in a certain sense, saying the same thing. Yet, so far we have no rule which *builds* implications that do not already occur as premises in our proofs. The mechanics of such a rule are more involved than what we have seen so far. So let us proceed with care. Let us suppose that $p \to q$ is the case. If we *temporarily* assume that $\neg q$ holds, we can use MT to infer $\neg p$. Thus, assuming $p \to q$ we can show that $\neg q$ **implies** $\neg p$; but the latter we express *symbolically* as $\neg q \to \neg p$. To summarise, we have found an argumentation for $p \to q \vdash \neg q \to \neg p$:

1	$p \to q$	premise
2	$\neg q$	assumption
3	$\neg p$	MT 1, 2
4	$\neg q \to \neg p$	→i 2–3

The box in this proof serves to demarcate the scope of the temporary assumption $\neg q$. What we are saying is: let's make the assumption of $\neg q$. To do this, we open a box and put $\neg q$ at the top. Then we continue applying other rules as normal, for example to obtain $\neg p$. But this still depends on the assumption of $\neg q$, so it goes inside the box. Finally, we are ready to apply $\rightarrow$i. It allows us to conclude $\neg q \rightarrow \neg p$, but that conclusion no longer *depends* on the assumption $\neg q$. Compare this with saying that 'If you are French, then you are European.' The truth of this sentence does not depend on whether anybody is French or not. Therefore, we write the conclusion $\neg q \rightarrow \neg p$ outside the box.

This works also as one would expect if we think of $p \rightarrow q$ as a *type* of a procedure. For example, p could say that the procedure expects an integer value x as input and q might say that the procedure returns a boolean value y as output. The validity of $p \rightarrow q$ amounts now to an assume-guarantee assertion: if the input is an integer, then the output is a boolean. This assertion can be true about a procedure while that same procedure could compute strange things or crash in the case that the input is not an integer. Showing $p \rightarrow q$ using the rule $\rightarrow$i is now called *type checking*, an important topic in the construction of compilers for typed programming languages.

We thus formulate the rule $\rightarrow$i as follows:

$$\frac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \psi\end{array}}}{\phi \rightarrow \psi}\ \rightarrow\text{i}.$$

It says: in order to prove $\phi \rightarrow \psi$, make a temporary assumption of ϕ and then prove ψ. In your proof of ψ, you can use ϕ and any of the other formulas such as premises and provisional conclusions that you have made so far. Proofs may nest boxes or open new boxes after old ones have been closed. There are rules about which formulas can be used at which points in the proof. Generally, we can only use a formula ϕ in a proof at a given point if that formula occurs *prior* to that point and if no box which encloses that occurrence of ϕ has been closed already.

The line immediately following a closed box has to match the pattern of the conclusion of the rule that uses the box. For implies-introduction, this means that we have to continue after the box with $\phi \rightarrow \psi$, where ϕ was the first and ψ the last formula of that box. We will encounter two more proof rules involving proof boxes and they will require similar pattern matching.

Example 1.9 Here is another example of a proof using $\rightarrow$i:

1	$\neg q \rightarrow \neg p$	premise
2	p	assumption
3	$\neg\neg p$	$\neg\neg$i 2
4	$\neg\neg q$	MT 1, 3
5	$p \rightarrow \neg\neg q$	$\rightarrow$i 2–4

which verifies the validity of the sequent $\neg q \rightarrow \neg p \vdash p \rightarrow \neg\neg q$. Notice that we could apply the rule MT to formulas occurring in or above the box: at line 4, no box has been closed that would enclose line 1 or 3.

At this point it is instructive to consider the one-line argument

1	p	premise

which demonstrates $p \vdash p$. The rule $\rightarrow$i (with conclusion $\phi \rightarrow \psi$) does not prohibit the possibility that ϕ and ψ coincide. They could both be instantiated to p. Therefore we may extend the proof above to

1	p	assumption
2	$p \rightarrow p$	$\rightarrow$i $1-1$

We write $\vdash p \rightarrow p$ to express that the argumentation for $p \rightarrow p$ does not depend on any premises at all.

Definition 1.10 Logical formulas ϕ with valid sequent $\vdash \phi$ are *theorems*.

Example 1.11 Here is an example of a theorem whose proof utilises most of the rules introduced so far:

1	$q \rightarrow r$	assumption
2	$\neg q \rightarrow \neg p$	assumption
3	p	assumption
4	$\neg\neg p$	$\neg\neg$i 3
5	$\neg\neg q$	MT 2, 4
6	q	$\neg\neg$e 5
7	r	$\rightarrow$e 1, 6
8	$p \rightarrow r$	$\rightarrow$i 3–7
9	$(\neg q \rightarrow \neg p) \rightarrow (p \rightarrow r)$	$\rightarrow$i 2–8
10	$(q \rightarrow r) \rightarrow ((\neg q \rightarrow \neg p) \rightarrow (p \rightarrow r))$	$\rightarrow$i 1–9

