Chapter A1

PL Proofs Introduced

Outside the logic classroom, when we want to convince ourselves that an inference is valid, we don’t often use techniques like the truth-table test or tree test. Instead we seek *proofs*—we try to derive the desired conclusion from the premisses by chaining together a sequence of obviously acceptable steps into a multi-step argument (see Chapter 5). This chapter begins to describe a formal framework for constructing PL proofs.

A1.1 Choices, choices …

There is, however, no single ‘right’ way of laying out proofs. And there is no single set of rules of inference that which is the ‘right’ set for classical propositional logic.

**Proof layouts.** In this book we have already dabbled with no less than three different ways of setting out multi-step arguments:

1. In Chapter 5, we met semi-formal proofs which were set out linearly, one step below another.
2. Then in Chapter 8 (see §8.2), we noted that some linear arguments can very naturally be rearranged into tree structures, with the initial assumptions appearing at the tips of branches, and the branches joining as we work down the tree until we are left with the final conclusion of the proof at the bottom of the trunk. (Our examples happened to feature arguments in augmented English about which strings of PL symbols are wffs.)
3. In Chapter 14, we found that some other arguments—namely ‘working backwards’ arguments about truth-valuations—are naturally set out as *inverted* trees. We start with some assumptions at the top of the tree, and work down trying to find contradictions, splitting branches at various choice points as we descend the downwards-branching tree.

That gives us three initial models to follow, if we want to develop proof structures for arguing in PL (and PLC, of course—I won’t keep adding that). Which should we choose?
Type-(3) trees are especially apt for setting out systematic attempts to find a reductio ad absurdum. (Recall, that’s what truth-trees are all about: we start by assuming the premisses and the negation of the conclusion of a given argument, and see whether these assumptions generate contradictions.) But not all ordinary proofs are indirect arguments by reductio; in fact, most of them aren’t. So it is natural to want ways of setting out arguments in PL that encompass direct as well as indirect arguments.

Both type-(1) linear proofs and type-(2) trees can encode direct arguments, in which we argue from initial premisses at the top to the final conclusion at the bottom. But for many purposes, the tree lay-out is to be preferred. Why? For a simple example, consider A in Chapter 8. We set out a linear and a tree proof that the string ‘¬(P ∧ ¬(¬Q ∨ R))’ is a wff. Here’s the linear version again, but without the commentary that helpfully tells us which step follows from which.

A

‘Q’ is a wff
‘¬Q’ is a wff
‘R’ is a wff
‘(¬Q ∨ R)’ is a wff
‘¬(¬Q ∨ R)’ is a wff
‘P’ is a wff
‘(P ∧ ¬(¬Q ∨ R))’ is a wff
‘¬(P ∧ ¬(¬Q ∨ R))’ is a wff

Inspection shows that each step is either a basic premiss, reminding us that an atom is a wff, or follows from earlier steps in accordance with the rules for building wffs. But, without the annotations, we have to search through previous steps to find which earlier propositions warrant the derivation of some later proposition. Contrast the tree proof:

A’

\[
\frac{\text{‘Q’ is a wff}}{\text{‘¬Q’ is a wff}} \quad \frac{\text{‘R’ is a wff}}{\text{‘(¬Q ∨ R)’ is a wff}} \quad \frac{\text{‘(P ∧ ¬(¬Q ∨ R))’ is a wff}}{\text{‘¬(¬Q ∨ R)’ is a wff}} \quad \frac{\text{‘¬(P ∧ ¬(¬Q ∨ R))’ is a wff}}{\text{‘¬(P ∧ ¬(¬Q ∨ R))’ is a wff}}
\]

Here, the result immediately below an inference line is derived from the proposition or pair of propositions immediately above the line. The geometric layout of the argument thus carries all the information we need about what depends on what. Which is very elegant.

Now, we noted in §5.4 that type-(1) linear proofs can also be adapted for the display of indirect arguments, by the use of the device of indenting the line of argument when a temporary supposition is in play. Type-(2) trees can also be adapted to encode indirect arguments as well. There’s a cost in the form of added complexity, but trees still trump linear proofs on the grounds of technical elegance. That’s why logicians rightly tend to favour them as the better way of presenting proofs as objects for serious ‘proof-theoretic’ study.
However, experience suggests that such tree proofs are probably not the best place for beginners to start. So the plan in this book is as follows. We will first develop a proof-system for propositional logic based on linear proofs with indented ‘sub-proofs’. Then we’ll show how to convert these linear proofs into type-(2) tree form (in Chapter 21). That way, you’ll get to see the pros and cons of each approach, and also be well prepared for encountering a variety of proof systems in more advanced work.

