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 provde in that context are of value for our work, too.

2.4.1. Premises, assumptions, and suppositions

We can ask whether a conclusion follows from assumptions we would not assert as premises, and one important device in proofs is the temporary consideration of assumptions that we merely *suppose* to be the case.

2.4.2. 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 us into blind alleys or delay the progress of a derivation.

2.4.3. Lemmas for *reductio* arguments

A lemma that is entailed by our goal is safe (though not necessarily progressive); this means that any lemma is safe when the goal is $\perp$.

2.4.4. 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 in this way may be more complex than those already present, so this use of lemmas can interfere with decisiveness.

Glen Helman 12 Jul 2012

2.4.1. Premises, assumptions, and suppositions

The rules using lemmas that we will discussion in the next couple of subsections employ a device that will be employed also in rules for most of the logical forms we will consider after conjunction. This device is the use, for stretches of a derivation, of assumptions that add to the content of the premises of the ultimate argument. We will refer to such assumptions as *suppositions*. Suppositions are assumptions made, not because we accept them but "for the sake of argument", and such assumptions are often introduced in ordinary argumentation by the verb suppose. That is, the stretch of discourse

> Suppose we do as you suggest. Then ....

comes to pretty much the same thing as

> Assume for the sake of argument that we do as you suggest. Then ....

And notice that the speaker here is not committed to accepting the suggestion.

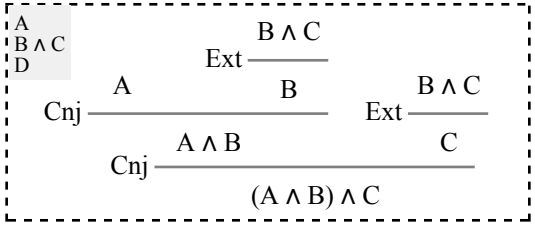
Suppositions can have a variety of roles in deductive reasoning, and the ones that will be important in later chapters are mostly rather different from their role in connection with lemmas, but all uses of suppositions have common features that can be captured by each of the ways of writing proofs that we have seen. In the case of conclusion trees, our approach to suppositions will be based on a more careful consideration of the role of premises. Indeed, premises, although not assumed *only* for the sake of argument are assumptions that are stated for its sake.

Now, consider again the first of the conclusion trees discussed in 2.2.1 .

$$\dfrac{\dfrac{\dfrac{B \wedge C}{A}\,\text{Ext} \qquad B}{A \wedge B}\,\text{Cnj} \qquad \dfrac{B \wedge C}{C}\,\text{Ext}}{(A \wedge B) \wedge C}\,\text{Cnj}$$

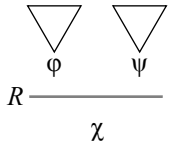
We looked at it there as a proof showing the validity of the argument A, B ∧ C / (A ∧ B) ∧ C. But it is equally well a proof for the argument A, B ∧ C, D / (A ∧ B) ∧ C. The extra premise D in the latter argument does not interfere because it remains true and all conclusions at the tips of branches are among the premises.

We can add a specification of the argument for which the conclusion tree is a proof by means of the following notation:



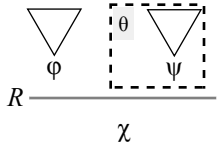
The premises are listed at the upper left, and the dotted line around this list and the tree indicates the *scope* of these assumptions. The tie between the list and the tree then lies in a requirement that the tip of every branch must be one of the assumptions within whose scope it lies. Although this notation for the scope of assumptions is specific to conclusion trees, the idea is not. In particular, one function of a scope line in derivations is to mark the scope of any assumptions at its top.

A two-premised argument like Cnj will appear in the text of a conclusion tree in the following way:



Here the triangles above the two premises represent any part of the conclusion tree growing up from that point (and there need be none since one or another premise might be the tip of a branch). This can be thought as a representation of a rule for building conclusion trees (from tip to root) by drawing further conclusions. A rule like the one shown here takes two conclusion trees and makes a larger one from them.

