3.4. Counterexamples to *reductio*s

3.4.0. Overview

All derivations that fail will now end in the failure of a *reductio*, and this produces some small changes in what we say about the failure of derivations.

Glen Helman 01 Aug 2011

3.4.1. When *reductio*s fail

The system of derivations for negation can be shown to be adequate by establishing the three properties of sufficiency, conservativeness, and decisiveness discussed in 2.3.

To say that a system is conservative is to say that all its rules are sound and safe. Soundness and safety say more than do the basic laws of negation; but, as was the case with conjunction, the natural way of establishing the basic laws for negation is enough to establish soundness and safety. The key to the argument in the laws for negation is the fact that, when it comes to dividing a gap, having a given sentence (φ or ¬ φ) as a resource comes to the same thing as having a contradictory sentence (¬ φ or φ, respectively) as a goal. This idea can be used to show each of the rules RAA, IP, and CR is both sound (in fact, strict) and safe, for it shows that the same interpretations divide the proximate arguments of gaps to which these rules apply and the child gaps that result from applying them. Since the rule Nc closes a gap, safety is not an issue; and, since we allow available but inactive resources to be used, we cannot expect to show more than strictness. But its soundness is clear: if the available resources include both φ and ¬ φ, no interpretation can make them all true, and a sound rule needs to insure some child gap is open only if the parent is divided by an interpretation that makes true all the available resources.

However, there is more to be said in the case of the properties of sufficiency and decisiveness. A system is sufficient if it has enough rules to close any dead-end gaps that cannot be divided. Given the rules we have now, a dead-end open gap must have ⊥ as its goal (since otherwise we could develop the gap with Cnj, RAA, or IP or close it with ENV), it cannot have a conjunction or a negated non-atomic sentence as a resource (since otherwise we could develop the gap with Ext or CR), it cannot have ⊥ among its resources (since otherwise we could close the gap using either QED or EFQ), and it cannot have both a sentence and its negation among its resources (since otherwise we could close the gap with Nc). So the proximate argument of a dead-end gap must be a *reductio* whose premises are limited to ⊥, atomic, and negated atomic sentences, with no sentence appearing both negated and unnegated among the premises. To show sufficiency, we must show that we can always divide such an argument. And we can do this by making an atomic sentence true when it appears among the premises and false when its negation appears. We can assign truth values in this way since no sentence appears both negated and unnegated, and an assignment like this will make all premises true and it will, of course, make the conclusion ⊥ false.

This argument for sufficiency tells us what we need to do in order to present a counterexample on the basis of a dead-end open gap. Here is an example of that.

```
         | ¬ (A ∧ ¬ (B ∧ C))   2
         | ¬ B                 (7)
         |---
         | | ¬ A
         | |---
         | | | | ¬ A
         | | | | ---
         | | | | ○              ¬ A, ¬ B ⊭ ⊥
         | | | | ---
         | | | | ⊥             4
         | | | ---
4 IP     | | | A               3
         | | |
         | | | | B ∧ C         6
         | | | | ---
6 Ext    | | | | B             (7)
6 Ext    | | | | C
         | | | | ●
         | | | | ---
7 Nc     | | | | ⊥           5
         | | | ---
5 RAA    | | | ¬ (B ∧ C)       3
         | | ---
3 Cnj    | | A ∧ ¬ (B ∧ C)     2
         | ---
2 CR     | | ⊥                 1
         | ---
1 IP     | A
```

A	B	C	¬ (A ∧ ¬ (B ∧ C)) , ¬ B / A
F	F	T	Ⓣ F T F Ⓣ Ⓕ

(Although this derivation has been continued as far as possible, it could have been ended after the dead-end gap appeared at stage 4.)

The proximate argument of the dead-end gap is ¬ A, ¬ B / ⊥. To divide this, we must make A and B false since their negations are active resources of the dead-end gap. The value assigned to C does not matter since neither it nor ¬ C appears among the premises of this argument. So, although C is assigned **T** in the counterexample presented above, an interpretation that made each of A, B, and C false would also be a counterexample.