Rules of inference A proof-system is characterized not just by our choice of permitted lay-outs for proofs, but also by our selection of the rules which allow us to deduce new wffs from previous steps in the proof. Different books rarely agree on their exact selection of rules, even when developing elementary propositional logic. That’s because we have to trade off various considerations of economy and simplicity, and there is no best way of doing this. However, the selection of rules we choose in this book is equivalent to those in other texts in the sense that it validates the same arguments; so we aren’t going to fuss too much about justifying our particular choices as against other equally good possibilities.

Note that finding proofs can involve ingenuity—and understanding how a given proof system works is one thing, having the ingenuity to discover proofs in the system is something else. There are exercises at the end of this and the next few chapters to help you test and develop your own proof-discovery skills. But don’t panic if these faze you. The really important thing for most of us is to be able to follow proofs, and see why they work.

A1.2 Rules for ‘∧’ and ‘¬’

Let’s warm up with a really trivial example. Consider the argument

B Popper is a philosopher and it isn’t the case that it isn’t true that Quine is one too. So Quine is a philosopher.

The inferential jump from the premiss to the conclusion is obviously valid. But we can, if we like, break it down into two even more obvious steps, as follows:

B′
(1) Popper is a philosopher and it isn’t the case that it isn’t true that Quine is one too. (Premiss)
(2) It isn’t the case that it isn’t true that Quine is a philosopher. (From 1)
(3) Quine is a philosopher. (From 2)

Rendering that into PL, using ‘P’ for ‘Popper is a philosopher’, and ‘Q’ for ‘Quine is a philosopher’, the given inference is

B” (P ∧ ¬¬Q) ⊃ Q

and the corresponding PL proof which shows that this inference is valid runs

B”’
(1) (P ∧ ¬¬Q) (Prem)
(2) ¬¬Q (1 ∧E)
(3) Q (2 DN)
We'll take the PL proof itself to be just the column of wffs. The line numbers on the left and the tersely abbreviated notes on the right are metalinguistic commentary (in abbreviated English).

The first line of our mini-proof is self explanatory. At the next line we infer a conjunct from a conjunction, i.e. we deploy the inference rule from a wff of the form \((A \land B)\) we can infer \(A\), or equally we can infer \(B\). This rule is tautologically sound (i.e. any application will yield a tautologically valid inference). Since applying the rule yields a conclusion from which the main connective of the premiss \((A \land B)\) has been eliminated, that invites the standard label ‘\(\land\)–elimination’ (or ‘\(\land\)E’ for short).

At the third line, we invoke another rule, Double Negation Elimination (or ‘DN’ for short): from a wff of the form \(\neg\neg A\), we can infer \(A\). That rule is also plainly sound.

So—labouring the obvious—we can get from the initial PL premiss to the conclusion by tautologically valid inference steps. Hence the overall leap from premiss to conclusion is valid (cf. Chapter 5).

Now take a more interesting case. Consider the argument

\[ C \quad \text{Popper is a philosopher. It isn’t the case that Popper is a philosopher and Quine isn’t. So Quine is a philosopher too.} \]

A moment’s reflection shows that this inference is also valid.

If you need to convince yourself of that, here’s an argument, set out in the style of §5.4:

\[ C' \]

\begin{align*}
(1) & \quad \text{Popper is a philosopher.} \quad \text{(Premiss)} \\
(2) & \quad \text{It isn’t the case that Popper is a philosopher and Quine isn’t.} \quad \text{(Premiss)} \\
(3) & \quad \text{Suppose temporarily, for the sake of argument:} \\
(4) & \quad \text{Quine is not a philosopher.} \quad \text{(Supposition)} \\
(5) & \quad \text{Popper is a philosopher and Quine isn’t.} \quad \text{(From 1, 3)} \\
(6) & \quad \text{Contradiction!} \quad \text{(From 4, 2)} \\
(7) & \quad \text{So the supposition that leads to this absurdity must be wrong.} \\
(8) & \quad \text{It isn’t the case that Quine is not a philosopher} \quad \text{(RAA)} \\
(9) & \quad \text{Quine is a philosopher.} \quad \text{(From 6)}
\end{align*}

Remember, when we make a new temporary supposition we indent the line of argument one step to the right (marking the new column with the vertical line); and we keep the argument indented so long as the supposition is in play. When we hit a contradiction and conclude that the supposition is wrong, we discharge that supposition; we move the line of argument back a column, thereby signalling that the supposition (and the sub-proof that depends on it) is no longer in play.

