2.4. Using lemmas

2.4.0. Overview

Although our system of derivations as it stands is both sound and complete, we will add rules that reflect the use of lemmas, both because of the importance of lemmas in ordinary explicit deductive reasoning and because the sorts of organization and simplification they provide in that context are of value here, too.

2.4.1. The dangers of lemmas
Although the use of lemmas is valuable in general, not all individual lemmas are valuable: the uncontrolled use of lemmas can lead to blind alleys or delay the progress of a derivation.

2.4.2. Lemmas for *reductio* arguments
A lemma that is entailed by the goal is always safe (though not necessarily progressive); this means that any lemma is safe when the goal is ⊥.

2.4.3. Attachment rules
Lemmas are certainly safe when we know we can prove them. We will use such lemmas to add to the available resources. The sentences added will generally be more complex than those already present, so this use of lemmas can interfere with decisiveness.

2.4.1. The dangers of lemmas

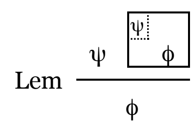
A fully general rule for introducing lemmas was cited in 2.3.2 as an example of an unsafe rule. That rule is unsafe because the lemma might not be a valid conclusion from our resources even though the conclusion we are trying the reach by way of it is a valid one. Such a rule would also prevent a system from being decisive because it would mean that a gap could always be developed further by introducing a lemma. However, as was noted earlier, more limited rules for introducing lemmas can be safe, and we will see that they can also be progressive. In this section we will look at the problems posed by lemmas more closely before considering a couple of special cases where these problems do not arise.

The law for lemmas of 1.4.2 can be stated as follows:

$\Gamma \Rightarrow \phi$ if both $\Gamma \Rightarrow \psi$ and $\Gamma, \psi \Rightarrow \phi$

Any possible world that is a counterexample to the first entailment will be a counterexample to one of the two on the right—the first of them if it makes ψ false and the second if it makes it true. So if both entailments on the right hold (that is, neither has a counterexample), the one on the left will hold, too. However, this principle holds only as an *if* claim; the corresponding *only if* does not hold in all cases. When $\Gamma \Rightarrow \phi$, we know that $\Gamma, \psi \Rightarrow \phi$ by the monotonicity of ⇒; but, since φ and ψ need have no connection with one another, knowing that $\Gamma \Rightarrow \phi$ would by itself give us no reason to suppose that $\Gamma \Rightarrow \psi$. Of course, in a case where we know that that $\phi \Rightarrow \psi$, we know $\Gamma \Rightarrow \psi$ because of the chain law; and we may know $\Gamma \Rightarrow \psi$ in other cases because of special connections between Γ and ψ. These are the two sorts of situation in which we will use lemmas but, before turning to them, let's look at what a fully general (and unsafe) rule for lemmas would be like.

If used in tree-form proofs this rule might take the following form:



Here the proof of φ divides into two branches. The first is a proof of the lemma ψ and the second is a proof of φ using ψ in addition to the already available assumptions and the exploitation chains that grow from them. The box around the right-hand branch is intended to indicate that the use of ψ is limited to that part of the proof, and ψ appears at the upper left of that box to indicate that it is available for this branch as a further assumption from which we may begin exploitation chains.

In derivations, we will use scope lines to mark the scope of added assumptions. And such assumptions are marked off from other resources along the scope line by the sort of horizontal line we use to mark off the initial premises of a derivation.

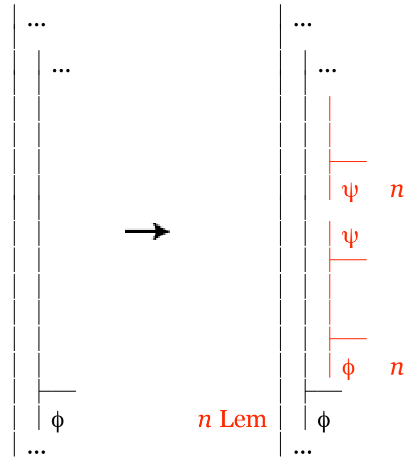


Fig. 2.4.1-1. Developing a derivation by introducing a lemma at stage *n* (a rule that will be part of our systems of derivations only in more restricted forms).