The basic issues regarding decisiveness were touched on when the rule IP was introduced in 3.3.1, but they deserve to be considered a little more fully. The system of derivations for conjunction is easily seen to be decisive because we cannot go on forever dropping and shortening sentences among the resources and goals. But we now have rules that can do things other than simplifying the resources and goals. In particular, we can add resources while dropping goals and vice versa, and, in the case of IP, we can do this by adding a resource that has one more connective than the goal that was dropped. The cases where we use IP and CR have been restricted so that we cannot go in circles, but an argument is needed to show that those restrictions are enough.

Decisiveness will follow if all our rules are progressive in the sense of bringing us closer to a dead end in a way that cannot be continued indefinitely. In judging this, we cannot now look only at the number of connectives in sentences. In the first place, atomic sentences have no connectives, but are a sign that a derivation has not reached its end when they appear as goals. And, second, negated atomic sentences do contain connectives but can appear as resources in a dead-end gap. Let us say that the sort of sentences that may appear in a gap that has reached a dead end are *minimal*. Then a minimal resource will be ⊤ or an atomic or negated atomic sentence and a minimal goal must be ⊥. Thus whether a given sentence counts as minimal depends on whether it appears as a resource or a goal.

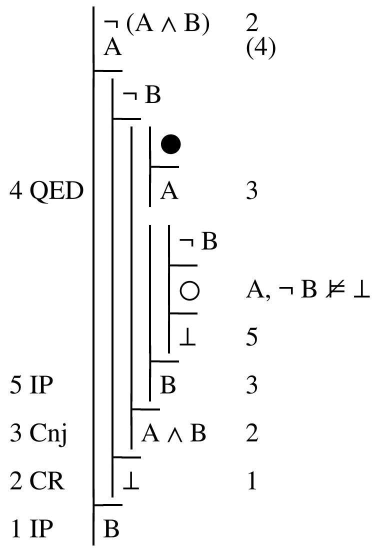
In order to measure distance from the end of a derivation, we will assign each resource and goal a *grade*. Minimal sentences form the lowest grade, and non-minimal sentences are graded above them and relative to one another by counting the connectives appearing in them. There are many ways of assigning numerical grades that would accomplish this. To be concrete, let us suppose we assign grade 0 to minimal sentences and then one more than the number of connectives to any other sentence. So atomic and negated atomic resources both have grade 0, but atomic and negated atomic goals have grades 1 and 2, respectively. As a goal, ⊥ has grade 0 while, as a resource, it has grade 1. (Notice also that, while ⊤ has grade 0 as a resource and grade 1 as a goal, its negation ¬ ⊤ has grade 2 whether it is a resource or a goal.)

Now, consider the whole group of active resources and goals of every open gap of a derivation. If we look at each of the rules for developing gaps, we see that the effect of applying any one of them will always be to eliminate an active resource or a goal. It may also add resources or goals, but any sentence that is added either has fewer connectives than the sentence dropped or, in the case of IP, is a minimal sentence when the sentence dropped was not minimal. Either way, additions will be sentences of a lower grade, so eventually all active sentences will be minimal and the process must end. Notice that if, for example, we allowed CR to apply to negated atomic sentences as well as negated non-atomic sentences, this rule would no longer be progressive since we could, for example, drop a minimal resource ¬ A and add the non-minimal goal A. However, when φ is not atomic, ¬ φ has a higher grade than φ because of the extra connective, so the restricted CR is progressive.

3.4.2. Some examples of consistency

The aim of this subsection is to consider a few examples, but its title makes a further general point. An interpretation that divides a dead-end open gap will divide a *reductio* argument and thus show that its premises can all be true together. That is, it will show that the active resources of a dead-end open gap form a consistent set. Counterexamples to arguments in chapter 2 did that, too, since they made all resources of the gap they divided true, but now that is the full significance of a counterexample since the goal of the gap it divides is ⊥ and is therefore automatically false.