Now let’s render all this into PL. Using the obvious translation key, what we want to show is that the following inference is valid:

\[ C'' \]

\[ P, \neg (P \land \neg Q) \vdash Q \]

And here’s an annotated PL proof that encodes the semi-formal proof we have just given:
This time, the first three lines are self-explanatory. From lines (1) and (3), we then infer their conjunction (4). The effect is to introduce an occurrence of ‘\(\land\)’. The inference rule we are applying here—from a wff \(A\) and a wff \(B\) we can infer their conjunction \((A \land B)\)—is thus naturally called ‘\(\land\)-Introduction’ (or ‘\(\land I\)’ for short). The rule is sound for bare conjunction.

We now have (4) and (2). Their joint assertion is absurd, and we next explicitly signal the absurdity. The rule is roughly, given a contradictory pair of the form \(A\) and \(\neg A\), we can introduce the absurdity marker ‘\(\ast\)’. Call that ‘the absurdity rule’ or ‘Abs’ for short.

So, the supposition (3) leads to absurdity; we can conclude that not-(3). That’s a reductio ad absurdum inference (hence ‘RAA’ for short). And the conclusion that (3) is false of course no longer depends on assuming that (3) is true: the supposition is discharged, and we move the line of argument back into the initial column. The rule RAA says, roughly, that if the supposition \(A\) leads to absurdity, then we can discharge the supposition and conclude \(\neg A\). Note that an application of RAA always introduces a new negation sign. So in the present case, we get (6), ‘\(\neg \neg Q\)’.

Hence, to derive the desired conclusion ‘\(Q\)’, we need to use the rule DN again. And then we are done.

We now have five rules of inference in play, \(\land I\), \(\land E\), Abs, RAA, DN. We can schematically represent them as follows:

\[
\begin{array}{c}
A \\
\vdash (A \land B) \\
B \\
\vdash \vdash A \quad (\land E) \\
\vdash (A \land B) \quad (\land I) \\
\hline
A \\
\vdash A \\
\vdash \vdash \neg A \\
\vdash \vdash \vdash \vdash \ast \\
\vdash \ast A \quad (DN) \\
\vdash \ast \neg A \quad (RAA)
\end{array}
\]
A6 An Introduction to Formal Logic

But be careful not to read too much into these schemas. For example, the previous wffs $A$ and $B$ from which $(A \land B)$ is inferred by $(\land I)$ don’t have to occur in the order $A$ first, then $B$. Nor (as saw in C″′) do they have to occur in the same column. Still, treated with caution, these schemas are a useful prop until we tidy things up officially in the next chapter.

Let’s immediately take a further pair of examples. Consider first the inference

\[ D \quad \text{Popper and Quine are philosophers. It isn’t the case that both Popper and Russell are philosophers. So it isn’t the case that both Quine and Russell are philosophers.} \]

Here is a semi-formal argument which shows that the inference is indeed valid:

\[ D' \]

(1) Popper and Quine are philosophers. (Premiss)
(2) It isn’t the case that both Popper and Russell are philosophers. (Premiss)

Suppose temporarily, for the sake of argument:

(3) Quine and Russell are philosophers. (Supposition)
(4) Popper is a philosopher. (From 1)
(5) Russell is a philosopher. (From 3)
(6) Popper and Russell are philosophers. (From 4, 5)

Contradiction!
(From 6, 2)

So the supposition that leads to this absurdity must be wrong.

(8) It isn’t the case that both Quine and Russell are philosophers (RAA)

Rendered into PL in the obvious way, the inference we want to warrant is

\[ D'' \quad (P \land Q), \neg (P \land R) \therefore \neg (Q \land R) \]

And the intuitive proof gets rendered as follows:

\[ D''' \]

(1) $(P \land Q)$ (Prem)
(2) $\neg (P \land R)$ (Prem)
(3) $(Q \land R)$ (Supp)
(4) $P$ (1 $\land E$)
(5) $R$ (3 $\land E$)
(6) $(P \land R)$ (4, 5 $\land I$)
(7) * (6, 2 Abs)
(8) $\neg (Q \land R)$ (3–7 RAA)

And we are done.

For another very simple example, consider the inference

\[ E \quad \neg \neg (P \land Q) \therefore \neg \neg (Q \land P) \]

That’s obviously valid. To derive the conclusion by a PL proof, we just need to extract the conjuncts from the premiss and put them back together in the opposite order, and then introduce a double negation.

