3.5. Being guided by the rules

3.5.0. Overview

Derivations are now more varied in form and sometimes more complex than in the last chapter, but simple knowledge of when rules may be applied is enough to guide their development

3.5.1. Approaching derivations
: Each rule can be applied independently of the others, and each choice of a rule to apply turns on a simple description of the circumstances in which it is applied.

3.5.2. An example
: An extended example illustrates the sort of thinking that guides the development of a derivation.

3.5.3. A procedure
: This sort of thinking can be summarized as a procedure for developing derivations.

Glen Helman 28 Aug 2008

3.5.1. Approaching derivations

The general rule for starting and continuing derivations is to do anything the rules permit you to do. That is, any rule that can be applied to a goal or active resource of any gap is a legitimate way of proceeding. Some choices may lead to longer derivations than others; but the safety of the rules insures that you can never go off in the wrong direction, and their progressiveness insures that you will always move some distance toward the end. The differences in length that result from different choices of rules are likely to be greatest in valid arguments. When arguments are not valid, you may well have to apply all possible rules. The differences between derivations will then result only from the order in which the rules are applied (though such differences can make for significant differences in length).

The basic rules we have accumulated are shown in the following tables. The one on the left shows the exploitation rules for resources and the planning rules for goals. The simplest way of approaching derivations is to apply these rules as often as possible using the rules from the right-hand table to close gaps whenvever possible.

Rules for developing gaps		
	for resources	*for goals*
conjunction $\varphi \wedge \psi$	Ext	Cnj
negation $\neg \varphi$	CR (if φ is not atomic and the goal is $\bot$)	RAA
atomic sentence		IP

Rules for closing gaps	
when to close	*rule*
the goal is also a resource	QED
sentences φ and $\neg\varphi$ are resources & the goal is $\bot$	Nc
$\top$ is the goal	ENV
$\bot$ is a resource	EFQ

The rules LFR and Adj are not shown. They can be used to simplify derivations in some cases but they are never needed; and, when a gap will not close, they may simply delay the inevitable dead end. For this reason, the rules in the tables are labeled *basic rules* and are counted as part of the *basic system of derivations*.

Glen Helman 28 Aug 2008

3.5.2. An example

As an example of the use of the basic system, let us look at the promised further derivation for the argument of 3.2.x **2d**. The possible ways of proceeding at each stage are described in the commentary at the left.

Stage 1. We have two premises and a goal and we look to any of them for our starting point. But our premises are negations and can be exploited only in a *reductio* argument—that is, only when the goal is ⊥. So we must begin by planning for the goal, and RAA is the rule for doing that.

```
       | ¬ (A ∧ B)
       | ¬ (C ∧ ¬ B)
       |---
       |
       |
       |---
       | ¬ (A ∧ C)
```

Stage 2. After applying RAA, the goal is ⊥. There is no rule to plan for such a goal; but we have three resources, and we are now in a position to exploit any one of them. The rule Ext for exploiting conjunctions is easy, and it sometimes leads to a shorter derivation to do that as soon as possible, so that is what we will do. But there would be nothing wrong with exploiting either of the premises with CR; we will eventually need to do that in any case.

```
       | ¬ (A ∧ B)
       | ¬ (C ∧ ¬ B)
       |---
       | | A ∧ C
       | |---
       | |
       | |---
       | | ⊥              1
1 RAA  |---
       | ¬ (A ∧ C)
```

Stage 3. The use of Ext has given us two new active resources in addition to the two premises, and our goal is still ⊥. The two added resources are atomic sentences and can never be exploited, so we must now exploit one of the premises by CR. Either one will do, but let us choose the first.

```
       | ¬ (A ∧ B)
       | ¬ (C ∧ ¬ B)
       |---
       | | A ∧ C          2
       | |---
2 Ext  | | A
2 Ext  | | C
       | |
       | |---
       | | ⊥              1
1 RAA  |---
       | ¬ (A ∧ C)
```