The assumption ψ is available only as long as the second of the two short scope lines continues, so it is effectively boxed off in derivations just as it is in the form of the rule displayed above for tree-form proofs.

The second of the two new gaps should be, if anything, easier to close than the original gap because it has a further resource. This increased ease is the point of introducing a lemma. The price we pay for this is the need to close the first new gap also. If the lemma is well chosen, reaching it may also be easier than closing the original gap by other means; but, because this rule is unsafe, we cannot be sure in general that the first new gap can be closed at all: ψ may not be a valid conclusion from its resources even if ϕ is. Because of this, the use of lemmas in ordinary deductive reasoning is accompanied by a readiness to backtrack, dropping the attempt to work by way of the lemma and looking for another approach to the proof. The notation of derivations is not designed to incorporate backtracking, so we will use lemmas only in cases where we can be sure there will be no need to do that.

Even in cases where we can sure backtracking is not necessary, the introduction of lemmas can interfere with decisiveness because it is in principle possible to keep introducing safe lemmas forever. In ordinary deductive reasoning, that problem is solved by good sense (or, if that's absent, by mortality). We will rely on those, too; but the system of derivations is designed to make the constraints on deductive reasoning explicit, so we will also consider the ways in which the use of safe lemmas might be limited further.

2.4.2. Lemmas for *reductio* arguments

We have seen that a lemma is bound to be safe if it is entailed by the goal we seek. That is, we can state following principle:

if $\phi \Rightarrow \psi$, then $\Gamma \Rightarrow \phi$ if and only if both $\Gamma \Rightarrow \psi$ and $\Gamma, \psi \Rightarrow \phi$

This tells us that when our desired conclusion ϕ implies a lemma ψ, we have not only the *if*-statement claim made by the law of lemmas but also the corresponding *only-if*-claim.

In principle, this idea can be applied to any case where the desired conclusion implies a sentence that might be useful in establishing it. We will build this into our system of derivations only in the case of the implications provided by EFQ, but implications provided by Ext are a better source of initial examples so we will first look at a use of that sort of lemma. Here is a derivation which uses the rule Lem to introduce a lemma that is the result of applying left Ext to the final goal.

```
       | A ∧ B                          1
       |-
1 Ext  | A                              (5),(9)
1 Ext  | B                              (4)
       | | | •
       | | |-
4 QED  | | | B                          3
       | | | •
       | | |-
5 QED  | | | A                          3
       | |-
3 Cnj  | | B ∧ A                        2
       | | B ∧ A                        (7),(10)
       | |-
       | | | •
       | | |-
7 QED  | | | B ∧ A                      6
       | | | | •
       | | | |-
9 QED  | | | | A                        8
       | | | | •
       | | | |-
10 QED | | | | B ∧ A                    8
       | | |-
8 Cnj  | | | A ∧ (B ∧ A)                6
       | |-
6 Cnj  | | (B ∧ A) ∧ (A ∧ (B ∧ A))      2
       |-
2 Lem  | (B ∧ A) ∧ (A ∧ (B ∧ A))
```

Here the rule Lem is applied at stage 2 with the left component of the goal as the lemma. This yields a slight shortening of the derivation since we are able to use the lemma to conclude B ∧ A by QED at stages 7 and 9 rather than repeating the proof used at stages 3-5 twice.

The simplification here is slight, and it occurs at all only because of a repetition in the goal that we would not expect to encounter often. While we would have more opportunities to use this sort of lemma in later chapters, there would not be enough to lead us to introduce a special rule, and this serves us only as an example. It is worth remembering, however, that it is legitimate pattern of deductive reasoning to first conclude one of the two components of a conjunction and then use that component to conclude the other (as we here have used the lemma B ∧ A to conclude A ∧ (B ∧ A)).