Now, we can’t directly apply the rule $\land E$ to the premiss (why?): we need first to strip off the negation signs with DN. So off we go ...
Proofs Introduced

A7

Now what? Do we need a kind of converse to the DN rule, a rule that allows us to introduce a couple of negation signs? No: we can just use RAA again, like this:

\[
\begin{align*}
(6) & \quad \neg (Q \land P) \quad \text{(Supp)} \\
(7) & \quad \ast \quad \text{(5,6 Abs)} \\
(8) & \quad \neg \neg (Q \land P) \quad \text{(6–7 RAA)}
\end{align*}
\]

And again we are done.

A quick remark on our conventions for recording the commentary on the right. The rule $\land I$, for example, takes two wffs as input and delivers their conjunction as output. In the commentary, we give the lines numbers of the input wffs in the order in which the conjuncts appear in the output. That's why we have ‘$(4,3 \land I)$’ in the current example. Likewise, an application of the Absurdity rule requires a pair of previous wffs of the form $A, \neg A$; and we give the line numbers of those wffs in that order, i.e. the number of the ‘bare’ wff first, the number of the negated wff second. That's why we had ‘$(6,2 \text{ Abs})$’ in the previous example. Finally an application of RAA requires an indented ‘sub-proof’ running from an initial supposition to a final absurdity. The commentary gives the line number range of the whole sub-proof.

A1.3 More examples

We have now encountered five basic rules for arguing with conjunction and negation in PL: in this section, we'll take some more examples of our rules in action, emphasizing our convention for laying out proofs involving temporary suppositions. We'll be giving some tips about proof discovery as we go; but—to repeat—what is crucial is that you learn to follow proofs and see why they work.

So next consider the inference

\[
\begin{align*}
F & \quad \neg (P \land Q), \quad \neg (\neg Q \land \neg R), \quad P \therefore \quad \neg \neg P \land R
\end{align*}
\]

which in fact is valid (interpret the atoms however you like). How can we prove this? To establish a conjunction we typically need to establish each conjunct. The first conjunct ‘$\neg \neg P$’ quickly follows from the premiss ‘$P$’, thus:

\[
\begin{align*}
(1) & \quad \neg (P \land Q) \quad \text{(Prem)} \\
(2) & \quad \neg (\neg Q \land \neg R) \quad \text{(Prem)} \\
(3) & \quad P \quad \text{(Prem)} \\
(4) & \quad \neg P \quad \text{(Supp)} \\
(5) & \quad \ast \quad \text{(3,4 Abs)} \\
(6) & \quad \neg \neg P \quad \text{(4–5 RAA)}
\end{align*}
\]
Now we need to get to ‘R’. There’s more than one route to that conclusion, but here’s a simple one. First we note that ‘P’ and ‘¬(P ∧ Q)’ intuitively entail ‘¬Q’; and we’ll be able to establish that by assuming ‘Q’ and deriving a contradiction. Once we’ve got ‘¬Q’, that together with ‘¬Q ∧ ¬R)’ intuitively entails ‘¬¬R’, and we can establish that entailment by another reductio proof. And then we’ll be more or less done. (Before looking at the formal version which follows, check that you understand informally why this strategy should work.)

Note carefully how our convention for laying out proofs—indenting when we make a supposition, moving back a column when we discharge it—helps us keep track of what is going on, by making it very clear which of the three temporary suppositions is in play at which stage of the proof.

This convention really comes into its own, however, when we have more than one supposition in play at the same time. Here’s another inference:

\[
G \quad \neg(P \land \neg Q) \quad \therefore \quad \neg((P \land R) \land \neg(Q \land R)).
\]

This is tautologically valid. (That may not immediately be obvious. But recall that \(\neg(A \land \neg C)\) is equivalent to \((A \supset C)\); so \(G\) is truth-functionally equivalent to

\[
G' \quad (P \supset Q) \quad \therefore \quad ((P \land R) \supset (Q \land R))
\]

which is intuitively valid; for given if \(P\) then \(Q\), it follows that if \(P\) and also \(R\), then \(Q\) and also \(R\).)