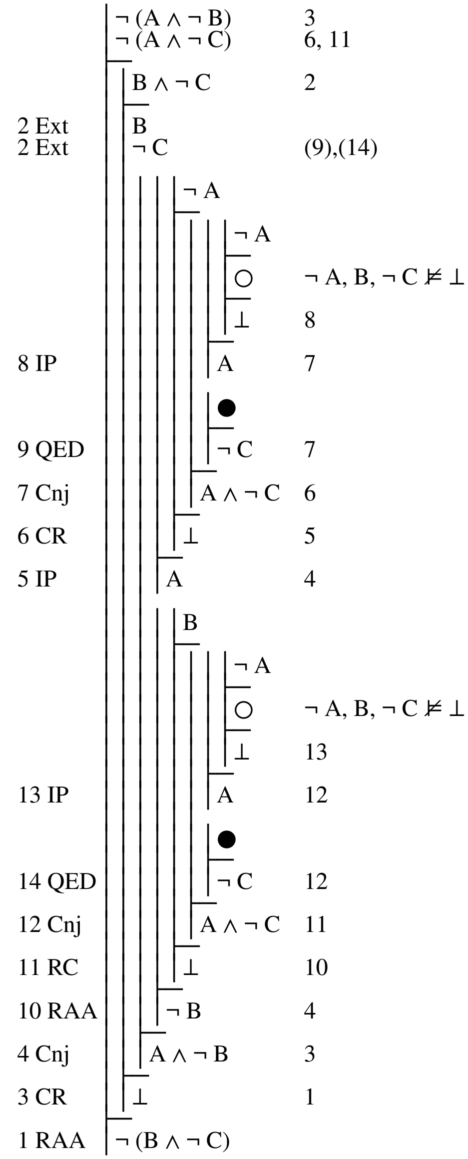
Here is a simple example that exhibits a common pattern.



A	B	¬ (A ∧ B)	, A	/ B
T	F	Ⓣ F	Ⓣ	Ⓕ

It may seem odd to continue to stage 5 since, before IP is applied, the resources of the second gap are fully exploited and its goal is not among them. So, in this case, it is clear before stage 5 that the gap will not close. But, with enough thought, it would have been clear before stage 1 that some gap would not close so the simple fact that a dead-end gap can be foreseen is not grounds for declaring one. A dead-end gap is an indication of failure made fully explicit. What we count as fully explicit is a conventional matter, and we will treat as fully explicit only what cannot be made more explicit by the system of derivations. In this case, that requires the final use of IP (though the closure of the first gap at stage 4 might have been ignored).

Here is a somewhat longer example. It is developed following the most straightforward approach, in which resources are exploited in the order in which they appear (when there is a choice).



A	B	C	¬ (A ∧ ¬ B)	, ¬ (A ∧ ¬ C)	/ ¬ (B ∧ ¬ C)
F	T	F	Ⓣ F F	Ⓣ F T	Ⓕ T T

The derivation could have been shortened significantly by reversing the order in which the first two resources were exploited, but it would have been shorter still (no matter what order these resources were exploited in) if we stopped after reaching a dead-end gap at stage 8. Stopping then would be perfectly legitimate, and the derivation is continued here only for the sake of the example. One reason for continuing a dervation after an open gap has been reached would be that we wanted, for some reason, to discover all the interpretations that might divide the ultimate argument. In fact, in this case, there is only one such interpretation, and both open gaps lead us to the same thing.

Glen Helman 10 Sep 2011

3.4.s. Summary

1 The adequacy of our current system is established by showing that it is sufficient, conservative, and decisive. The arguments for sufficiency and decisiveness take a slightly different form from those used in the last chapter. A gap that remains open at a dead end will now always have ⊥ as its goal and its resources are limited to ⊤, atomic sentences, and negated atomic sentences, with no resource being the negation of another. Any such gap can be divided by an interpretation that makes all its active resources true, so the rules are sufficient to close any gap that cannot be divided. Also, we can show that our new rules will not lead us on forever by showing that they are progressive by leading us always to replace goals or resources by others of a lower grade eventually leading us to goals and resources that are minimal, a class that includes ⊤, atomic sentences and negated atomic sentences in the case of resources and ⊥ alone in the case of goals.

2 Dead-end gaps will now have proximate arguments that are *reductio*s, so the failure of a derivation will turn on the failure of a *reductio* and thus on the fact that the premises of the *reductio* form a consistent set. Thus any example of the failure of entailment will henceforth be traced to the consistency of some set.