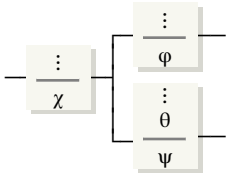
Now we might imagine a different sort of rule that applies in the following way:



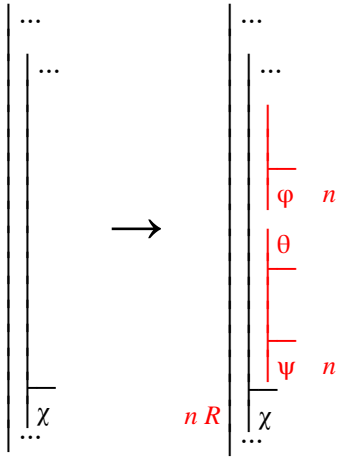
Here the second conclusion tree comes with an assumption θ whose scope surrounds the conclusion tree with ψ as its root. Although θ may appear as the tip of any branch above ψ, it will not be available to use as the tip of a branch above φ since that part of the conclusion tree is outside its scope. So θ will count as a supposition that is added to whatever assumptions have the whole larger tree in their scope.

Although the pattern shown above is the one we will encounter with lemmas, it is not the only one possible. Rules involving suppositions can apply to differing numbers of smaller conclusion trees, and differing numbers of these may come with suppositions. But one thing must always be true: we must in some way pay a price for using a supposition in part of a conclusion tree. In the portion of the tree below the scope of a supposition, it is no longer an assumption being made, and it is said to be *discharged*. In particular, is no longer in effect when we reach the conclusion of the rule. This means that a rule like *R* above cannot have a conclusion that simply incorporates the content of its two premises φ and ψ in the way Cnj does, for we have concluded ψ only with the help of a supposition that is no longer being made. Different rules will compensate for this in different ways, perhaps by some requirement about what is shown in the part of the tree above the other premise or by weakening the content of the conclusion. The rules for lemmas we will consider do the former, but the latter approach will be used quite often in the rules of later chapters. These concern compounds that have less content than one or more or their components, so the weakening used to compensate for a supposition is very natural.

The appearance of suppositions in argument trees and derivations follow these ideas in unsurprising ways. A segment of an argument tree that plans for the use of the rule *R* above would take the form shown on the right. The second of the two arguments to which the validity of the conclusion χ is reduced has the supposition as a premise in addition to the premises it shares with the argument above it.



And a derivation rule for *R* would look like this:



The supposition appears at the top of the scope line of the second child gap, and that scope line marks its scope in the derivation. In particular, it is not an available resource after that scope line ends, for the end of its scope line is the point at which it is discharged from our service. The sentence χ counts as a conclusion from φ and ψ, but its relation to them must reflect the fact that the gap above ψ might be filled using an assumption that is no longer being made.

Glen Helman 15 Jul 2012

2.4.2. The dangers of lemmas

A fully general rule for introducing lemmas was cited in 2.3.4 as an example of an unsafe rule because the resources of a gap might not entail the lemma even when the proximate argument of the gap is valid. Such a rule would also prevent a system from being decisive because it would always be possible to develop a gap further by introducing a lemma. However, as was noted earlier, more limited rules for introducing lemmas can be safe, and we will see later that they can also be progressive. In this section we will look at the problems posed by lemmas more closely before going on, in 2.4.3 and 2.4.4, to consider a couple of special cases where these problems do not arise.

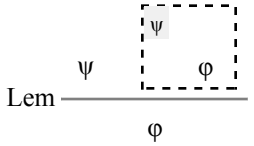
The law for lemmas of 1.4.7 can be stated as follows:

$$\Gamma \vDash \varphi \text{ if both } \Gamma \vDash \psi \text{ and } \Gamma, \psi \vDash \varphi$$

Let us recall why this is true: 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), then the one on the left will hold, too.