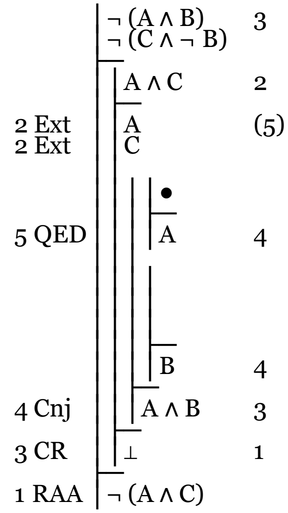
Stage 4. This use of CR has set our goal as the conjunction A ∧ B, and we can plan to get that by Cnj. Indeed, that's all we can do because we cannot exploit the second premise until our goal is again ⊥.

```
       | ¬ (A ∧ B)          3
       | ¬ (C ∧ ¬ B)
       |---
       | | A ∧ C            2
       | |---
2 Ext  | | A
2 Ext  | | C
       | |
       | | |
       | | |---
       | | | A ∧ B          3
       | |---
3 CR   | | ⊥                1
1 RAA  |---
       | ¬ (A ∧ C)
```

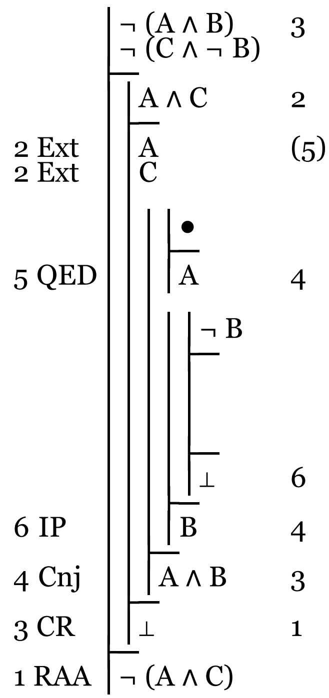
Stage 5. The use of Cnj has divided the gap into two open gaps, and we could go on to work on either of them. The goal of the first is also one of its resources, so we can close it immediately by QED and that's what we will do. But it would be fine to leave it open while we developed the second gap. It would even be possible to develop the first gap by planning for its goal with IP. While, of course, that would make for a longer derivation, we would eventually run out of things to do and would be forced to notice that the gap could be closed. (It would close on different grounds but, because the rules are safe and sufficient, there would be some reason for closing it.)

```
       | ¬ (A ∧ B)              3
       | ¬ (C ∧ ¬ B)
       |---
       | | A ∧ C                2
       | |---
2 Ext  | | A
2 Ext  | | C
       | |
       | | | |
       | | | |---
       | | | | A              4
       | | |
       | | | |---
       | | | | B              4
       | | |---
4 Cnj  | | | A ∧ B            3
3 CR   | | ⊥                    1
1 RAA  |---
       | ¬ (A ∧ C)
```

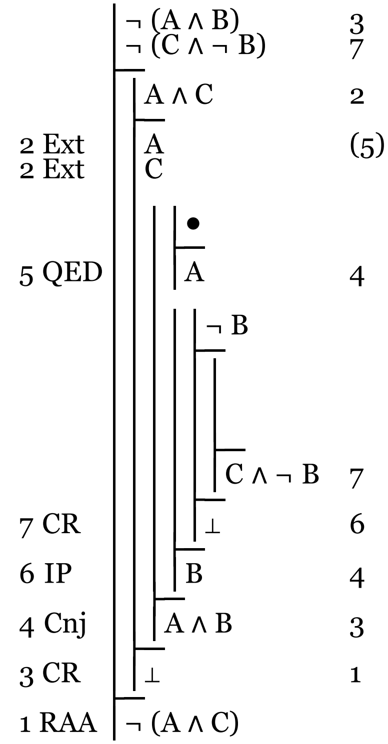
Stage 6. Now that the first gap is closed by RAA, we have only the second to work on. And, since the goal of this gap is not ⊥, we cannot exploit the second premise. Moreover, our other two resources are atomic sentences. So we must plan for the atomic goal, and the rule for doing that is IP.



Stage 7. The use of IP has made our goal ⊥ again, so we are forced to turn to our resources for guidance. We have added one, ¬ B; but it is the negation of an atomic sentence so, like A and C, it will never be exploited. But, since we are again working on a *reductio* argument, we can now exploit the second premise by CR.



Stage 8. This exploitation of the of the second premise by CR has left us with active resources that are all atomic sentences or negated atomic sentences. They can never be exploited, but our goal is a conjunction so we can plan to derive it by Cnj.