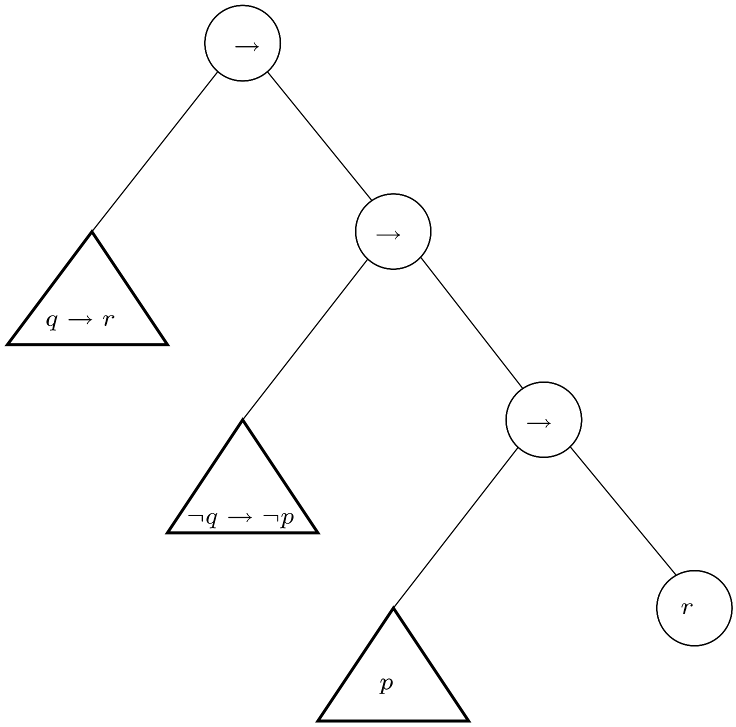

Figure 1.1. Part of the structure of the formula $(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$ to show how it determines the proof structure.

Therefore the sequent $\vdash (q \to r) \to ((\neg q \to \neg p) \to (p \to r))$ is valid, showing that $(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$ is another theorem.

Remark 1.12 Indeed, this example indicates that we may transform any proof of $\phi_1, \phi_2, \dots, \phi_n \vdash \psi$ in such a way into a proof of the theorem

$$\vdash \phi_1 \to (\phi_2 \to (\phi_3 \to (\cdots \to (\phi_n \to \psi) \dots)))$$

by 'augmenting' the previous proof with n lines of the rule $\to$i applied to ϕ_n, ϕ_{n-1},..., ϕ_1 in that order.

The nested boxes in the proof of Example 1.11 reveal a pattern of using elimination rules first, to deconstruct assumptions we have made, and then introduction rules to construct our final conclusion. More difficult proofs may involve several such phases.

Let us dwell on this important topic for a while. How did we come up with the proof above? Parts of it are *determined* by the structure of the formulas we have, while other parts require us to be *creative*. Consider the logical structure of $(q \to r) \to ((\neg q \to \neg p) \to (p \to r))$ schematically depicted in Figure 1.1. The formula is overall an implication since $\to$ is the root of the tree in Figure 1.1. But the only way to build an implication is by means of the rule $\rightarrow$i. Thus, we need to state the assumption of that implication as such (line 1) and have to show its conclusion (line 9). If we managed to do that, then we know how to end the proof in line 10. In fact, as we already remarked, this is the only way we could have ended it. So essentially lines 1, 9 and 10 are completely determined by the structure of the formula; further, we have reduced the problem to filling the gaps in between lines 1 and 9. But again, the formula in line 9 is an implication, so we have only one way of showing it: assuming its premise in line 2 and trying to show its conclusion in line 8; as before, line 9 is obtained by $\rightarrow$i. The formula $p \rightarrow r$ in line 8 is yet another implication. Therefore, we have to assume p in line 3 and hope to show r in line 7, then $\rightarrow$i produces the desired result in line 8.

The remaining question now is this: how can we show r, using the three assumptions in lines 1–3? This, and only this, is the creative part of this proof. We see the implication $q \rightarrow r$ in line 1 and know how to get r (using $\rightarrow$e) if only we had q. So how could we get q? Well, lines 2 and 3 almost look like a pattern for the MT rule, which would give us $\neg\neg q$ in line 5; the latter is quickly changed to q in line 6 via $\neg\neg$e. However, the pattern for MT does not match right away, since it requires $\neg\neg p$ instead of p. But this is easily accomplished via $\neg\neg$i in line 4.

The moral of this discussion is that the logical structure of the formula to be shown tells you a lot about the structure of a possible proof and it is definitely worth your while to exploit that information in trying to prove sequents. Before ending this section on the rules for implication, let's look at some more examples (this time also involving the rules for conjunction).

Example 1.13 Using the rule $\wedge$i, we can prove the validity of the sequent

$$p \wedge q \rightarrow r \vdash p \rightarrow (q \rightarrow r):$$

1	$p \wedge q \rightarrow r$	premise
2	p	assumption
3	q	assumption
4	$p \wedge q$	$\wedge$i 2, 3
5	r	$\rightarrow$e 1, 4
6	$q \rightarrow r$	$\rightarrow$i 3–5
7	$p \rightarrow (q \rightarrow r)$	$\rightarrow$i 2–6

Example 1.14 Using the two elimination rules $\wedge e_1$ and $\wedge e_2$, we can show that the 'converse' of the sequent above is valid, too:

1	$p \to (q \to r)$	premise
2	$p \wedge q$	assumption
3	p	$\wedge e_1$ 2
4	q	$\wedge e_2$ 2
5	$q \to r$	$\to$e 1, 3
6	r	$\to$e 5, 4
7	$p \wedge q \to r$	$\to$i 2–6

The validity of $p \to (q \to r) \vdash p \wedge q \to r$ and $p \wedge q \to r \vdash p \to (q \to r)$ means that these two formulas are equivalent in the sense that we can prove one from the other. We denote this by

$$p \wedge q \to r \dashv\vdash p \to (q \to r).$$

Since there can be only one formula to the right of $\vdash$, we observe that each instance of $\dashv\vdash$ can only relate *two* formulas to each other.