But this principle is stated only as an if claim, and the corresponding only if statement is not always true. When $\Gamma \vDash \varphi$, we know that $\Gamma, \psi \vDash \varphi$ by the monotonicity of $\vDash$; but, since φ and ψ need have no connection with one another, knowing that $\Gamma \vDash \varphi$ would by itself give us no reason to suppose that $\Gamma \vDash \psi$. Of course, in a case where we know that $\varphi \vDash \psi$, we would know $\Gamma \vDash \psi$ because of the chain law, and there are other cases where would know $\Gamma \vDash \psi$ because of special connections between Γ and ψ. We will use lemmas in special cases like these; but, before turning to them, let's look at what a fully general rule for lemmas would be like.

If used in conclusion trees a rule for lemmas would take the following form:

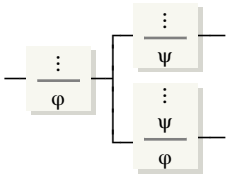


Before applying this rule to reach the conclusion φ, two things must be done. The first is to prove the lemma ψ, and the second is to prove φ, having available the lemma ψ in addition to the already available assumptions. The supposition ψ is offered for use as a branch tip only in the part of the tree above φ. Below φ, it has been discharged.

The value of using Lem rather than proving ψ on the right to begin with lies first in allowing us to consider the two components of the argument separately.

But we could also save some work in cases where the assumption ψ appears more than once in the tree on the right since we would not need put the proof of ψ above each of these. Both of these functions lie behind the use of lemmas in mathematical proofs. In the next section, we will see a simple example of the second sort of use.

The appearance of a corresponding step in an argument tree is shown below. Notice that the assumption ψ is dropped between the second of the arguments to which the rule is applies and the result of applying it.



That is, to prove φ we first try on the one hand to prove ψ and, on the other, to use the supposition that φ is true to prove φ. Here the price we pay for the use of ψ as a premise in the second argument on the right is need to reach it as a conclusion in the argument above it. And that is what makes lemmas dangerous. Although the validity of the two arguments on the right will insure the validity of the argument on the left. The fact that we can validly conclude φ from certain premises certainly does not insure that we can conclude a sentence ψ (which, so far as the statement of the rule goes, might be anything) from the same premises.

Finally, here is the derivation rule for lemmas.

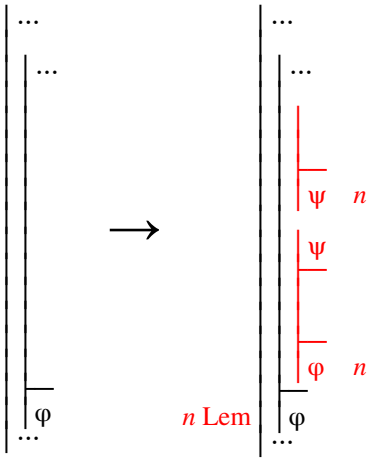


Fig. 2.4.2-1. Developing a derivation by introducing a lemma at stage *n* (a rule that will be part of our systems of derivations only in more restricted forms).

Remember that the assumption ψ is available only to the right of its scope line and that, after that scope line ends, it is said to have been discharged. The part of the proof in which this assumption is available corresponds to the right side of the tree in the conclusion trees and the lower branch in the argument trees we have been looking at.

Again, we gain something from proceeding in this way—but at a price. The second of the two new gaps in a derivation developed using Lem 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 properly chosen, that may also be easier than closing the original gap; but, because this rule is unsafe, we cannot be sure in general that the first new gap can be closed at all even if the original one could be closed eventually in other ways. Because of this, when lemmas are introduced in ordinary deductive reasoning we must be prepared to backtrack, to abandon the attempt to work by way of the lemma and look 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.

Indeed, we will not incorporate the rule Lem in the general form given here into our system of derivations. Instead, we will employ more specific rules that are based on the idea behind it. Even in cases where we can sure backtracking is not necessary, the introduction of lemmas could interfere with decisiveness if there were enough safe lemmas to keep introducing them forever. So our restrictions on the use of alternatives to Lem will be more severe than would be required merely to insure their safety.