Stages 9-10. Cnj has divided the gap in two, but each of these two open gaps can be closed by QED, and we will go on to do that at the next two stages. Each gap also has a goal that we might plan for; and, as noted earlier, there would be nothing wrong with doing that. In fact, doing it in this case would not lead to a much longer derivation since, once we planned for the goals of these gaps, there would be nothing more we could do with either gap except close it.

	¬ (A ∧ B)	3
	¬ (C ∧ ¬ B)	7
	A ∧ C	2
2 Ext	A	(5)
2 Ext	C	
	•	
5 QED	A	4
	¬ B	
	C	8
	¬ B	8
8 Cnj	C ∧ ¬ B	7
7 CR	⊥	6
6 IP	B	4
4 Cnj	A ∧ B	3
3 CR	⊥	1
1 RAA	¬ (A ∧ C)	

The complete derivation is shown below. You can replay its development by moving the cursor across the series of numbers above it.

0 1 2 3 4 5 6 7 8 9 10

	¬ (A ∧ B)	3
	¬ (C ∧ ¬ B)	7
	A ∧ C	2
2 Ext	A	(5)
2 Ext	C	(9)
	•	
5 QED	A	4
	¬ B	(10)
	•	
9 QED	C	8
	•	
10 QED	¬ B	8
8 Cnj	C ∧ ¬ B	7
7 CR	⊥	6
6 IP	B	4
4 Cnj	A ∧ B	3
3 CR	⊥	1
1 RAA	¬ (A ∧ C)	

Glen Helman 28 Aug 2008

3.5.3. A procedure

The common features of the thinking used at each stage in the development of this derivation can be captured in a procedure that can be applied repeatedly to guide the development of any derivation. Restarting the procedure introduces a new stage in the derivation, and the procedure will be restarted one final time after the last stage in order to confirm that the derivation has reached its end.

1. *Choose an open gap*. Find all the open gaps of the derivation. If there are none, the derivation is closed and you are done. If there are more than one, pick one to work on (it does not matter which).
2. *Identify its proximate argument*. Find the goal and the active resources of the gap you are working on; and, for each of these, identify the kind of sentence it is—that is, decide whether it is $\top$, $\bot$, a conjunction, a negation, or an atomic sentence.
3. *Check for closure*. Check whether the gap can be closed using one of the rules in the following table:

rule	*conditions for applying it*
QED	the goal is among the resources
Nc	the goal is $\bot$, and there are sentences φ and $\neg\varphi$ among the resources
ENV	the goal is $\top$
EFQ	$\bot$ is a resource

   If the conditions for applying one of these rules are met, apply the rule, and start again at step 1.
4. *Choose a sentence to work on*. Find which, if any, of the goal and active resources, has a rule that may be applied at this stage. That is, for each of these sentences check whether a basic rule (outlined in the table below) applies to a resource or goal of that sort and check whether any additional requirements are met.

kind of sentence		*rule for this sentence* *as a resource*	*as a goal*
conjunction		Ext	Cnj
negated	non-atomic sent.	CR (when the goal is $\bot$)	RAA
	atomic sentence	*none*	
atomic sentence		*none*	IP
$\top$ or $\bot$		*none*	*none*

   If there is no sentence to which a rule can be applied, you have reached a dead-end open gap; mark it as such and you are done. If there is more than one sentence to which a rule may be applied, pick one to work on (it does not matter which).
5. *Apply a rule*. Apply the rule you have identified to the sentence you have found, and start again at step 1.

The choice of a gap or sentence to work on does not matter in the sense that whether a gap eventually closes or reaches a dead-end does not depend on the way such a choice is made. Of course, such a choice can make a difference for the length of the derivation; but the difference will often amount to only one stage or a line or two.

This procedure describes a way of applying the rules; and, even though it allows some choice, it is more restrictive than the rules alone. For example, it forces you to close a gap you are working one if that is possible even if the rules would also allow you to develop the gap further. In that case, it simply enforces good sense, but it is also restrictive in one way that can length a derivation: no allowance is made for the option of exploiting a resource in more than one gap at once (i.e., in the same stage of development). Consequently, you should regard this procedure as merely a rough guide that may be supplemented by shortcuts such as this one when you see that they are possible. Other such shortcuts are the use of available but inactive resources with rules like QED and the use of non-basic rules like Adj and LFR.