The pattern *Ex Falso Quodlibet* provides the basis for a much more important use of lemmas. An argument whose conclusion is ⊥ is often called a ***reductio argument***—where *reductio* is short for the Latin phrase *reductio ad absurdum* ('reduction to absurdity'). We will often need to use a lemma to complete such an argument and, since EFQ tells us that ⊥ entails any sentence, we know that any lemma we choose is safe. We will call the rule implementing this idea ***Lemma for Reductio*** or LFR:

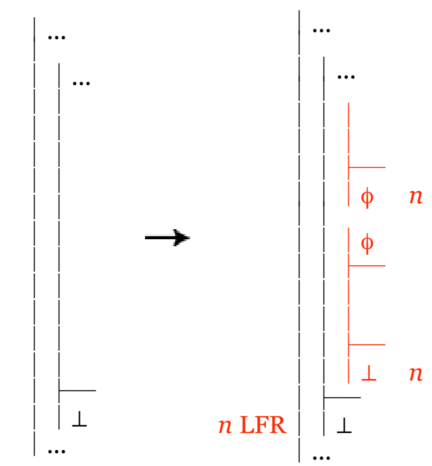


Fig. 2.4.2-1. Developing a derivation by introducing a lemma for a *reductio* at stage *n*.

We know this is safe from the general argument above, but it is also easy to see that directly. Any interpretation that divided either of the new gaps would certainly have to make all active resources of the original gap true; but an interpretation that did that would divide the original gap since the goal of that gap is ⊥ and is therefore bound to be false. So an interpretation cannot divide either of the new gaps unless it already divided their parent.

While this rule is certainly important, we are not yet in a position to illustrate it because, as yet, we have no non-trivial examples of formally valid *reductio* arguments. A *reductio* is formally valid only if its premises constitute a formally inconsistent set (that is, one whose members cannot be all true on any extensional interpretation), and the only formally inconsistent sets available with our current analyses of sentences contain ⊥ either as a premise or as a component of a premise. And we can show that such a set of premises entails ⊥ while using nothing but Ext and QED. This situation will change in the next chapter but, even there, our chief use of lemmas will be in a special modified version of LFR that is designed to exploit a resource at the same time it introduces a lemma.

That rule will be a direct rule, but rules that introduce lemmas usually will not be direct; and, in order to be sure a system employing them was decisive, we need to show that they can be considered progressive (on the right measure of distance from the end). The use of Lem illustrated first—i.e., a use where the lemma is a component of the goal—could be regarded as progressive provided we limited it to cases where the lemma was not already an active resource. But the second use of Lem—i.e., the use of LFR—can undermine decisiveness even if we forbid using a lemma that repeats a resource since the form of LFR places no constraints on the number of different lemmas that might be introduced. It would be enough to limit the lemmas that might be introduced to sentences that also appear as components of either active resources or goals, but weaker restrictions would also work. In general, we will not attempt to formulate the sort of restriction that would enable us to prove decisiveness for a system with LFR. The value of the rule is a practical one; and, in practice, the constraint of good sense in its use is restrictive enough.

2.4.3. Attachment rules

When discussing the minimal soundness of QED in 2.3.2 we saw that it would be legitimate for a rule to close a gap whenever its goal is entailed by active resources. (If that legitimacy seems obvious, remember that such a rule is only *minimally* sound because it can enable us to sidestep the consequences of unsafe rules used earlier in the derivation.) We will not employ such a sweeping rule but we will extend the use of QED (and later rules which use inactive resources) by rules which add new resources to those available in a gap without changing either exploiting active resources or analyzing the goal. The first example of such a rule is the following way of developing a gap, which we will call ***Adjunction***:

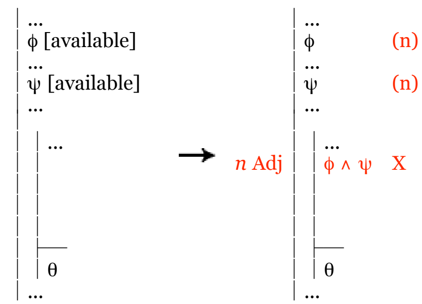


Fig. 2.4.3-1. Developing a derivation by applying Adj at stage *n*.