Glen Helman 14 Jul 2012

2.4.3. Lemmas for *reductio* arguments

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

If $\varphi \vDash \psi$, then $\Gamma \vDash \varphi$ if and only if both $\Gamma \vDash \psi$ and $\Gamma, \psi \vDash \varphi$

which tells us that, when $\varphi \vDash \psi$, it is not only sound but also safe to introduce a lemma ψ in a derivation whose goal is φ. It is the only if part of this principle—and, more specifically, the part that says $\Gamma \vDash \varphi$ only if $\Gamma \vDash \psi$—that requires the assumption that $\varphi \vDash \psi$.

In order to apply the idea of this principle in derivations, we can look for convenient ways of insuring that $\varphi \vDash \psi$. The obvious valid arguments of this form among those we have identified so far are EFQ and the two forms of Ext. Although EFQ will prove to be the more important, Ext is a better source of examples at the moment and we will consider it first. Here is a derivation which uses the rule Lem to introduce a lemma that is the result of applying left extraction to the final goal.

	$A \wedge B$	1
1 Ext	A	(5),(9)
1 Ext	B	(4)
4 QED	● B	3
5 QED	● A	3
3 Cnj	$B \wedge A$	2
	$B \wedge A$	(7),(10)
7 QED	● $B \wedge A$	6
9 QED	● A	8
10 QED	● $B \wedge A$	8
8 Cnj	$A \wedge (B \wedge A)$	6
6 Cnj	$(B \wedge A) \wedge (A \wedge (B \wedge A))$	2
2 Lem	$(B \wedge A) \wedge (A \wedge (B \wedge A))$	

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 $B \wedge A$ by QED at stages 7 and 9 rather than repeating the proof used at stages 3-5 twice.

The basic idea here—isolating a component of an argument to avoid repeating it—is an important one. However, the actual simplification in this case is limited, and we would have few opportunities to use lemmas whose safety was assured by Ext. So we will not build this use of lemmas into our system of derivations, it will serve us only as an initial example.

The pattern *Ex Falso Quodlibet* provides the basis for a much more imporant use of lemmas. An argument whose conclusion is $\perp$ is often called a reductio *argument*; reductio here 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 $\perp$ 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:

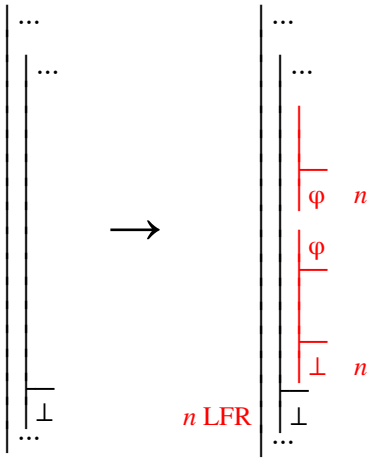


Fig. 2.4.3-1. Developing a derivation by introducing a lemma for a *reductio* at stage *n*.

The principle associated with this rule is the following:

$\Gamma \vDash \perp$ if and only if both $\Gamma \vDash \varphi$ and $\Gamma, \varphi \vDash \perp$

Although this follows from the principle stated at the beginning of the section and the validity of EFQ, it is instructive to look at its justification more directly. The if part again is just an instance of the law for lemmas. The only if part tells us that any counterexample lurking in one of the child gaps will also lurk in the parent. But this must be so, for any counterexample lurking in a child will make the active resources of the parent true (since they all remain active in each of the children) and because every interpretation makes $\perp$ false.

Unfortunately, we are not yet in a position to illustrate this rule because 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 $\perp$ either as a member or as a component of one. Such a set a can be shown to entail $\perp$ with use of nothing but Ext and QED, so introducing LFR would merely complicate that argument.