Example 1.15 Here is an example of a proof that uses introduction *and* elimination rules for conjunction; it shows the validity of the sequent $p \to q \vdash p \wedge r \to q \wedge r$:

1	$p \to q$	premise
2	$p \wedge r$	assumption
3	p	$\wedge e_1$ 2
4	r	$\wedge e_2$ 2
5	q	$\to$e 1, 3
6	$q \wedge r$	$\wedge$i 5, 4
7	$p \wedge r \to q \wedge r$	$\to$i 2–6

The rules for disjunction The rules for disjunction are different in spirit from those for conjunction. The case for conjunction was concise and clear: proofs of $\phi \wedge \psi$ are essentially nothing but a concatenation of a proof of ϕ and a proof of ψ, plus an additional line invoking $\wedge$i. In the case of disjunctions, however, it turns out that the *introduction* of disjunctions is by far easier to grasp than their elimination. So we begin with the rules $\vee i_1$ and $\vee i_2$. From the premise ϕ we can infer that 'ϕ **or** ψ' holds, for we already know that ϕ holds. Note that this inference is valid for any choice of ψ. By the same token, we may conclude 'ϕ **or** ψ' if we already have ψ. Similarly, that inference works for any choice of ϕ. Thus, we arrive at the proof rules

$$\frac{\phi}{\phi \vee \psi}\ \vee\mathrm{i}_1 \qquad \frac{\psi}{\phi \vee \psi}\ \vee\mathrm{i}_2.$$

So if p stands for 'Agassi won a gold medal in 1996.' and q denotes the sentence 'Agassi won Wimbledon in 1996.' then $p \vee q$ is the case because p is true, regardless of the fact that q is false. Naturally, the constructed disjunction depends upon the assumptions needed in establishing its respective disjunct p or q.

Now let's consider or-elimination. How can we use a formula of the form $\phi \vee \psi$ in a proof? Again, our guiding principle is to *disassemble* assumptions into their basic constituents so that the latter may be used in our argumentation such that they render our desired conclusion. Let us imagine that we want to show some proposition χ by assuming $\phi \vee \psi$. Since we don't know which of ϕ and ψ is true, we have to give *two* separate proofs which we need to combine into one argument:

1. First, we assume ϕ is true and have to come up with a proof of χ.
2. Next, we assume ψ is true and need to give a proof of χ as well.
3. Given these two proofs, we can infer χ from the truth of $\phi \vee \psi$, since our case analysis above is exhaustive.

Therefore, we write the rule $\vee$e as follows:

$$\frac{\phi \vee \psi \qquad \boxed{\begin{array}{c}\phi\\ \vdots\\ \chi\end{array}} \qquad \boxed{\begin{array}{c}\psi\\ \vdots\\ \chi\end{array}}}{\chi}\ \vee\mathrm{e}.$$

It is saying that: if $\phi \vee \psi$ is true and – no matter whether we assume ϕ or we assume ψ – we can get a proof of χ, then we are entitled to deduce χ anyway. Let's look at a proof that $p \vee q \vdash q \vee p$ is valid:

1	$p \vee q$	premise
2	p	assumption
3	$q \vee p$	$\vee\mathrm{i}_2$ 2
4	q	assumption
5	$q \vee p$	$\vee\mathrm{i}_1$ 4
6	$q \vee p$	$\vee$e 1, 2–3, 4–5

Here are some points you need to remember about applying the $\vee$e rule.

- For it to be a sound argument we have to make sure that the conclusions in each of the two cases (the χ in the rule) are actually the same formula.
- The work done by the rule $\vee$e is the combining of the arguments of the two cases into one.
- In each case you may not use the temporary assumption of the other case, unless it is something that has already been shown before those case boxes began.
- The invocation of rule $\vee$e in line 6 lists three things: the line in which the disjunction appears (1), and the location of the two boxes for the two cases (2–3 and 4–5).

If we use $\phi \vee \psi$ in an argument where it occurs only as an assumption or a premise, then we are missing a certain amount of information: we know ϕ, or ψ, but we don't know which one of the two it is. Thus, we have to make a solid case for each of the two possibilities ϕ or ψ; this resembles the behaviour of a `CASE` or `IF` statement found in most programming languages.