Glen Helman 01 Aug 2011

3.4.x. Exercise questions

1. The following arguments are not formally valid. In each case, use a derivation to show this and present a counterexample that the derivation leads you to.

- **a.** $\neg B \,/\, \neg (A \wedge \neg B)$
- **b.** $\neg (A \wedge B) \,/\, \neg A \wedge \neg B$
- **c.** $\neg (A \wedge B), \neg (B \wedge C) \,/\, \neg (A \wedge C)$

2. Use derivations to check the following claims of entailment. If the claim fails, present a counterexample that the derivation leads you to.

- **a.** $\neg (A \wedge \neg B) \vDash B$
- **b.** $\neg (A \wedge B) \vDash \neg (B \wedge A)$
- **c.** $\neg (A \wedge \neg B) \vDash \neg (B \wedge \neg A)$
- **d.** $\neg (A \wedge B), \neg (B \wedge C), B \vDash \neg A \wedge \neg C$
- **e.** $\neg (A \wedge \neg (B \wedge \neg (C \wedge \neg D))) \vDash \neg (A \wedge \neg (B \wedge D))$

For more exercises, use the exercise machine.

Glen Helman 01 Aug 2011

3.4.xa. Exercise answers

1. a.

	$\neg B$	
	$A \wedge \neg B$	2
2 Ext	A	
2 Ext	$\neg B$	
	○	$A, \neg B \nvDash \bot$
	$\bot$	1
1 RAA	$\neg (A \wedge \neg B)$	

A	B	¬ B	/	¬ (A ∧ ¬ B)
T	F	Ⓣ		Ⓕ T T

b.

	$\neg (A \wedge B)$	3,8
	A	(5)
	●	
5 QED	A	4
	$\neg B$	
	○	$A, \neg B \nvDash \bot$
	$\bot$	6
6 IP	B	4
4 Cnj	$A \wedge B$	3
3 CR	$\bot$	2
2 RAA	$\neg A$	1
	B	(11)
	$\neg A$	
	○	$\neg A, B \nvDash \bot$
	$\bot$	10
10 IP	A	9
	●	
11 QED	B	9
9 Cnj	$A \wedge B$	8
8 CR	$\bot$	7
7 RAA	$\neg B$	1
1 Cnj	$\neg A \wedge \neg B$	

A	B	¬ (A ∧ B)	/	¬ A ∧ ¬ B	
T	F	Ⓣ F		F Ⓕ T	(divides the first gap)
F	T	Ⓣ F		T Ⓕ F	(divides the second gap)

c.

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

A	B	C	¬ (A ∧ B) , ¬ (B ∧ C) / ¬ (A ∧ C)
T	F	T	Ⓣ F Ⓣ F Ⓕ T

2. a.

	¬ (A ∧ ¬ B)	2
	¬ B	(5)
	¬ A	
	○	¬ A, ¬ B ⊭ ⊥
	⊥	4
4 IP	A	3
	●	
5 QED	¬ B	3
3 Cnj	A ∧ ¬ B	2
2 CR	⊥	1
1 RAA	B	

A	B	¬ (A ∧ ¬ B) / B
F	F	Ⓣ F T Ⓕ

b.

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

c.

```
          | ¬ (A ∧ ¬ B)      3
          |----
          | | B ∧ ¬ A        2
          | |----
2 Ext     | | B
2 Ext     | | ¬ A
          | | | | | ¬ A
          | | | | |----
          | | | | | ○        ¬ A, B ⊭ ⊥
          | | | | |----
          | | | | | ⊥        5
          | | | |----
5 IP      | | | | A          4
          | | | | | B
          | | | | |----
          | | | | | ○        ¬ A, B ⊭ ⊥
          | | | | |----
          | | | | | ⊥        6
          | | | |----
6 IP      | | | | ¬ B        4
          | | |----
4 Cnj     | | | A ∧ ¬ B      3
          | |----
3 CR      | | ⊥              1
          |----
1 RAA     | ¬ (B ∧ ¬ A)
```