Can we validate \(G\) using our rules? We are trying to prove something of the form \(\neg A\). The obvious ploy is again to suppose \(A\) to be true and seek a contradiction. So we begin like this:

\[
G^* \quad (1) \quad \neg(P \land \neg Q) \quad \text{(Prem)}
\]

(2) \[(P \land R) \land \neg(Q \land R)] \quad \text{(Supp)}
\]

We can immediately proceed to extract information from the conjunctive (2), by repeatedly using \(\land E\):

\[
(3) \quad (P \land R) \quad \text{(2 \& E)}
\]

(4) \[(Q \land R) \quad \text{(2 \& E)}
\]

(5) \(P \quad \text{(3 \& E)}
\]

(6) \(R \quad \text{(3 \& E)}
\]

So far, pretty automatic: but now what? We want to find an absurdity (we are aiming for an application of RAA); and since we haven’t yet made use of premiss
the obvious thing to do is try to find its contradictory, i.e. what we need to prove is \((P \land \neg Q)\). And—intuitively—(4) and (6) do indeed lead to \(\neg Q\), which together with (5) will give us just what we want. But how do (4) and (6) get us to \(\neg Q\)? We’ve seen this kind of move before; we need to suppose \(Q\) and get a contradiction. This requires us, then, to make a second supposition: so to signal this fact, we indent the chain of argument another step, thus (starting over again to make the structure perfectly clear):

\[
\begin{array}{ll}
G^* & (1) \neg(P \land \neg Q) \quad \text{(Prem)} \\
 & (2) \neg((P \land R) \land \neg(Q \land R)) \quad \text{(Supp)} \\
 & (3) (P \land R) \quad (2 \land E) \\
 & (4) \neg(Q \land R) \quad (2 \land E) \\
 & (5) P \quad (3 \land E) \\
 & (6) R \quad (3 \land E) \\
 & (7) Q \quad (Supp) \\
 & (8) (Q \land R) \quad (7,6 \land I) \\
 & (9) \ast \quad (8,4 \land E) \\
 & (10) \neg Q \quad (7–9 \land E)
\end{array}
\]

We have now shown that \(\neg Q\), as planned; and having finished with the second supposition, we move one column leftwards (we are no longer in the part of the argument which depends on the supposition (7)). However, as is clear from the layout, the original supposition that we made at line (2) is still in play. But we are nearly home …

\[
\begin{array}{ll}
 & (11) (P \land \neg Q) \quad (5,10 \land E) \\
 & (12) \ast \quad (11, 1 \land E) \\
 & (13) \neg((P \land R) \land \neg(Q \land R)) \quad (2–12 \land E)
\end{array}
\]

We are now back in the ‘home’ column, and all suppositions are discharged.

Here’s another example where we need to nest sub-proofs two deep. The following is valid by the truth-table test:

\[
\begin{array}{ll}
H & \neg(P \land Q), \neg(P \land R) \vdash \neg(P \land \neg(Q \land R))
\end{array}
\]

To validate this argument in our proof system, we need to establish a negated wff, so the obvious policy is once more to assume the opposite, quickly extract the two conjuncts, and then aim for a reductio. In other words, the overall shape of the proof will look like this:

\[
\begin{array}{ll}
 & (1) \neg(P \land Q) \quad \text{(Prem)} \\
 & (2) \neg(P \land R) \quad \text{(Prem)} \\
 & (3) (P \land \neg((Q \land R) \land \neg(Q \land R))) \quad \text{(Supp)} \\
 & (4) P \quad (3 \land E) \\
 & (5) \neg((Q \land R) \land \neg(Q \land R)) \quad (3 \land E) \\
 & \ast \quad \vdots \\
 & (n+1) \neg(P \land \neg(Q \land R)) \quad (3–n \land E)
\end{array}
\]

How should we fill in the dots?
We can’t apply any of our rules of inference. (Note that it would be a horrible howler to think ‘Ah, \(\neg(\neg Q \land \neg R)\) starts with two negation signs so we can infer \(Q\)’). So we need to make another temporary supposition. What new supposition will it be worth making? Well, (4) combined with ‘\(Q\)’ gives us a straight contradiction with (1); so we can derive ‘\(\neg Q\)’ by reductio. Similarly, (4) combined with \(R\) gives us a straight contradiction with (2); so we can similarly derive ‘\(\neg R\)’ by reductio. But note that this yields ‘\((\neg Q \land \neg R)\)’ which contradicts (5). Or putting that all together, we have:

\[
\begin{array}{c}
H' \\
(1) & \neg(P \land Q) & \text{(Prem)} \\
(2) & \neg(P \land R) & \text{(Prem)} \\
(3) & (P \land \neg(\neg Q \land \neg R)) & \text{(Supp)} \\
(4) & P & \text{(3 \&E)} \\
(5) & \neg(\neg Q \land \neg R) & \text{(3 \&E)} \\
(6) & Q & \text{(Supp)} \\
(7) & (P \land Q) & \text{(4,6 \&I)} \\
(8) & \ast & \text{(7,1 Abs)} \\
(9) & \neg Q & \text{(6–8 RAA)} \\
(10) & R & \text{(Supp)} \\
(11) & (P \land R) & \text{(4,10 \&I)} \\
(12) & \ast & \text{(11,2 Abs)} \\
(13) & \neg R & \text{(10–12 RAA)} \\
(14) & (\neg Q \land \neg R) & \text{(9,13 \&I)} \\
(15) & \ast & \text{(14,5 Abs)} \\
(16) & \neg(P \land \neg(\neg Q \land \neg R)) & \text{(3–15 PAA)} \\
\end{array}
\]

Make sure you follow the columnar structure of the proof. In particular, note how the display makes it clear that the new supposition made at (6) is discharged before we make our third supposition at (10).

### A1.4 Other rules?

We’ve seen, then, some first examples of PL proofs using the rules \&I, \&E, Abs, RAA, DN. But are there other rules for making inferences in ‘\&’ and ‘\neg’ which we could legitimately introduce? Are there other rules we need to introduce?

The answer to the first question is (obviously) ‘yes’. For example, as we noted in passing before (see \(E'\)), alongside the DN elimination rule from a uff of the form \(\neg \neg A\), we can infer \(A\), we could equally well introduce the converse double negation introduction rule from a uff of the form \(A\), we can infer \(\neg \neg A\). But it would be redundant to do so. For whenever we had occasion to invoke that DNI rule, we could equally well get to the same result by using RAA in the following three-step pattern:

\[
\begin{align*}
(m) & \quad A \\
\ldots & \\
(n) & \quad \neg A & \text{(Supp)} \\
(n+1) & \quad \ast & \text{(m,n Abs)} \\
(n+2) & \quad \neg \neg A & \text{(n,n+1 RAA)}
\end{align*}
\]
Likewise, it would be perfectly legitimate to introduce another rule governing bare conjunction—we might have called it ‘∧S’ (short for ‘∧-symmetry’): *given (A \land B), we can infer (B \land A).* This would be a perfectly acceptable rule of inference to adopt—it is as obviously correct as the other rules for bare conjunction (if not for all cases of ordinary ‘and’). So why not adopt it? Again, it would be redundant. Instead of using the ∧S rule at some stage in a proof to move from (A \land B) to (B \land A), we can get the same effect by another three-step manoeuvre which fills in the schema

\[
\begin{array}{c}
(m) & (A \land B) \\
\vdots & \\
(n) & A & (m \land E) \\
(n+1) & B & (m \land E) \\
(n+2) & (B \land A) & (n+1, n \land I)
\end{array}
\]

In so far as we favour *economy in the length of proofs*, we have some very small reason for adopting the new rules DNI and ∧S, as that would avoid ever having to use either of those three-step shuffles. But in so far as we favour *economy in the number of rules*, that counts against adopting the rules, given that we can do without them. Logicians—who like to keep proof systems uncluttered—think that the gain in the first sort of economy is not worth the cost in terms of the second sort of economy; so we won’t be adding those rules to our basic set of inference rules.

Our second question was: is there any other rule we *do* need to add? The answer is ‘no’. This isn’t obvious (though a bit of experimentation should at least convince you that it is a plausible claim). But it can be shown that our set of rules is a *complete* set of rules for conjunction and negation. Or to put it more carefully: any tautologically valid PL inference (involving wffs only using the connectives ‘∧’ and ‘¬’) can be given a proof using just our five rules. We’ll prove this completeness result later, in Chapter 22.

So, our five rules form one complete set of rules for conjunction and negation. But (as we indicated in §1) there is no unique ‘best’ system of rules. To introduce just one variant system, recall that our version of RAA always introduces a negation sign: *if the supposition A leads to absurdity, then we can discharge the supposition and conclude ¬A.* But we could also have introduced a companion rule—often called ‘classical reductio’ (so ‘CR’ for short)—that runs: *if the supposition ¬A leads to absurdity, then we can discharge the supposition and conclude A.* Now this would be a redundant addition to our system, since (as we’ve seen) as we can mimic its effect using the existing rules RAA and DN, thus:

\[
\begin{array}{c}
(m) & \neg A & \text{(Supp)} \\
(n) & \ast & \text{(Abs)} \\
(n+1) & \neg A & (m-n \text{ RAA}) \\
(n+2) & A & (n+1 \text{ DN})
\end{array}
\]

But conversely, if we have CR, we can mimic the effect of DN:
So we could drop the rule DN and replace it by CR without changing the overall power of our proof system.