The added conjunction functions as a lemma, but this rule represents a way of using lemmas that has a number of special features. Notice that the lemma ϕ ∧ ψ does not lie to the right of a new scope line, as it does in the second gap introduced by LFR. There are two reasons for this. First, we have not branched the gap so the added resource, so there is no second gap to set outside the scope of the new resource. More importantly, we do not need to mark this new resource off as an added assumption made for a part of the derivation because it is entailed by resources already available throughout the gap we are developing.

Notice also that we treat this rule not as a way to plan for our goal but simply as a way to add resources. It does not exploit resources in order to add others, however, and the stage numbers to the right of the premises are parenthesized to indicate that they have not been exploited.

The X to the right of ϕ ∧ ψ is intended to indicate that this resource need not be exploited further. One way to think about this is to suppose that ϕ ∧ ψ has been introduced as something already exploited. That is, although it need not have been a once active but exploited resource (and there would be no point in adding it if it was), it has a status similar to such resources. It is marked at its right as exploited resources are, but it is not marked by a stage number because it was not exploited at some previous stage.

Adjunction is one example of a group of rules we will refer to as ***attachment rules***. Any such rule *R* will exhibit the following general pattern.

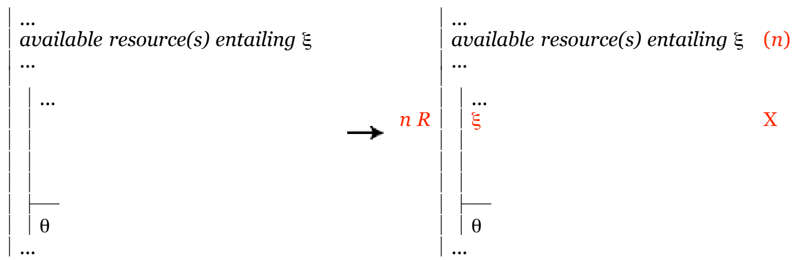


Fig. 2.4.3-2. Developing a derivation by applying an attachment rule *R* at stage *n*.

Since the added lemma does not become an active resource, the proximate argument of the child gap that is produced by an attachment rule is the same as the parent's proximate argument. As a result, safety and (utter) soundness hold just as they would for a gap that was completely unchanged. The only question about such a rule concerns its impact on the (minimal) soundness of rules like QED that use merely available resources. But any resource made available by attachment rule will be entailed by the active resources of the gap together with the active resources of its ancestors. Any interpretation that divides a gap and all its ancestors will then make such an resource true; and this will be enough for us to insure the soundness of rules using available resources. (To rehearse the argument for the case of QED again: if the goal of a gap is among its available resources, no interpretation can divide both it and all its ancestors because, to do this, an interpretation would need to make the goal false while making not only the active resources but also all other available resources true, and this is impossible when the goal is among the available resources. The key to this argument is being sure that all available resources are true whenever the active resources of a gap and all its ancestors are true, and attachment rules do not undermine our confidence in this.)

A rule like Adj clearly raises questions about decisiveness since the lemma it introduces is more complex than the premises it is based on. This increased complexity will be typical of attachment rules and is the reason for their name. A requirement that the lemma be a component of a goal or active resource would be a natural way of insuring decisiveness since such cases will represent the most valuable uses of attachment rules. Here, though, we need to remember that a sentence is a component of itself. Indeed, one common use of attachment rules will be to introduce the goal itself as an available resource in order to apply QED and close the gap. The following derivation is a simple example of this in the case of Adj.

	A ∧ B	1
	C	(4)
1 Ext	A	(3)
1 Ext	B	(4)
	•	
3 QED	A	2
4 Adj	B ∧ C	X,(5)
	•	
5 QED	B ∧ C	2
2 Cnj	A ∧ (B ∧ C)	

With two uses of Cnj, we would not have needed Adj and, with two uses of Adj, we would not have needed Cnj; but it is the sort of mixed use of the two illustrated here that brings us closest to typical patterns of explicit deductive argument.

2.4.s. Summary

The introduction of a lemma is one way of dividing up the work of a proof. We can implement this idea in derivations by dividing a gap into two, with one child having the lemma as a goal and the other having it as a further assumption to use in reaching the goal of the parent gap. The rule Lemma (Lem) that does this is not safe in general nor is it always progressive, and we will use only special instances of it.