Example 1.16 Here is a more complex example illustrating these points. We prove that the sequent $q \to r \vdash p \vee q \to p \vee r$ is valid:

1	$q \to r$	premise
2	$p \vee q$	assumption
3	p	assumption
4	$p \vee r$	$\vee\mathrm{i}_1$ 3
5	q	assumption
6	r	$\to$e $1, 5$
7	$p \vee r$	$\vee\mathrm{i}_2$ 6
8	$p \vee r$	$\vee$e $2, 3{-}4, 5{-}7$
9	$p \vee q \to p \vee r$	$\to$i $2{-}8$

Note that the propositions in lines 4, 7 and 8 coincide, so the application of $\vee$e is legitimate.

We give some more example proofs which use the rules $\vee$e, $\vee\mathrm{i}_1$ and $\vee\mathrm{i}_2$.

Example 1.17 Proving the validity of the sequent $(p \vee q) \vee r \vdash p \vee (q \vee r)$ is surprisingly long and seemingly complex. But this is to be expected, since the elimination rules break $(p \vee q) \vee r$ up into its atomic constituents p, q and r, whereas the introduction rules then built up the formula $p \vee (q \vee r)$.

1	$(p \vee q) \vee r$	premise
2	$(p \vee q)$	assumption
3	p	assumption
4	$p \vee (q \vee r)$	$\vee\mathrm{i}_1$ 3
5	q	assumption
6	$q \vee r$	$\vee\mathrm{i}_1$ 5
7	$p \vee (q \vee r)$	$\vee\mathrm{i}_2$ 6
8	$p \vee (q \vee r)$	$\vee\mathrm{e}$ $2, 3{-}4, 5{-}7$
9	r	assumption
10	$q \vee r$	$\vee\mathrm{i}_2$ 9
11	$p \vee (q \vee r)$	$\vee\mathrm{i}_2$ 10
12	$p \vee (q \vee r)$	$\vee\mathrm{e}$ $1, 2{-}8, 9{-}11$

Example 1.18 From boolean algebra, or circuit theory, you may know that disjunctions distribute over conjunctions. We are now able to prove this in natural deduction. The following proof:

1	$p \wedge (q \vee r)$	premise
2	p	$\wedge\mathrm{e}_1$ 1
3	$q \vee r$	$\wedge\mathrm{e}_2$ 1
4	q	assumption
5	$p \wedge q$	$\wedge\mathrm{i}$ $2, 4$
6	$(p \wedge q) \vee (p \wedge r)$	$\vee\mathrm{i}_1$ 5
7	r	assumption
8	$p \wedge r$	$\wedge\mathrm{i}$ $2, 7$
9	$(p \wedge q) \vee (p \wedge r)$	$\vee\mathrm{i}_2$ 8
10	$(p \wedge q) \vee (p \wedge r)$	$\vee\mathrm{e}$ $3, 4{-}6, 7{-}9$

verifies the validity of the sequent $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$ and you are encouraged to show the validity of the 'converse' $(p \wedge q) \vee (p \wedge r) \vdash p \wedge (q \vee r)$ yourself.

A final rule is required in order to allow us to conclude a box with a formula which has already appeared earlier in the proof. Consider the sequent $\vdash p \to (q \to p)$, whose validity may be proved as follows:

1	p	assumption
2	q	assumption
3	p	copy 1
4	$q \to p$	$\to$i 2–3
5	$p \to (q \to p)$	$\to$i 1–4

The rule 'copy' allows us to repeat something that we know already. We need to do this in this example, because the rule $\to$i requires that we end the inner box with p. The copy rule entitles us to copy formulas that appeared before, unless they depend on temporary assumptions whose box has already been closed. Though a little inelegant, this additional rule is a small price to pay for the freedom of being able to use premises, or any other 'visible' formulas, more than once.

The rules for negation We have seen the rules $\neg\neg$i and $\neg\neg$e, but we haven't seen any rules that introduce or eliminate single negations. These rules involve the notion of *contradiction*. This detour is to be expected since our reasoning is concerned about the inference, and therefore the preservation, of truth. Hence, there cannot be a direct way of inferring $\neg\phi$, given ϕ.

Definition 1.19 Contradictions are expressions of the form $\phi \wedge \neg\phi$ or $\neg\phi \wedge \phi$, where ϕ is any formula.

Examples of such contradictions are $r \wedge \neg r$, $(p \to q) \wedge \neg(p \to q)$ and $\neg(r \vee s \to q) \wedge (r \vee s \to q)$. Contradictions are a very important notion in logic. As far as truth is concerned, they are all equivalent; that means we should be able to prove the validity of

$$\neg(r \vee s \to q) \wedge (r \vee s \to q) \dashv\vdash (p \to q) \wedge \neg(p \to q) \tag{1.2}$$

since both sides are contradictions. We'll be able to prove this later, when we have introduced the rules for negation.

Indeed, it's not just that contradictions can be derived from contradictions; actually, *any* formula can be derived from a contradiction. This can be confusing when you first encounter it; why should we endorse the argument $p \wedge \neg p \vdash q$, where

$$p: \text{The moon is made of green cheese.}$$
$$q: \text{I like pepperoni on my pizza.}$$

considering that our taste in pizza doesn't have anything to do with the constitution of the moon? On the face of it, such an endorsement may seem absurd. Nevertheless, natural deduction does have this feature that any formula can be derived from a contradiction and therefore it makes this argument valid. The reason it takes this stance is that $\vdash$ tells us all the things we may infer, provided that we can assume the formulas to the left of it. This process does not care whether such premises make any sense. This has at least the advantage that we can match $\vdash$ to checks based on semantic intuitions which we formalise later by using truth tables: if all the premises compute to 'true', then the conclusion must compute 'true' as well. In particular, this is not a constraint in the case that one of the premises is (always) false.