A1.5 Some clarifications

If you have carefully followed the examples in §§2 and 3, you should have a basic grip on the principles of our PL proof system for ‘∧’ and ‘¬’ (even if the presentation has been a bit quick and dirty). The system will be extended to deal with ‘∨’ in the next chapter; and only when we’ve done that will we tidy up the rules and regulations into an official version. But we should perhaps pause for some final clarifications in this chapter.

The first, and perhaps crucial, point to emphasize (or re-emphasize) can be introduced by means of an example. The inference

I  ¬(P ∧ Q) ∴ ¬P

is plainly invalid (compare ‘It isn’t the case that Romeo and Juliet are both male, so it isn’t the case that Romeo is male’). But now consider the following:

I′  (1) ¬(P ∧ Q)  (Prem)
    (2) P  (Supp)
    (3) Q  (Supp)
    (4) (P ∧ Q)  (2,3 ∧I)
    (5) *  (4,1 Abs)
    (6) ¬Q  (3–5 RAA)
    (7) Q  (4, ∧E)
    (8) *  (7,6 Abs)
    (9) ¬P  (2–8 RAA)

We seem to have proved that ‘¬P’ follows from ‘¬(P ∧ Q)’ after all! What has gone wrong?

A careful check shows that the howler is at line (7). Of course, given a conjunction ‘(P ∧ Q)’ we can infer ‘Q’. The trouble is that by line (7) we are no longer ‘given’ the conjunction—for that wff appears in a sub-proof which is over and done with (the supposition that heads the sub-proof has been discharged and is no longer in play). A key principle governing PL proofs of our type is:

When a sub-proof is finished, and the supposition at its head is discharged, we may no longer use wffs in the sub-proof as premisses to derive further wffs.

Strictly speaking, this principle errs on the side of caution. There can be wffs in a sub-proof that don’t actually depend on the discharged supposition at the head of the sub-proof but only on earlier wffs—see, for instance, D″ line (4). But
caution is out watchword, and we will write the full-strength principle into the official specification of our proof system. Another structural principle we’ve taken for granted but is worth highlighting is this:

A proof does not halt until all suppositions are discharged and the line of argument is back in the ‘home’ column.

Next, consider the inference

\[ J \quad P, \neg P \therefore Q \]

This is tautologically valid (there is no valuation of the atoms which makes the premises true and conclusion false because there is no valuation of the atoms which makes the premises true: cf. §§6.1, 13.8). But we just claimed that our proof system for arguments depending on ‘\(\land\)’ and ‘\(\neg\)’ is complete—so in particular, any tautologically valid argument involving just ‘\(\neg\)’ can be warranted by a proof in our system. But how can \(J\) be warranted? Well, consider first the simple and unproblematic proof

\[
\begin{align*}
J' & \quad (1) \quad \neg P \\
& \quad (2) \quad (P \land \neg Q) \\
& \quad (3) \quad P \\
& \quad (4) \quad * \\
& \quad (5) \quad \neg (P \land \neg Q) \\
\end{align*}
\]

Now recall the argument

\[
\begin{align*}
C'' & \quad (1) \quad P \\
& \quad (2) \quad \neg (P \land \neg Q) \\
& \quad (3) \quad \neg Q \\
& \quad (4) \quad (P \land \neg Q) \\
& \quad (5) \quad * \\
& \quad (6) \quad \neg \neg Q \\
& \quad (7) \quad Q
\end{align*}
\]

Graft these two together like this,

\[
\begin{align*}
J' & \quad (1) \quad P \\
& \quad (2) \quad \neg P \\
& \quad (3) \quad (P \land \neg Q) \\
& \quad (4) \quad P \\
& \quad (5) \quad * \\
& \quad (6) \quad \neg (P \land \neg Q) \\
& \quad (7) \quad \neg Q \\
& \quad (8) \quad (P \land \neg Q) \\
& \quad (9) \quad * \\
& \quad (10) \quad \neg \neg Q \\
& \quad (11) \quad Q
\end{align*}
\]

and we get the required proof of ‘\(Q\)’ from the premises ‘\(P\)’, ‘\(\neg P\)’.

