3.3.4. 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 $\phi \wedge \psi$	Ext	Cnj
negation $\neg \phi$	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 $\neg\phi$ 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***.

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. The goal is now ⊥ and 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. We now have four active resources, the two premises and the two sentences we extracted at stage 2, 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 we will choose the first.

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

Stage 4. Our goal is now the conjunction A ∧ B so we could plan to get that by Cnj. And 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. We now have 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. Since the first gap is now closed, we can only work on the second. 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.

	│ ¬ (A ∧ B)	3
	│ ¬ (C ∧ ¬ B)	
	├──	
	│ │ A ∧ C	2
	│ ├──	
2 Ext	│ │ A	(5)
2 Ext	│ │ C	
	│ │ │ │ •	
	│ │ │ ├──	
5 QED	│ │ │ │ A	4
	│ │ │ │	
	│ │ │ ├──	
	│ │ │ │ B	4
	│ │ ├──	
4 Cnj	│ │ │ A ∧ B	3
	│ ├──	
3 CR	│ │ ⊥	1
	├──	
1 RAA	│ ¬ (A ∧ C)	

Stage 7. Our goal is now ⊥ 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.

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

Stage 8. Our active resources are all atomic sentences or negated atomic sentences so they can never be exploited, but our goal is a conjunction so we can plan to derive it by Cnj.

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

Stages 9-10. Each of the two open gaps we have now 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. And 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 one 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 18 Sep 2004