The fact that $\bot$ can prove anything is encoded in our calculus by the proof rule bottom-elimination:

$$\frac{\bot}{\phi}\ \bot\mathrm{e}.$$

The fact that $\bot$ itself represents a contradiction is encoded by the proof rule not-elimination:

$$\frac{\phi \quad \neg\phi}{\bot}\ \neg\mathrm{e}.$$

Example 1.20 We apply these rules to show that $\neg p \vee q \vdash p \rightarrow q$ is valid:

1	$\neg p \vee q$			
2	$\neg p$	premise	q	premise
3	p	assumption	p	assumption
4	$\bot$	$\neg$e 3, 2	q	copy 2
5	q	$\bot$e 4	$p \rightarrow q$	$\rightarrow$i 3–4
6	$p \rightarrow q$	$\rightarrow$i 3–5		
7	$p \rightarrow q$			$\vee$e 1, 2–6

Notice how, in this example, the proof boxes for $\vee$e are drawn side by side instead of on top of each other. It doesn't matter which way you do it.

What about introducing negations? Well, suppose we make an assumption which gets us into a contradictory state of affairs, i.e. gets us $\bot$. Then our assumption cannot be true; so it must be false. This intuition is the basis for the proof rule $\neg$i:

$$\frac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \bot\end{array}}}{\neg\phi}\ \neg\mathrm{i}.$$

Example 1.21 We put these rules in action, demonstrating that the sequent $p \to q,\ p \to \neg q \vdash \neg p$ is valid:

1	$p \to q$	premise
2	$p \to \neg q$	premise
3	p	assumption
4	q	$\to$e $1, 3$
5	$\neg q$	$\to$e $2, 3$
6	$\bot$	$\neg$e $4, 5$
7	$\neg p$	$\neg$i $3-6$

Lines 3–6 contain all the work of the $\neg$i rule. Here is a second example, showing the validity of a sequent, $p \to \neg p \vdash \neg p$, with a contradictory formula as sole premise:

1	$p \to \neg p$	premise
2	p	assumption
3	$\neg p$	$\to$e $1, 2$
4	$\bot$	$\neg$e $2, 3$
5	$\neg p$	$\neg$i $2-4$

Example 1.22 We prove that the sequent $p \to (q \to r),\ p,\ \neg r \vdash \neg q$ is valid,

without using the MT rule:

1	$p \to (q \to r)$	premise
2	p	premise
3	$\neg r$	premise
4	q	assumption
5	$q \to r$	$\to$e 1, 2
6	r	$\to$e 5, 4
7	$\bot$	$\neg$e 6, 3
8	$\neg q$	$\neg$i 4–7

Example 1.23 Finally, we return to the argument of Examples 1.1 and 1.2, which can be coded up by the sequent $p \land \neg q \to r$, $\neg r$, $p \vdash q$ whose validity we now prove:

1	$p \land \neg q \to r$	premise
2	$\neg r$	premise
3	p	premise
4	$\neg q$	assumption
5	$p \land \neg q$	$\land$i 3, 4
6	r	$\to$e 1, 5
7	$\bot$	$\neg$e 6, 2
8	$\neg\neg q$	$\neg$i 4–7
9	q	$\neg\neg$e 8

1.2.2 Derived rules

When describing the proof rule *modus tollens* (MT), we mentioned that it is not a primitive rule of natural deduction, but can be derived from some of the other rules. Here is the derivation of

$$\frac{\phi \to \psi \quad \neg\psi}{\neg\phi} \ \mathrm{MT}$$

from →e, ¬e and ¬i:

1	$\phi \to \psi$	premise
2	$\neg\psi$	premise
3	ϕ	assumption
4	ψ	→e 1, 3
5	$\bot$	¬e 4, 2
6	$\neg\phi$	¬i 3−5

We could now go back through the proofs in this chapter and replace applications of MT by this combination of →e, ¬e and ¬i. However, it is convenient to think of MT as a shorthand (or a macro).

The same holds for the rule

$$\frac{\phi}{\neg\neg\phi}\ \neg\neg\text{i}.$$

It can be derived from the rules ¬i and ¬e, as follows:

1	ϕ	premise
2	$\neg\phi$	assumption
3	$\bot$	¬e 1, 2
4	$\neg\neg\phi$	¬i 2−3

There are (unboundedly) many such derived rules which we could write down. However, there is no point in making our calculus fat and unwieldy; and some purists would say that we should stick to a minimum set of rules, all of which are independent of each other. We don't take such a purist view. Indeed, the two derived rules we now introduce are extremely useful. You will find that they crop up frequently when doing exercises in natural deduction, so it is worth giving them names as derived rules. In the case of the second one, its derivation from the primitive proof rules is not very obvious.