Now, it has to be admitted that there is something a bit odd about this proof.
You might well think that as soon as we've got an explicit contradiction—as we have here in the initial premisses (1) and (2)—an alert logical referee should blow the whistle and stop the game. Our absurdity rule merely says that you can note a contradiction and close off a stretch of reasoning with the absurdity marker when you have a pair \( A, \neg A \) in play. It doesn't insist that you must stop a line of reasoning whenever a contradictory pair emerges in the line of argument. But you might reasonably retort 'If you hit a contradiction, that shows that something has already gone amiss already; ignoring it and letting the proof continue regardless will just get us into more trouble'.

On the other hand, if we allow the proofs \( J' \) and \( C''' \) while disallowing \( J'' \), that means we can't always chain short proofs into longer proofs. But the principle that short proofs can legitimately be chained together into longer ones seems to be of the very essence of the idea of a proof (i.e. the idea that if we can get from premisses \( A_i \) to \( C \) by a series of little steps that are separately in order, then the big leap from premisses to conclusion is also in order). And while there are ingenious suggestions on the market about how to finesse this point, the standard theory of proofs takes it that proofs can be freely chained together. Hence, on the standard theory, since the proofs \( J' \) and \( C''' \) are in order, their combination \( J'' \) must be acceptable as well.

A couple of final, much simpler, points. Note that the obviously valid inference

\[
\begin{align*}
K & \quad P \vdash (P \land P) \\
\end{align*}
\]

needs a corresponding PL proof if our system is to be complete—and it has one if we allow the following:

\[
\begin{align*}
K' & \quad (1) \quad P \quad (\text{Prem}) \\
& \quad (2) \quad (P \land P) \quad (1,1 \land I) \\
\end{align*}
\]

That is to say, we need to allow the same wff to be used twice over as the dual inputs to an application of a rule like \( \land I \). But that's surely acceptable. We have 'P' and we have 'P'—so we are entitled to infer '(P \land P)'.

What about the even simpler inference

\[
\begin{align*}
L & \quad P \vdash P \\
\end{align*}
\]

which is trivially valid (see §6.1)? We could extend the argument \( K' \) to yield

\[
\begin{align*}
L' & \quad (1) \quad P \quad (\text{Prem}) \\
& \quad (2) \quad (P \land P) \quad (1,1 \land I) \\
& \quad (3) \quad P \quad (2 \land E) \\
\end{align*}
\]

But while that is acceptable, we should surely be able to prove the validity of \( L \) without relying on rules governing conjunction (which doesn't feature in the original inference). And we can. Consider

\[
\begin{align*}
L'' & \quad (1) \quad P \quad (\text{Prem}) \\
\end{align*}
\]

That micro-proof is a proof, which has the single premiss 'P' and whose conclusion, i.e. whose final line, is ... 'P' again. (We'll see in the next chapter how one-
line proofs can play a more useful role: the point we are making here is simply that we *do* count them as legitimate.)

And now, at last, it is time to try your hand at producing some PL proofs …!

### A1.6 Summary

- One way of warranting an inference as valid is to produce a proof, showing that the conclusion can be reached from the premisses via a sequence of patently valid inference steps. There is, however, no one right way of structuring proofs, particularly proofs involving indirect reasoning (like reductio arguments). But one neat way (already prefigured in §5.4) is to use the Fitch’s device of indenting the line of argument every time a new supposition is made, and moving the line of argument back a column when the supposition is discharged.

- The five rules of inference ∧I, ∧E, Abs, RAA, DN (schematically displayed in §2) are evidently sound, and also form a complete set for PL inferences in conjunction and negation, in the sense any tautologically valid inference involving just those connectives can be shown to be valid by a proof using only those rules.

- When a sub-proof is finished, and the supposition at its head is discharged, we may no longer use wffs in the sub-proof to derive further wffs.

### Exercises A1

Use PL proofs to show that the following inferences are valid:

| i | (P ∧ ¬¬Q), ¬¬R : (Q ∧ R) |
| ii | (P ∧ (Q ∧ R)) : ((R ∧ P) ∧ Q) |
| iii | (¬P ∧ S) : ¬¬¬P |
| iv | ¬P, ¬¬¬P ∧ S : ¬S |
| v | (Q ∧ R), ¬¬¬¬S : (S ∧ S) |
| vi | ¬(P ∧ Q) : ¬(P ∧ (R ∧ Q)) |
| vii | P, ¬(P ∧ ¬Q), ¬(Q ∧ ¬¬R) : R |
| viii | ¬(P ∧ ¬¬Q), ¬(Q ∧ ¬¬R) : ¬(P ∧ ¬¬R) |
| ix | ¬(¬P ∧ ¬Q ∧ R), ¬(¬S ∧ P) : ¬(¬R ∧ S) |