Glen Helman 28 Aug 2008

3.5.s. Summary

1 Any step in a derivation that is allowed by the basic rules (that is, for now, all rules except LFR and Adj) is safe and will take the derivation some way towards completion. We call the system of derivations limited to those rules the basic system. There will often be different orders in which the basic rules can be applied, and such differences may lead to longer or shorter derivations. The use of non-basic rules can sometimes shorten derivations still further, but they may not bring a derivation any closer to is final state.

2 Although insight or foresight can help to shorten a derivation, all that is needed to complete a derivation is an understanding of what rules may be applied at any given stage. This is illustrated in the commentary on an extended example.

3 Derivations can be approached systematically through a 5-step procedure that is applied repeatedly until all gaps close or the derviation reaches a dead end.

The following table collects all rules we have now seen (and, as with the table of 2.4.s, the rule labels are links to the original statements of the rules):

Rules for developing gaps		
	for resources	*for goals*
atomic sentence		IP
negation $\neg\,\varphi$	CR (if φ is not atomic and the goal is $\bot$)	RAA
conjunction $\varphi \wedge \psi$	Ext	Cnj

Rules for closing gaps		
when to close	*rule*	
the goal is also a resource	QED	*Basic system*
sentences φ and $\neg\,\varphi$ are resources & the goal is $\bot$	Nc	
$\top$ is the goal	ENV	
$\bot$ is a resource	EFQ	

Attachment rule		
added resource	*rule*	*Added rules (optional)*
$\varphi \wedge \psi$	Adj	
Rule for lemmas		
prerequisite	*rule*	
the goal is $\bot$	LFR	

Glen Helman 28 Aug 2008

3.5.x. Exercise questions

1. For each of the claims of entailment shown below, construct a derivation using only the basic rules and annotate it to show explicitly how it is the result of following the procedure given in 3.5.3. Provide one note for each pass through the procedure—i.e., one note for each stage followed by one for the final pass through the procedure that confirms that the derivation is done. Each note should indicate (i) the open gap chosen (or the fact that all gaps are closed), (ii) the proximate argument of this gap and either the rule (or rules) by which it may be closed or the rule (or rules) that may be applied to develop it, and (iii) whether the gap is closed, developed, or marked as a dead end (together with the rule used if there was a choice).

a. $\neg\, A \Rightarrow \neg\, (B \wedge A)$
b. $A \wedge B \Rightarrow B \wedge A$
c. $B \Rightarrow B \wedge A$
d. $\neg\, (A \wedge B), A \Rightarrow \neg\, B$
e. $\neg\, (A \wedge B), \neg\, (B \wedge C) \Rightarrow \neg\, B$

2. More than one derivation using the basic rules can be constructed for each of the claims of entailment below. In each case construct two and also recognize any further possibilities by noting each stage at which there was a choice between different ways of developing the derivation.

a. $A \wedge B \Rightarrow B \wedge A$
b. $\neg\, (A \wedge B), B \wedge C \Rightarrow \neg\, A$
c. $\neg\, (A \wedge B), \neg\, (B \wedge C) \Rightarrow \neg\, B$

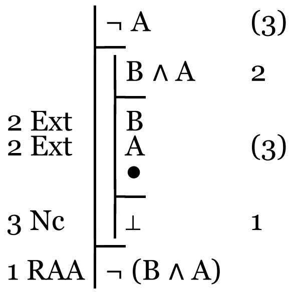
The exercise machine does not generate exercises of this sort; but, of course, you may use it to generate the derivations that are described in the answers.

Glen Helman 28 Aug 2008

3.5.xa. Exercise answers

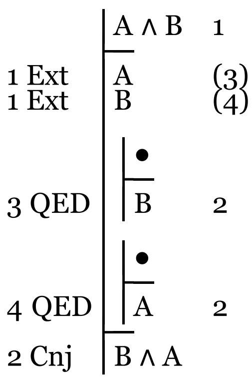
1. The rules that may be applied are indicated in the annotations for these derivations by bracketed subscripts on elements of the proximate argument, on the sentence to which the rule is applied in the case of development rules and on the slash between resources and goal in the case of closure rules.

a.