The first one has the Latin name *reductio ad absurdum*. It means 'reduction to absurdity' and we will simply call it *proof by contradiction* (PBC for short). The rule says: if from $\neg\phi$ we obtain a contradiction, then we are entitled to deduce ϕ:

$$\frac{\boxed{\begin{matrix}\neg\phi\\ \vdots\\ \bot\end{matrix}}}{\phi}\ \text{PBC}.$$

This rule looks rather similar to ¬i, except that the negation is in a different place. This is the clue to how to derive PBC from our basic proof rules. Suppose we have a proof of $\bot$ from $\neg\phi$. By →i, we can transform this into a proof of $\neg\phi \to \bot$ and proceed as follows:

1	$\neg\phi \to \bot$	given
2	$\neg\phi$	assumption
3	$\bot$	→e 1, 2
4	$\neg\neg\phi$	¬i 2–3
5	ϕ	¬¬e 4

This shows that PBC can be derived from →i, ¬i, →e and ¬¬e.

The final derived rule we consider in this section is arguably the most useful to use in proofs, because its derivation is rather long and complicated, so its usage often saves time and effort. It also has a Latin name, *tertium non datur*; the English name is the law of the excluded middle, or LEM for short. It simply says that $\phi \vee \neg\phi$ is true: whatever ϕ is, it must be either true or false; in the latter case, $\neg\phi$ is true. There is no third possibility (hence *excluded middle*): the sequent $\vdash \phi \vee \neg\phi$ is valid. Its validity is implicit, for example, whenever you write an if-statement in a programming language: '`if` B $\{C_1\}$ `else` $\{C_2\}$' relies on the fact that $B \vee \neg B$ is always true (and that B and $\neg B$ can never be true at the same time). Here is a proof in natural deduction that derives the law of the excluded middle from basic proof rules:

1	$\neg(\phi \vee \neg\phi)$	assumption
2	ϕ	assumption
3	$\phi \vee \neg\phi$	$\vee$i$_1$ 2
4	$\bot$	¬e 3, 1
5	$\neg\phi$	¬i 2–4
6	$\phi \vee \neg\phi$	$\vee$i$_2$ 5
7	$\bot$	¬e 6, 1
8	$\neg\neg(\phi \vee \neg\phi)$	¬i 1–7
9	$\phi \vee \neg\phi$	¬¬e 8

Example 1.24 Using LEM, we show that $p \to q \vdash \neg p \vee q$ is valid:

1	$p \to q$	premise
2	$\neg p \vee p$	LEM
3	$\neg p$	assumption
4	$\neg p \vee q$	$\vee\mathrm{i}_1$ 3
5	p	assumption
6	q	$\to$e 1, 5
7	$\neg p \vee q$	$\vee\mathrm{i}_2$ 6
8	$\neg p \vee q$	$\vee$e 2, 3–4, 5–7

It can be difficult to decide which instance of LEM would benefit the progress of a proof. Can you re-do the example above with $q \vee \neg q$ as LEM?

1.2.3 Natural deduction in summary

The proof rules for natural deduction are summarised in Figure 1.2. The explanation of the rules we have given so far in this chapter is *declarative*; we have presented each rule and justified it in terms of our intuition about the logical connectives. However, when you try to use the rules yourself, you'll find yourself looking for a more *procedural* interpretation; what does a rule do and how do you use it? For example,

- $\wedge$i says: to prove $\phi \wedge \psi$, you must first prove ϕ and ψ separately and then use the rule $\wedge$i.
- $\wedge\mathrm{e}_1$ says: to prove ϕ, try proving $\phi \wedge \psi$ and then use the rule $\wedge\mathrm{e}_1$. Actually, this doesn't sound like very good advice because probably proving $\phi \wedge \psi$ will be harder than proving ϕ alone. However, you might find that you *already have* $\phi \wedge \psi$ lying around, so that's when this rule is useful. Compare this with the example sequent in Example 1.15.
- $\vee\mathrm{i}_1$ says: to prove $\phi \vee \psi$, try proving ϕ. Again, in general it is harder to prove ϕ than it is to prove $\phi \vee \psi$, so this will usually be useful only if you've already managed to prove ϕ. For example, if you want to prove $q \vdash p \vee q$, you certainly won't be able simply to use the rule $\vee\mathrm{i}_1$, but $\vee\mathrm{i}_2$ will work.
- $\vee$e has an excellent procedural interpretation. It says: if you have $\phi \vee \psi$, and you want to prove some χ, then try to prove χ from ϕ and from ψ in turn. (In those subproofs, of course you can use the other prevailing premises as well.)
- Similarly, $\to$i says, if you want to prove $\phi \to \psi$, try proving ψ from ϕ (and the other prevailing premises).
- $\neg$i says: to prove $\neg\phi$, prove $\bot$ from ϕ (and the other prevailing premises).