In the next chapter, the rule LFR will serve us as a temporary expedient, but we will eventually introduce other special rules that are designed to cover the case where LFR would be most useful, and LFR itself will be ignored. One reason is that the free use of LFR would undermine decisiveness since the form of the rule places no constraints on the number of different lemmas that might be introduced. Something like a limitation to sentences that already appear as components of active resources and goals would be sufficient to insure decisiveness and would still permit the more important uses of the rule, but we will not attempt to formulate the sort of restriction that would enable us to prove decisiveness for a system with LFR. It is simply not important enough to bother. Apart from its role as a temporary expedient, it will serve us mainly as a way of displaying the connection between the special rules to be introduced later and the idea of a lemma.

Glen Helman 11 Jul 2012

2.4.4. Attachment rules

The second sort of case in which use of a lemma is bound to safe is one in which it is clearly entailed by the available resources. A principle justifying that use would take the following form:

If $\Gamma \vDash \psi$, then $\Gamma \vDash \varphi$ if and only if $\Gamma, \psi \vDash \varphi$

If we were to drop the only if, this would be just a another way of stating the law for lemmas, and we know that $\Gamma \vDash \varphi$ only if $\Gamma, \psi \vDash \varphi$ by monotonicity.

There might seem to be little point in regarding this as a case of a lemma at all. If a statement is entailed by the resources, why not just have a rule that allows us to add it to our active resources? Indeed, as far as soundness and safety are concerned, this would come to the same thing. But this second use of lemmas is motivated (as well as constrained) by considerations of decisiveness. That is, our system is decisive in part because active resources are added only to reduce the complexity of proximate arguments, and it may be useful to reach our goal by way of a lemma that is entailed by our resources but is more complex than they are. Adding such a sentence as an active resource would open the door to going around in circles in which we add complexity only to simplify and then add complexity again, and so on.

When discussing the soundness of QED in 2.3.4, we saw that it would be legitimate for a rule to close a gap when its goal is not among its active resources—or even among the active resources of its ancestor gaps—if the goal was entailed by available resources. We will not use such a sweeping rule but we will introduce a few rules in special cases that add to the available resources of a gap without changing either its active resources or its goal.

An example is the following way of developing a gap, which we will call *Adjunction*:

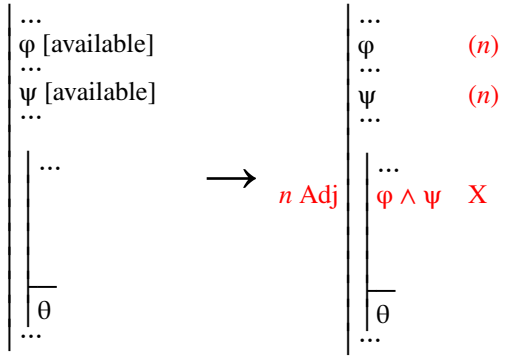


Fig. 2.4.4-1. Developing a derivation by applying Adj at stage *n*.

The added conjunction functions as a lemma, so this rule represents a way of using lemmas. However, it has a number of special features, both by comparison with a rule like LFR and by comparison with other rules we have seen.

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, for two reasons. First, we have not branched the gap so the added resource is available throughout the gap. And, second, we do not need to mark this new resource off as an added assumption because it is entailed by those already present.

Notice also that we treat this rule not as a way to plan for our goal but simply as a way to add resources. However, it does not exploit resources in order to add others and 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 once been an active resource that has since been exploited (and it would already be part of the available resources if it had been), it has a status similar to such resources.

One example of the use of Adj is provided by the example in 2.4.3 (though we are thinking of the lemma differently now: there we thought of it as something entailed by the goal while here we think of it as something entailed by the resources).

	A ∧ B	1
1 Ext	A	(2), (6)
1 Ext	B	(2)
2 Adj	B ∧ A	X, (4), (7)
4 QED	● B ∧ A	3
6 QED	● A	5
7 QED	● B ∧ A	5
5 Cnj	A ∧ (B ∧ A)	3
3 Cnj	(B ∧ A) ∧ (A ∧ (B ∧ A))	

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

This example also exhibits the sort of foresight or insight that is required to Adj and similar rules. At stage 2, after simplifying our resources as far as possible, we look ahead to the analysis of the goal (before we have built that analysis into our derivation at stages 3 and 5), and we see that we will need to establish B ∧ A twice to reach it. Noticing that we already have the makings of this sentence among our resources, we then assemble it using Adj to have it available for later use.

