

### 3.3.2. Using lemmas to complete *reductios*

Now that we have IP, we are in a position to provide a proof for any argument whose validity depends only on the properties of  $\top$ ,  $\perp$ , conjunction, and negation. To do so we would often need to use LFR—or, in simpler cases, Adj—to make any use of negative resources. This poses no problems when we construct derivations for valid arguments, but LFR does not itself exploit resources so negated compounds would remain as active resources until a gap is closed. In order to count an open gap as having reached a dead end, we would need some description of the conditions under which LFR had been used often enough. Such a description could certainly be given; and, in the last two chapters, we will need to take an analogous approach in the case of one of the rules for quantifiers. But it is possible to keep track of the use of negative resources by way of a genuine exploitation rule.

The basis for such an approach was the third lesson drawn from the law of negation as a premise: if a *reductio* that has  $\neg \phi$  as a premise is valid, then the lemma  $\phi$  that we need in order to use  $\neg \phi$  to complete the *reductio* is a valid conclusion from the premises other than  $\neg \phi$ . That means not only that it is safe to introduce  $\phi$  as a lemma but also that  $\neg \phi$  can be dropped from the active resources of the gap in which we establish the lemma. Of course,  $\neg \phi$  is needed along with the lemma to reach the goal  $\perp$ , but we need not introduce a second gap to reach this goal (as would be done with LFR), for such a gap would close immediately by Nc. And, indeed, the law for negation as a premise tells us not only that we can reach the needed lemma (provided that the *reductio* is valid to begin with) but also that this is all that we need to do, for it says that the *reductio* argument  $\Gamma, \neg \phi / \perp$  is valid only if the argument  $\Gamma / \phi$  is, too—but also that  $\Gamma, \neg \phi / \perp$  is valid if  $\Gamma / \phi$  is.

We will call the rule that implements these ideas **Completing a Reductio** (CR).

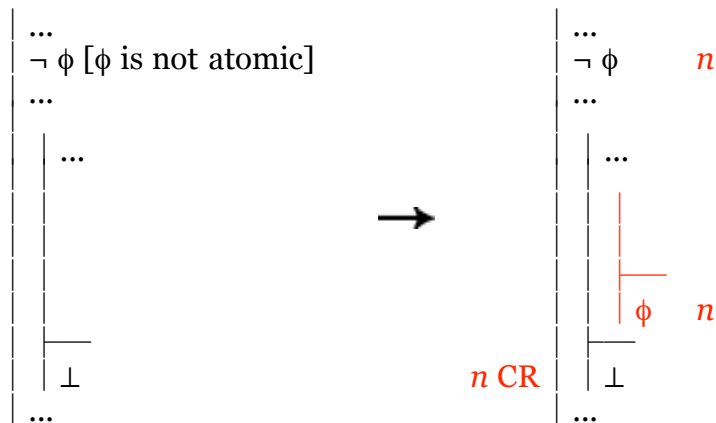
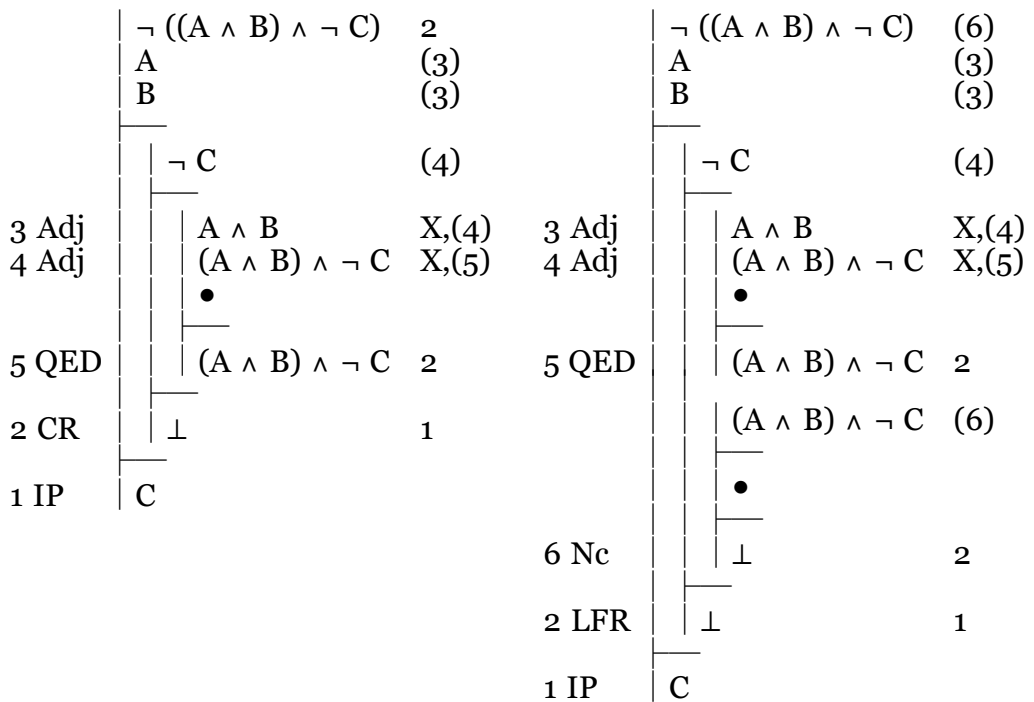


Fig. 3.3.2-1. Developing a derivation by exploiting a negated compound at stage  $n$ .

The motivation for this rule lies in its use with the negations of non-atomic sentences; and, in fact, we must limit its use to such sentences. It is sound and safe in the case of atomic sentence, but it would not be progressive in that case (given the way we are measuring distance from a dead end) because it would replace a resource that we never exploit by a goal that we could go on to plan for by IP. The rules IP and CR carry us in different directions between gaps whose proximate arguments have the forms  $\Gamma, \neg \phi / \perp$  and  $\Gamma / \phi$ , so, if there is any overlap in the sentences  $\phi$  to which they apply, a derivation could move back and forth between the two forever. We block such circles by limiting IP to cases where  $\phi$  is atomic and CR to cases where  $\phi$  is non-atomic.

The following derivations show, on the left, the use of CR in a derivation for the argument from [3.2.2](#) that was used as an illustration in the last subsection and, on the right, an analogous use of LFR:



The most obvious difference between the two is the extra argument in the second in which the lemma is  $(A \wedge B) \wedge \neg C$  is explicitly used. But the more important difference is that while the first premise is exploited at stage 2 in the left-hand derivation, it remains unexploited in the second. It is true that the absence of the second gap introduced with LFR makes the first derivation shorter, but these derivations are not designed to be the most efficient way of reaching a conclusion and the left-hand derivation is longer than the one at the end of [3.3.1](#). This extra length is the result of introducing  $(A \wedge B) \wedge \neg C$  as an explicit goal. That provides us with a stage at which we can mark its negation as exploited, but it also makes explicit the resource aimed at by the two uses of adjunction.