The basic rules of natural deduction:

	introduction	*elimination*
$\wedge$	$\dfrac{\phi \quad \psi}{\phi \wedge \psi}\ \wedge\mathrm{i}$	$\dfrac{\phi \wedge \psi}{\phi}\ \wedge\mathrm{e}_1 \qquad \dfrac{\phi \wedge \psi}{\psi}\ \wedge\mathrm{e}_2$
$\vee$	$\dfrac{\phi}{\phi \vee \psi}\ \vee\mathrm{i}_1 \qquad \dfrac{\psi}{\phi \vee \psi}\ \vee\mathrm{i}_2$	$\dfrac{\phi \vee \psi \quad \boxed{\begin{array}{c}\phi\\ \vdots\\ \chi\end{array}} \quad \boxed{\begin{array}{c}\psi\\ \vdots\\ \chi\end{array}}}{\chi}\ \vee\mathrm{e}$
$\rightarrow$	$\dfrac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \psi\end{array}}}{\phi \rightarrow \psi}\ \rightarrow\mathrm{i}$	$\dfrac{\phi \quad \phi \rightarrow \psi}{\psi}\ \rightarrow\mathrm{e}$
$\neg$	$\dfrac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \bot\end{array}}}{\neg\phi}\ \neg\mathrm{i}$	$\dfrac{\phi \quad \neg\phi}{\bot}\ \neg\mathrm{e}$
$\bot$	(no introduction rule for $\bot$)	$\dfrac{\bot}{\phi}\ \bot\mathrm{e}$
$\neg\neg$		$\dfrac{\neg\neg\phi}{\phi}\ \neg\neg\mathrm{e}$

Some useful derived rules:

$$\dfrac{\phi \rightarrow \psi \quad \neg\psi}{\neg\phi}\ \mathrm{MT} \qquad\qquad \dfrac{\phi}{\neg\neg\phi}\ \neg\neg\mathrm{i}$$

$$\dfrac{\boxed{\begin{array}{c}\neg\phi\\ \vdots\\ \bot\end{array}}}{\phi}\ \mathrm{PBC} \qquad\qquad \dfrac{}{\phi \vee \neg\phi}\ \mathrm{LEM}$$

Figure 1.2. Natural deduction rules for propositional logic.

At any stage of a proof, it is permitted to introduce any formula as assumption, by choosing a proof rule that opens a box. As we saw, natural deduction employs boxes to control the scope of assumptions. When an assumption is introduced, a box is opened. Discharging assumptions is achieved by closing a box according to the pattern of its particular proof rule. It's useful to make assumptions by opening boxes. *But don't forget you have to close them in the manner prescribed by their proof rule.*

OK, but how do we actually go about constructing a proof? Given a sequent, you write its premises at the top of your page and its conclusion at the bottom. Now, you're trying to fill in the gap, which involves working simultaneously on the premises (to bring them towards the conclusion) and on the conclusion (to massage it towards the premises).

Look first at the conclusion. If it is of the form $\phi \rightarrow \psi$, then apply[6] the rule $\rightarrow$i. This means drawing a box with ϕ at the top and ψ at the bottom. So your proof, which started out like this:

$$\begin{array}{c} \vdots \\ \text{premises} \\ \vdots \\ \phi \rightarrow \psi \end{array}$$

now looks like this:

$$\begin{array}{c} \vdots \\ \text{premises} \\ \vdots \end{array}$$

$$\begin{array}{ll} \boxed{\begin{array}{ll} \phi & \qquad\qquad \text{assumption} \\ & \\ & \\ \psi & \end{array}} & \\ \phi \rightarrow \psi & \qquad \rightarrow\text{i} \end{array}$$

You still have to find a way of filling in the gap between the ϕ and the ψ. But you now have an extra formula to work with and you have simplified the conclusion you are trying to reach.

[6] Except in situations such as $p \rightarrow (q \rightarrow \neg r), p \vdash q \rightarrow \neg r$ where $\rightarrow$e produces a simpler proof.

The proof rule ¬i is very similar to →i and has the same beneficial effect on your proof attempt. It gives you an extra premise to work with and simplifies your conclusion.

At any stage of a proof, several rules are likely to be applicable. Before applying any of them, list the applicable ones and think about which one is likely to improve the situation for your proof. You'll find that →i and ¬i most often improve it, so always use them whenever you can. There is no easy recipe for when to use the other rules; often you have to make judicious choices.

1.2.4 Provable equivalence

Definition 1.25 Let ϕ and ψ be formulas of propositional logic. We say that ϕ and ψ are *provably equivalent* iff (we write 'iff' for 'if, and only if' in the sequel) the sequents $\phi \vdash \psi$ and $\psi \vdash \phi$ are valid; that is, there is a proof of ψ from ϕ and another one going the other way around. As seen earlier, we denote that ϕ and ψ are provably equivalent by $\phi \dashv\vdash \psi$.

Note that, by Remark 1.12, we could just as well have defined $\phi \dashv\vdash \psi$ to mean that the sequent $\vdash (\phi \to \psi) \wedge (\psi \to \phi)$ is valid; it defines the same concept. Examples of provably equivalent formulas are

$$\begin{array}{ll} \neg(p \wedge q) \dashv\vdash \neg q \vee \neg p & \neg(p \vee q) \dashv\vdash \neg q \wedge \neg p \\ p \to q \dashv\vdash \neg q \to \neg p & p \to q \dashv\vdash \neg p \vee q \\ p \wedge q \to p \dashv\vdash r \vee \neg r & p \wedge q \to r \dashv\vdash p \to (q \to r). \end{array}$$

The reader should prove all of these six equivalences in natural deduction.