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.

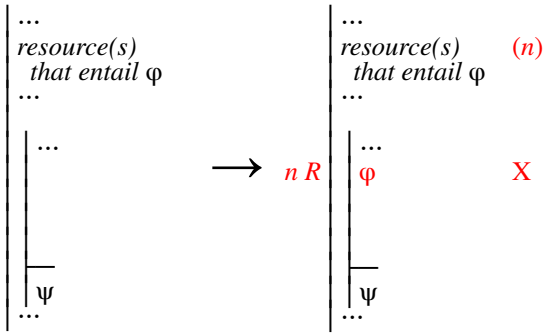


Fig. 2.4.4-2. Developing a derivation by applying an attachment rule *R* at stage *n*.

Since the lemma φ is not an active resource, the proximate argument of child gap is the same as the parent's proximate argument. This means that safety and soundness (even strictness) hold as they would for a gap that is completely unchanged. A rule like this must be considered when arguing for the soundness of rules like QED that use merely available resources, but the required argument was already considered in 2.3.4: an counterexample that lurks in a gap and all its ancestors will already make true not only all the available resources but also any sentence φ that is entailed by them, so we do nothing to change the situation by adding such a sentence φ to the available resources.

Of course, a rule like Adj does raise 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. We will not state the sort restriction on the use of attachment rules that would enable us to prove decisiveness; and, for practical purposes, the most valuable constraint on their use is simple good sense. But, as a rule of thumb, it is natural to limit the use of such rules to cases where the lemma is a component of a goal or active resource since such cases will represent the principal grounds for using these rules in any case. But it is important to remember that a sentence is a component of itself, and one common use of these rules

will be to introduce the goal itself as an available resource in order to apply QED.

Glen Helman 11 Jul 2012

2.4.s. Summary

1 Using a lemma is one way of dividing up the work of a proof. We might use lemmas in derivations by dividing a gap into two gaps, one with the lemma as a goal and the other with it as a further assumption to use in reaching the original goal. A rule Lemma (Lem) that does this is not safe in general, and we will use only special instances of it.

2 A lemma is always safe when it is entailed by the current goal. We can use this idea in *reductio* arguments, arguments whose goal is $\bot$. Since $\bot$ entails any sentence, the rule that introduces lemmas in such circumstances, Lemma for *Reductio* (LFR), will be safe (though some restriction on its use is needed to insure it is progressive).

3 A lemma is also safe if we know we can establish it. We will use this sort of lemma only in attachment rules, 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, 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*		*Basic system*
logical form	*as a resource*	*as a goal*	*when to close*	*rule*	
conjunction $\varphi \wedge \psi$	Ext	Cnj	the goal is also a resource	QED	
			$\top$ is the goal	ENV	
			$\bot$ is a resource	EFQ	
			Attachment rule		*Added rules (optional)*
			added resource	*rule*	
			$\varphi \wedge \psi$	Adj	
			Rule for lemmas		
			prerequisite	*rule*	
			the goal is $\bot$	LFR	

Glen Helman 11 Jul 2012

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 and exercises **2** and **4** of 2.3.x). They will work best as exercises in the use of Adj if you avoid using Cnj.

1. $A \vDash A \wedge A$

2. $A \wedge B, B \wedge C, C \wedge D \vDash A \wedge D$

3. $A \wedge B \vDash A \wedge (B \wedge A)$

4. $A, B \wedge C, D \vDash (C \wedge (B \wedge A)) \wedge B$

The exercise machine doesn't incorporate attachment rules, so, while it can generate exercises where Adj would be useful, that rule won't be used in any answers it produces.

Glen Helman 11 Jul 2012

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 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	$(C \wedge (B \wedge A)) \wedge B$	

Glen Helman 11 Jul 2012