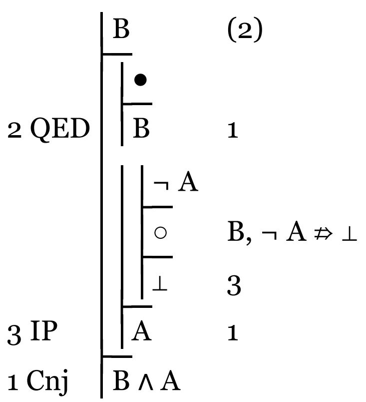
1. One open gap. Prox. arg.: ¬ A / ¬ (B ∧ A)$_{[RAA]}$. Developled.
2. One open gap. Prox. arg.: ¬ A, B ∧ A$_{[Ext]}$ / ⊥. Developed.
3. One open gap. Prox. arg.: ¬ A, B, A /$_{[Nc]}$ ⊥. Closed gap.
4. No gaps open.

b.



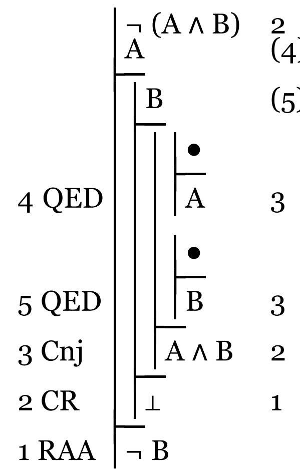
1. One open gap. Prox. arg.: A ∧ B$_{[Ext]}$ / B ∧ A$_{[Cnj]}$. Developed by Ext.
2. One open gap. Prox. arg.: A, B / B ∧ A$_{[Cnj]}$. Developed.
3. Chose first open gap. Prox. arg.: A, B /$_{[QED]}$ A. Closed gap.
4. One open gap. Prox. arg.: A, B /$_{[QED]}$ B. Closed gap.
5. No gaps open.

c.



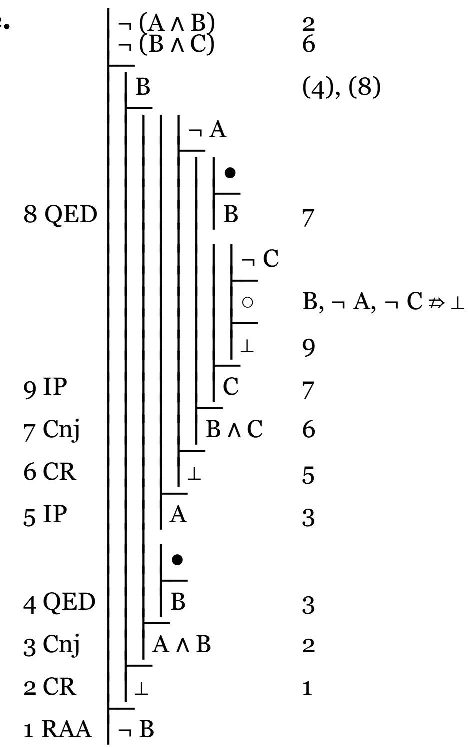
1. One open gap. Prox. arg.: B / B ∧ A$_{[Cnj]}$. Developed.
2. Chose first open gap. Prox. arg.: B /$_{[QED]}$ B. Closed gap.
3. One open gap. Prox. arg.: B / A$_{[IP]}$. Developed.
4. One open gap. Prox. arg.: B, ¬ A / ⊥. Marked dead end.

d.



1. One open gap. Prox. arg.: ¬ (A ∧ B), A / ¬ B$_{[RAA]}$. Developed.
2. One open gap. Prox. arg.: ¬ (A ∧ B)$_{[CR]}$, A, B / ⊥. Developed.
3. One open gap. Prox. arg.: A, B / A ∧ B$_{[Cnj]}$. Developed.
4. Chose first open gap. Prox. arg.: A, B /$_{[QED]}$ A. Closed gap.
5. One open gap. Prox. arg.: A, B /$_{[QED]}$ B. Closed gap.
6. No gaps open.

e.