1.2.5 An aside: proof by contradiction

Sometimes we can't prove something *directly* in the sense of taking apart given assumptions and reasoning with their constituents in a constructive way. Indeed, the proof system of natural deduction, summarised in Figure 1.2, specifically allows for *indirect* proofs that lack a constructive quality: for example, the rule

$$\frac{\boxed{\begin{array}{c} \neg\phi \\ \vdots \\ \bot \end{array}}}{\phi}\ \text{PBC}$$

allows us to prove ϕ by showing that $\neg\phi$ leads to a contradiction. Although 'classical logicians' argue that this is valid, logicians of another kind, called 'intuitionistic logicians,' argue that to prove ϕ you should do it directly, rather than by arguing merely that $\neg\phi$ is impossible. The two other rules on which classical and intuitionistic logicians disagree are

$$\frac{}{\phi \vee \neg\phi}\ \text{LEM} \qquad \frac{\neg\neg\phi}{\phi}\ \neg\neg\text{e}.$$

Intuitionistic logicians argue that, to show $\phi \vee \neg\phi$, you have to show ϕ, or $\neg\phi$. If neither of these can be shown, then the putative truth of the disjunction has no justification. Intuitionists reject $\neg\neg$e since we have already used this rule to prove LEM and PBC from rules which the intuitionists do accept. In the exercises, you are asked to show why the intuitionists also reject PBC.

Let us look at a proof that shows up this difference, involving real numbers. Real numbers are floating point numbers like 23.54721, only some of them might actually be infinitely long such as 23.138592748500123950734..., with no periodic behaviour after the decimal point.

Given a positive real number a and a *natural* (whole) number b, we can calculate a^b: it is just a times itself, b times, so $2^2 = 2 \cdot 2 = 4$, $2^3 = 2 \cdot 2 \cdot 2 = 8$ and so on. When b is a *real* number, we can also define a^b, as follows. We say that $a^0 \stackrel{\text{def}}{=} 1$ and, for a non-zero rational number k/n, where $n \neq 0$, we let $a^{k/n} \stackrel{\text{def}}{=} \sqrt[n]{a^k}$ where $\sqrt[n]{x}$ is the real number y such that $y^n = x$. From real analysis one knows that any real number b can be approximated by a sequence of rational numbers k_0/n_0, k_1/n_1, ... Then we define a^b to be the real number approximated by the sequence a^{k_0/n_0}, a^{k_1/n_1}, ... (In calculus, one can show that this 'limit' a^b is unique and independent of the choice of approximating sequence.) Also, one calls a real number *irrational* if it can't be written in the form k/n for some integers k and $n \neq 0$. In the exercises you will be asked to find a semi-formal proof showing that $\sqrt{2}$ is irrational.

We now present a proof of a fact about real numbers in the informal style used by mathematicians (this proof can be formalised as a natural deduction proof in the logic presented in Chapter 2). The fact we prove is:

Theorem 1.26 *There exist irrational numbers a and b such that a^b is rational.*

PROOF: We choose b to be $\sqrt{2}$ and proceed by a case analysis. Either b^b is irrational, or it is not. (Thus, our proof uses $\vee$e on an instance of LEM.)

(i) Assume that b^b is rational. Then this proof is easy since we can choose irrational numbers a and b to be $\sqrt{2}$ and see that a^b is just b^b which was assumed to be rational.

(ii) Assume that b^b is *ir*rational. Then we change our strategy slightly and choose a to be $\sqrt{2}^{\sqrt{2}}$. Clearly, a is irrational by the assumption of case (ii). But we know that b is irrational (this was known by the ancient Greeks; see the proof outline in the exercises). So a and b are both irrational numbers and

$$a^b = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{(\sqrt{2}\cdot\sqrt{2})} = (\sqrt{2})^2 = 2$$

is rational, where we used the law $(x^y)^z = x^{(y \cdot z)}$.

Since the two cases above are exhaustive (*either* b^b is irrational, *or* it isn't) we have proven the theorem. □

This proof is perfectly legitimate and mathematicians use arguments like that all the time. The exhaustive nature of the case analysis above rests on the use of the rule LEM, which we use to prove that either b is rational or it is not. Yet, there is something puzzling about it. Surely, we have secured the fact that there are irrational numbers a and b such that a^b is rational, but are we in a position to specify an actual pair of such numbers satisfying this theorem? More precisely, which of the pairs (a, b) above fulfils the assertion of the theorem, the pair $(\sqrt{2}, \sqrt{2})$, or the pair $(\sqrt{2}^{\sqrt{2}}, \sqrt{2})$? Our proof tells us nothing about *which* of them is the right choice; it just says that at least one of them works.

Thus, the intuitionists favour a calculus containing the introduction and elimination rules shown in Figure 1.2 and excluding the rule $\neg\neg$e and the derived rules. Intuitionistic logic turns out to have some specialised applications in computer science, such as modelling type-inference systems used in compilers or the staged execution of program code; but in this text we stick to the full so-called classical logic which includes all the rules.

1.3 Propositional logic as a formal language

In the previous section we learned about propositional atoms and how they can be used to build more complex logical formulas. We were deliberately informal about that, for our main focus was on trying to understand the precise mechanics of the natural deduction rules. However, it should have been clear that the rules we stated are valid for *any* formulas we can form, as long as they match the pattern required by the respective rule. For example,