A	B	¬ (A ∧ ¬ B) / ¬ (B ∧ ¬ A)
F	T	Ⓣ F F Ⓕ T T

d.

```
          | ¬ (A ∧ B)                    3,8
          | ¬ (B ∧ C)                    12
          | B                            (6),(10),(14)
          |----
          | | | A                        (5)
          | | |----
          | | | | | ●
          | | | | |----
5 QED     | | | | | A                    4
          | | | | | ●
          | | | | |----
6 QED     | | | | | B                    4
          | | | |----
4 Cnj     | | | | A ∧ B                3
          | | |----
3 CR      | | | ⊥                        2
          | |----
2 RAA     | | ¬ A                      1
          | | | C                        (15)
          | | |----
          | | | | | | ¬ A
          | | | | | |----
          | | | | | | | | ●
          | | | | | | | |----
14 QED    | | | | | | | | B            13
          | | | | | | | | ●
          | | | | | | | |----
15 QED    | | | | | | | | C            13
          | | | | | | |----
13 Cnj    | | | | | | | B ∧ C        12
          | | | | | |----
12 CR     | | | | | | ⊥              11
          | | | | |----
11 IP     | | | | | A                  9
          | | | | | ●
          | | | | |----
10 QED    | | | | | B                  9
          | | | |----
9 Cnj     | | | | A ∧ B                8
          | | |----
8 CR      | | | ⊥                        7
          | |----
7 RAA     | | ¬ C                      1
          |----
1 Cnj     | ¬ A ∧ ¬ C
```

```
          | ¬ (A ∧ B)            3
          | ¬ (B ∧ C)            8
          | B                    (6),(10)
          |----
          | | | A                (5)
          | | |----
          | | | | | ●
          | | | | |----
5 QED     | | | | | A            4
          | | | | | ●
          | | | | |----
6 QED     | | | | | B            4
          | | | |----
4 Cnj     | | | | A ∧ B        3
          | | |----
3 CR      | | | ⊥                2
          | |----
2 RAA     | | ¬ A              1
          | | | C                (11)
          | | |----
          | | | | | ●
          | | | | |----
10 QED    | | | | | B            9
          | | | | | ●
          | | | | |----
11 QED    | | | | | C            9
          | | | |----
9 Cnj     | | | | B ∧ C        8
          | | |----
8 CR      | | | ⊥                7
          | |----
7 RAA     | | ¬ C              1
          |----
1 Cnj     | ¬ A ∧ ¬ C
```

The derivation at the left above exploits resources in their order of appearance; while the alternative derivation at the right chooses, at stage 8, the resource that is most closely connected with other resources of the gap in which it is exploited. Notice that derivation at the left is eventually led, at stage 12, to exploit the same resource to the same effect.

e.

	¬ (A ∧ ¬ (B ∧ ¬ (C ∧ ¬ D)))	3
	A ∧ ¬ (B ∧ D)	2
2 Ext	A	(5)
2 Ext	¬ (B ∧ D)	8
	●	
5 QED	A	4
	B ∧ ¬ (C ∧ ¬ D)	7
7 Ext	B	(10)
7 Ext	¬ (C ∧ ¬ D)	12
	●	
10 QED	B	9
	¬ D	(15)
	¬ C	
	○	A, B, ¬ C, ¬ D ⊭ ⊥
	⊥	14
14 IP	C	13
	●	
15 QED	¬ D	13
13	C ∧ ¬ D	12
12 CR	⊥	11
11 IP	D	9
9 Cnj	B ∧ D	8
8 CR	⊥	6
6 RAA	¬ (B ∧ ¬ (C ∧ ¬ D))	4
4 Cnj	A ∧ ¬ (B ∧ ¬ (C ∧ ¬ D))	3
3 CR	⊥	1
1 RAA	¬ (A ∧ ¬ (B ∧ D))	

A	B	C	D	¬ (A ∧ ¬ (B ∧ ¬ (C ∧ ¬ D))) / ¬ (A ∧ ¬ (B ∧ D))
T	T	F	F	Ⓣ F F T T F T Ⓕ T T F

Glen Helman 10 Sep 2011