1. One open gap. Prox. arg.: ¬ (A ∧ B), ¬ (B ∧ C) / ¬ B$_{[RAA]}$. Developed.
2. One open gap. Prox. arg.: ¬ (A ∧ B)$_{[CR]}$, ¬ (B ∧ C)$_{[CR]}$, B / ⊥. Developed by first CR.
3. One open gap. Prox. arg.: ¬ (B ∧ C), B / A ∧ B$_{[Cnj]}$. Developed.
4. Chose second open gap. Prox. arg.: ¬ (B ∧ C), B /$_{[QED]}$ B. Closed gap.
5. One open gap. Prox. arg.: ¬ (B ∧ C), B / A$_{[IP]}$. Developed.
6. One open gap. Prox. arg.: ¬ (B ∧ C)$_{[CR]}$, B, ¬ A / ⊥. Developed.
7. One open gap. Prox. arg.: B, ¬ A / B ∧ C$_{[Cnj]}$. Developed.
8. Chose first open gap. Prox. arg.: B, ¬ A /$_{[QED]}$ B. Closed gap.
9. One open gap. Prox. arg.: B, ¬ A / C$_{[IP]}$. Developed.
10. One open gap. Prox. arg.: B, ¬ A, ¬ C / ⊥. Marked dead end.

2. The stages at which choices are made are indicated by references to notes below that describe the choices that were made.

a.

	A ∧ B	1*
1* Ext	A	(3†)
1* Ext	B	(4)
	●	
3† QED	B	2
	●	
4 QED	A	2
2 Cnj	B ∧ A	

* Chose Ext instead of Cnj
† Chose first of 2 gaps

	A ∧ B	2†, 5
2† Ext	A	
2† Ext	B	(3‡)
	●	
3‡ QED	B	1*
	¬ A	(6)
5 Ext	A	(6)
5 Ext	B	
	●	
6 Nc	⊥	4§
4§ IP	A	1*
1* Cnj	B ∧ A	

* Chose Cnj instead of Ext
† Chose first of 2 gaps and Ext instead of IP
‡ Chose first of 2 gaps
§ Chose IP instead of Ext

b.

	¬ (A ∧ B)	3
	B ∧ C	1*
1* Ext	B	(6)
1* Ext	C	
	A	(5†)
	●	
5† QED	A	4
	●	
6 QED	B	4
4 Cnj	A ∧ B	3
3 CR	⊥	2
2 RAA	¬ A	

* Chose Ext instead of RAA
† Chose first of 2 gaps

	¬ (A ∧ B)	2†
	B ∧ C	5**
	A	(4§)
	●	
4§ QED	A	3‡
5** Ext	B	(6)
5** Ext	C	
	●	
6 QED	B	3‡
3‡ Cnj	A ∧ B	2†
2† CR	⊥	1*
1* RAA	¬ A	

* Chose RAA instead of Ext
† Chose CR instead of Ext
‡ Chose Cnj instead of Ext
§ Chose first of 2 gaps
** Chose Ext instead of IP

c.

	¬ (A ∧ B)	2*
	¬ (B ∧ C)	6
	B	(4†), (8‡)
	¬ A	
	●	
8‡ QED	B	7
	¬ C	
	○	B, ¬ A, ¬ C ⇏ ⊥
	⊥	9
9 IP	C	7
7 Cnj	B ∧ C	6
6 CR	⊥	5
5 IP	A	3
	●	
4† QED	B	3
3 Cnj	A ∧ B	2*
2* CR	⊥	1
1 RAA	¬ B	

* Chose CR on 1^{st} premise instead of 2^{nd}
† Chose second of 2 gaps
‡ Chose first of 2 gaps

	¬ (A ∧ B)	5‡
	¬ (B ∧ C)	2*
	B	
	B	3
	¬ C	
	B	6**
	¬ C	
	○	B, ¬ A, ¬ C ⇏ ⊥
	⊥	7††
7†† IP	C	6**
6** Cnj	A ∧ B	5‡
5‡ CR	⊥	4†
4† IP	C	3
3 Cnj	B ∧ C	2*
2* CR	⊥	1
1 RAA	¬ B	

* Chose CR on 2^{nd} premise instead of 1^{st}
† Chose second of 2 gaps
‡ Chose second of 2 gaps
§ Chose second of 2 gaps
** Chose second of 2 gaps
†† Chose third of 3 gaps
(Notice that two gaps remain incomplete at the end. They would close if attention were turned to them, but the procedure ends work on a derivation as soon as any open gap has reached a dead end.)

Glen Helman 28 Aug 2008