A lemma is always safe when it is entailed by the goal it is designed to help us reach. The principal use of this idea will come in arguments whose goal is ⊥—that is, in *reductio* arguments. Since ⊥ entails any sentence a rule Lemma for *Reductio* (LFR), which allows free use of lemmas in *reductio* arguments will be safe (though some restriction on its use is needed to insure it is progressive).

A lemma is also safe if it is entailed by things we already know. Rules applying this idea will be designed for particular sorts of entailment and, since such a lemma is known to follow from our resources, there is no need to divide the gap or even introduce a new scope line. Indeed, we will use this sort of lemma only in attachment rules that add the lemma as an available but inactive resource. The first example of this sort of rule is Adjunction (Adj) which adds a conjunction when both conjuncts are already available. Although attachment rules can help us to close gaps sooner, the rules themselves are not direct, and some care is needed in their use if they are to be progressive.

The derivation rules we have so far are summarized in the table below. The names of the rules are links to the point in the text where they were initially described; look there to see the actual form taken by the rule.

Rules for developing gaps			*Rules for closing gaps*		
	for resources	*for goals*	*when to close*	*rule*	
conjunction $\phi \wedge \psi$	Ext	Cnj	the goal is also a resource	QED	
			⊤ is the goal	ENV	
			⊥ is a resource	EFQ	*Basic system*
			Attachment rule		
			added resource	*rule*	*Added rules (optional)*
			$\phi \wedge \psi$	Adj	

Glen Helman 09 Sep 2004

2.4.x. Exercise questions

Use the basic system of derivations along with the attachment rule Adj to establish the following. These repeat entailments from earlier exercises and examples (specifically, **b** and **d** of exercise 2.2.x.2, exercises **2** and **4** of 2.3.x, and the example of 2.4.2). They will work best as exercises in the use of Adj if you avoid using Cnj.

1. $A \Rightarrow A \wedge A$
2. $A \wedge B, B \wedge C, C \wedge D \Rightarrow A \wedge D$
3. $A \wedge B \Rightarrow A \wedge (B \wedge A)$
4. $A, B \wedge C, D \Rightarrow (C \wedge (B \wedge A)) \wedge B$
5. $A \wedge B \Rightarrow (B \wedge A) \wedge (A \wedge (B \wedge A))$

Glen Helman 01 Aug 2004

2.4.xa. Exercise answers

The answers below avoid the use of Cnj in order to maximize the use of the rule Adj. In some cases, a mixed use of the two rules would have produced a more natural argument.

1.

	A	(1)
1 Adj	$A \wedge A$	X,(2)
	•	
2 QED	$A \wedge A$	

2.

	$A \wedge B$	1
	$B \wedge C$	2
	$B \wedge D$	3
1 Ext	A	(4)
1 Ext	B	
2 Ext	B	
2 Ext	C	
3 Ext	B	
3 Ext	D	(4)
4 Adj	$A \wedge D$	X,(5)
	•	
5 QED	$A \wedge D$	

3.

	$A \wedge B$	1
1 Ext	A	(2),(3)
1 Ext	B	(2)
2 Adj	$B \wedge A$	X,(3)
3 Adj	$A \wedge (B \wedge A)$	X,(4)
	•	
4 QED	$A \wedge (B \wedge A)$	

4.

	A	(2)
	$B \wedge C$	1
	D	
1 Ext	B	(2),(4)
1 Ext	C	(3)
2 Adj	$B \wedge A$	X,(3)
3 Adj	$C \wedge (B \wedge A)$	X,(4)
4 Adj	$(C \wedge (B \wedge A)) \wedge B$	X,(5)
	•	
5 QED	$(C \wedge (B \wedge A)) \wedge B$	

5.

	$A \wedge B$	1
1 Ext	A	(2),(3)
1 Ext	B	(2)
2 Adj	$B \wedge A$	X,(3),(4)
3 Adj	$A \wedge (B \wedge A)$	X,(4)
4 Adj	$(B \wedge A) \wedge (A \wedge (B \wedge A))$	X,(5)
	●	
5 QED	$(B \wedge A) \wedge (A \wedge (B \wedge A))$	